



Design and Development of the Hello Engineer Android and Web-Based Construction Consultation Application

Ahmad Yudi^{1*}, Siti Rahma², Arief Ichwani³, Ahmad Arif Apriadi⁴

^{1,2}Teknik Sipil, Fakultas Teknik, Institut Teknologi Sumatera, Indonesia

³Teknik Informatika, Fakultas Teknik, Institut Teknologi Sumatera, Indonesia

⁴PhD Student, Department of Civil Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok

ABSTRACT



Type: Research Paper

Article:

Received: May 12, 2026

Revised: July 16, 2026

Accepted: September 13, 2026

Issued: October 30, 2026

© Yudi et al. (2026).



This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

*Correspondence Address:

ahmad.yudi@si.itera.ac.id

Access to qualified construction experts is often constrained by location, scheduling, and fragmented service channels. This study designed and developed Hello Engineer, an integrated construction-consultation platform consisting of a web-based administrative application and two Android client applications for clients and consultants. The system supports consultant registration and verification, expertise-based discovery, chat consultation, appointment scheduling, laboratory-test ordering, location-based services, ratings, and payment processing. Development followed the Dynamic Systems Development Method because the project required iterative prototyping, continuous product-owner involvement, and priority-based adjustment of features within fixed time and resource constraints. Functional black-box testing covered nine scenarios across authentication, account management, consultant selection, appointment booking, chat consultation, laboratory-test ordering, and logout. All nine scenarios passed, producing a functional-test success rate of 100%. The result demonstrates the technical feasibility of integrating previously fragmented construction advisory services in one role-based mobile and web architecture. The evaluation is limited to functional verification; standardized user-based usability, performance, security, and scalability tests remain necessary before production deployment.

Keywords: API, civil engineering consultant, android application, website

1. INTRODUCTION

Construction services increasingly depend on digital channels for coordination, information exchange, and decision support. Research on Industry 4.0 in construction shows that digital platforms can improve connectivity and service delivery, while adoption remains constrained by fragmented workflows, uneven capabilities, and organizational barriers [1–3]. Building information modelling research likewise emphasizes structured information exchange and collaboration among project stakeholders [4–7].

However, most construction digitalization research focuses on project-level modelling, organizational collaboration, and asset delivery. It provides less evidence on client-facing platforms that connect an individual seeking advice with a verified construction specialist and then support the full service transaction. In practice, a client may need to search for an expert, confirm expertise, arrange a consultation, determine a meeting location, order a laboratory service, and complete payment through separate channels. This fragmentation increases search effort and makes service status difficult to track.

Hello Engineer addresses this service-integration gap through a role-based platform for three stakeholder groups: administrators, consultants, and clients. The system combines consultant registration and document verification, expertise-based discovery, chat consultation, appointment scheduling, consultant ratings, laboratory-test ordering, location services, and online payment. The contribution is therefore system integration in an Indonesian construction-service context, not a new recommendation algorithm or a new software-development method. This scope distinguishes the application from project-centric BIM and Construction 4.0 platforms discussed in prior research [6–8].

The study asks whether these service functions can be implemented coherently through a web administration layer and two Android applications supported by a server-side API. The application was developed iteratively using the Dynamic Systems Development Method (DSDM) and evaluated through black-box functional testing. The study contributes (1) a stakeholder-specific requirements model, (2) an integrated client-server architecture for construction consultation services, and (3) functional evidence from nine end-to-end test scenarios.

2. METHODS

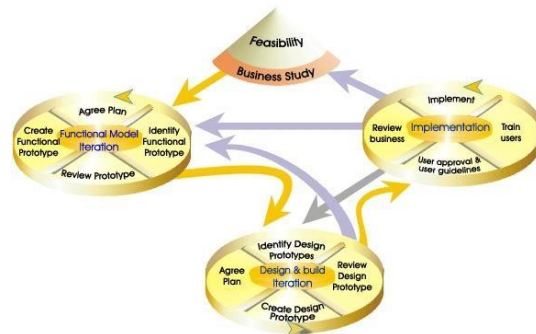


Figure 1. The DSDM Life Cycle [9]

This study used DSDM, an agile project framework that fixes time, cost, and quality while allowing lower-priority scope to be adjusted through iterative delivery. Agile development is appropriate when requirements must be clarified through frequent stakeholder feedback rather than frozen completely at project initiation [10–14]. DSDM was selected instead of a sequential Waterfall approach because Hello Engineer involved three stakeholder groups, evolving interface requirements, external services such as payment and location, and repeated product-owner review. Compared with Scrum, DSDM also supplied an explicit feasibility and foundations structure and a defined approach to prioritizing requirements within fixed delivery constraints.

The study standardized stakeholder terminology as follows. The administrator manages accounts, consultant verification, service data, and transactions through the web application. A consultant is a verified construction professional who offers consultation or appointment services through the Android consultant application. A client is a service seeker who discovers consultants, books services, communicates, orders laboratory tests, and reviews transaction status through the Android client application. Requirements were documented by stakeholder and separated into functional and non-functional categories in accordance with requirements-engineering practice [15,16].

The DSDM process comprised feasibility, business study, functional model iteration, design and build iteration, and implementation. Its

concrete deliverables were a feasibility assessment and risk log; stakeholder and business requirements; prioritized functional and non-functional requirements; interface prototypes; a client-server architecture; a relational database schema; use-case models; Android and web builds; and black-box test scenarios. Figure 1 summarizes the lifecycle used in the project.

During feasibility, the team assessed the technical environment, stakeholder access, and the viability of integrating mobile clients, a web administrator interface, a database, location services, and a payment gateway. The business study then translated the service workflow into stakeholder-specific requirements and prioritized core transactions. During functional model iteration, prototypes were reviewed with the product owner and requirements were refined. The design and build iteration produced the database, API-connected applications, and test cases. Implementation produced the working prototype and the functional-test report.

The architecture follows a client-server pattern. The Android applications communicate with server-side endpoints that mediate access to application data, while the web application supports administrative operations. This separation supports centralized business rules and consistent data access. REST principles provide the conceptual basis for resource-oriented network services, while HTTP defines request and response semantics [17–19]. Location-based functions obtain device position for service coordination [20].

Security and privacy were treated as mandatory deployment requirements because the platform processes account credentials, consultant documents, location data, consultation messages, and payment status. The required controls include authenticated sessions or tokens, role-based authorization, salted password hashing, TLS-protected API traffic, server-side input validation, rate limiting, audit logging, least-privilege database access, and explicit retention rules. OAuth guidance, application-security verification controls, and digital-identity standards provide appropriate implementation baselines [21–23]. Payment-card data should remain within the

payment gateway's compliant environment rather than being stored by Hello Engineer, consistent with PCI DSS [24]. The present study did not conduct penetration testing, so security effectiveness is not claimed as an empirical result.

3. RESULTS AND DISCUSSION

3.1 Feasibility Study

This phase describes the preparations within the software developer's internal environment, including: Research-object analysis

Analyze the application of the appropriate system development method in building the Hello Engineer application. Since this application is being built from scratch, close collaboration between developers and product owners is essential. Therefore, implementing the DSDM life cycle can serve as an effective approach to developing the application.

Analyzing System Requirements

Conducting a system requirements analysis to support software development. The specifications of the equipment used are as follows.

- Intel Core i3
- RAM 8GB
- SSD 512GB
- Android Studio Bumblebee
- XAMPP v3.3.0 (PHP and MySQL 8.0.7)

Targeted Achievements

The goal is to make it easier for clients to seek consultations in the fields of building construction, water resources, and road construction.

3.2 Business Studies

At this stage, a general description of the system is proposed to provide an overview of how the system works. The system framework consists of two main components: an Android mobile app for users and construction experts, and a web-based platform for the admin side. A simplified view of the system architecture is shown in Figure 2 below.

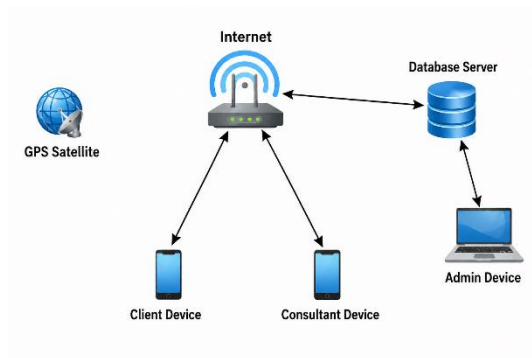


Figure 2. System Overview

The application in this system uses Location-Based Service (LBS) technology and a

client-server architecture. LBS is used to obtain location data for users (both users and consultants) through GPS positioning on mobile devices. In terms of network communication, a client is a device or process that sends data requests to the server. Meanwhile, the server provides the data requested by the client computer.

3.3 Functional Model Iteration

At this stage, the system's main functions are designed. A prototype of the application's user interface is created. System requirements are also specified. Menu design in the admin panel

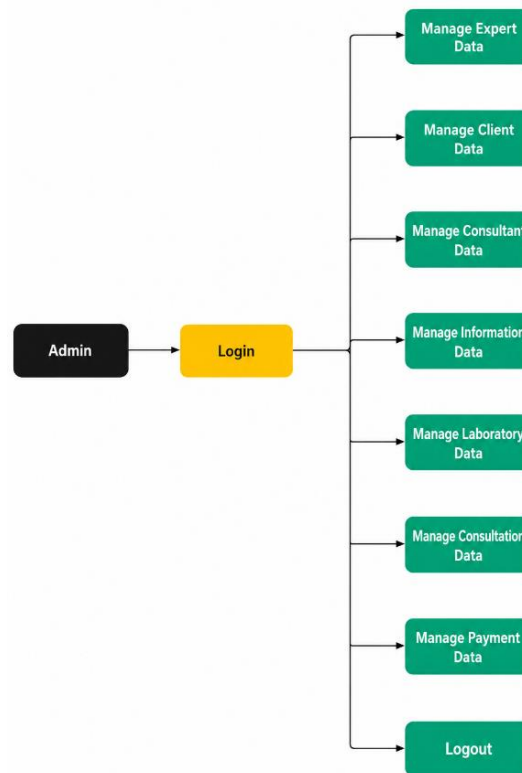


Figure 3. Admin Menu Design

The admin is one of the most important entities in the system because they play a crucial role in managing all transactions between consultants and users. The admin is responsible for ensuring the system operates smoothly for all users to achieve the organization's goals.

Menu Design on the Client Device

Clients are application users who play a role in the service transaction process. Clients can access information about construction experts provided by the administrator. Clients can also manage their accounts, view available lab types, view their service order history, and check the payment status for requested services.

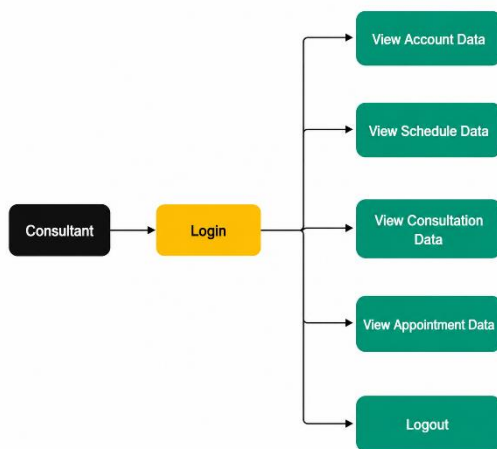


Figure 4. Client menu design

Meanwhile, system requirements analysis is the stage of determining the specifications of system requirements and the objectives to be achieved. The design was derived from interviews with administrators and the results of general field analysis. The system requirements from the administrator's perspective are shown in Table 1 below.

Table 1. Functional and non-functional requirements from the administrator's perspective

Functional Requirements	
No	Requirements analysis: The admin wants the system to be able to
1	Register a consultant
2	Register a user
3	Managing consultant data
4	Managing user data
5	Managing consultation data
6	Managing Information Data
7	Managing expertise data
8	Managing lab data
9	Managing payment data
10	Log out
Non-Functional Requirements	
1	The system has an easy-to-understand interface
2	The wait time from when a page begins to load until the content is displayed or generated takes no more than 10 seconds
3	The system can be run on several web browsers, such as Google Chrome, Mozilla Firefox, Opera, Internet Explorer, etc.

Next, the system requirements from the consultant's perspective are outlined in Table 2 below.

Table 2. Functional and non-functional requirements from the consultant's perspective

Functional Requirements	
No	Requirements analysis: The consultant wants the system to be able to
1	Create an account
2	Validating users
3	Manage your account
4	Consultation Services
5	Managing Appointments
6	View service order history
7	Managing data on consultation schedules and appointments
8	Log out
Non-Functional Requirements	
1	The system has an easy-to-understand interface
2	The wait time from when a page begins to load until the content is displayed or generated takes no more than 10 seconds
3	Consultant photo file extensions: jpeg, jpg, bmp, and png

Meanwhile, the system requirements from the client's perspective are outlined in Table 3 below.

Table 3. Functional and non-functional requirements from the client's perspective

Functional Requirements	
No	Requirements analysis: The client wants the system to be able to
1	Create an account
2	Validating users
3	Manage your account
4	Schedule a consultation
5	Making plans to meet
6	View service order history
7	Conducting lab tests
8	Log out
Non-Functional Requirements	
1	The system has an easy-to-understand interface
2	The wait time from when a page begins to load until the content is displayed or generated takes no more than 10 seconds
3	User photo file extensions: jpeg, jpg, bmp, and png

Planning and Development Iterations

The modeling to be presented includes designs in the form of database schemas and use case diagrams.

Database Schema

The database schema for the Hello Engineer application is as follows.

- Admin

id_admin	PK
username	
password	
name	

- User

id_user	PK
email_user	
password_user	
name_user	
telephone_user	
location_user	
job_user	
photo_user	
status_user	
activation_user	

- Consultant

id_konsultan	PK
consultant_email	
consultant_password	
consultant_name	
consultant_phone	
consultant_location	
consultant_resume	
consultant_experience	
photo_consultant	
rating_consultant	
status_consultant	
activation_consultant	
list_consultant	
level_consultant	

- Consultant_expertise

id_consultant_expertise	PK
id_consultant	FK
id_expertise	FK
document	
status	

- Skills

skill_id	PK
skill_image	
skill_name	
skill_description	

- Ordering

order_id	PK
rate_id	FK
date	
expiration	
total	
service_id	FK
status	
rate_name	

- Type

id_type	PK
type_name	
type_rate	
type_description	

- Consultation

consultation_id	PK
user_id	FK
consultant_id	FK
consultation_date	
consultation_status	
consultation_rating	
consultation_review	

- Schedule

id_schedule	PK
id_consultant	FK
day	
start	
end	
status	

- Promise

appointment_id	PK
user_id	FK
consultant_id	FK
schedule_id	FK
date	
time	
location	
appointment_status	
notes	

- Recipe

id_tarif	PK
rate_type	
rate_name	
rate_amount	
rate_level	

- Experience

id_experience	PK
id_consultant	FK
name	
description	

- Lab

id_lab	PK
lab_name	
lab_rates	
lab_test_details	
lab_photos	

- Chat

id_chat	PK
id_consultation	FK
content	
date	
sender	
read	
deleted	
attachment	

- Category

category_id	PK
category_name	

- Information

id_information	PK
id_category	FK
title_information	
url_information	
content_information	
image_information	
date_information	

Meanwhile, the use cases for the Hello Engineer app are shown in Figure 5 below.

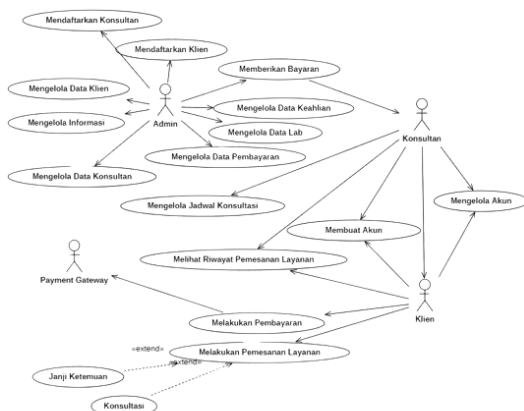


Figure 5. Hello Engineer Use Case Diagram

Hello Engineer Dashboard Page

On this page, anyone with expertise in the fields of structural engineering, geotechnical engineering, transport, construction management and water resources can register as a professional construction consultant. Then, the admin will verify the registration based on the documents uploaded.

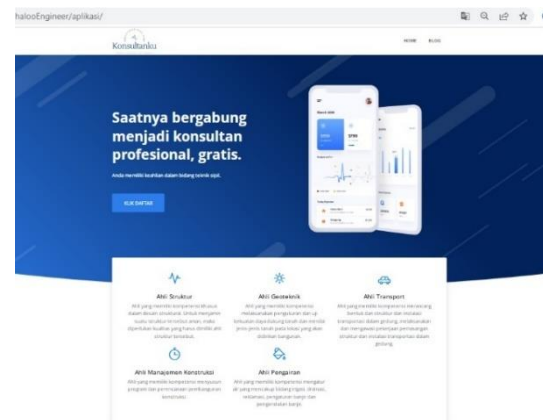


Figure 6. Hello Engineer Dashboard Page

Consultants who have been verified by the administrator can log in to the mobile app to provide services.

Hello Engineer Mobile Dashboard

On this page, clients can carry out various activities such as viewing consultants' areas of expertise, seeking advice, viewing consultants' schedules, booking appointments, and ordering laboratory tests. To seek advice, book an appointment or order a laboratory test, clients must pay the specified fee. All payments are processed via the Midtrans payment gateway.



Figure 7. Mobile Dashboard Page for Clients

Select Service Type

On this page, clients can view consultants' ratings before deciding which type of service to use. There are two types of service: booking an appointment or having a consultation via chat.

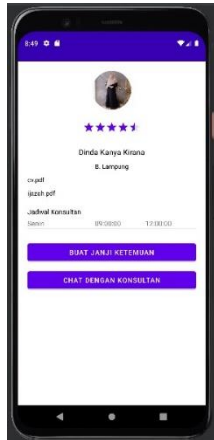


Figure 8. Service Type Selection Page

Booking an Appointment

On this page, the client will select an available consultant's schedule, specify the date and time, and choose a meeting venue. However, the consultant may send their approval or change the meeting location. Once the appointment has been agreed upon by both parties (the client and the consultant), the process will proceed directly to payment, as shown in Figure 9.

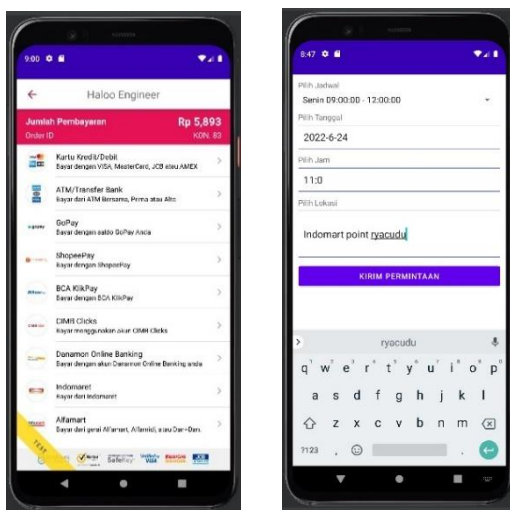


Figure 9. Book an Appointment Page

Consultation Page

Clients can book a consultation by selecting their preferred area of expertise. They then make a payment in accordance with the quoted rate. Once the client has made the payment, the system will automatically verify the payment details. Once the payment has been validated, the system will redirect the client to the chat page with their chosen consultant.



Figure 10. Consultation Page

Functional Black-Box Testing

Black-box functional testing evaluated whether each predefined input and action produced the expected system response. Table 4 contains nine scenarios: two login scenarios, one account-update scenario, three consultant-service scenarios, one consultation scenario, one laboratory-test scenario, and one logout scenario. All nine scenarios passed, yielding 9/9 successful cases and a functional-test success rate of 100%. This result verifies the tested functions but must not be interpreted as evidence of usability. Usability concerns effectiveness, efficiency, and satisfaction in a specified context of use and requires user-based measures such as task completion, error rate, completion time, and standardized instruments such as the System Usability Scale [25–28].

Table 4. Hello Engineer Functional Black-Box Testing

No.	Input or action	Expected result	Result
1	Valid login credentials	Login succeeds	Pass
2	Invalid login credentials	Login is rejected	Pass
3	Update account details	Changes are saved	Pass

No.	Input or action	Expected result	Result
4	Filter by expertise	Matching consultants appear	Pass
5	Book an appointment	Schedule and fee appear	Pass
6	Start chat consultation	Service fee appears	Pass
7	Open consultation history	Chat page opens	Pass
8	Select laboratory test	Requirements and fee appear	Pass
9	Select logout	Session ends	Pass

The functional result indicates that the integrated transaction flow was technically coherent for the scenarios tested. Clients could authenticate, manage account data, identify a consultant by expertise, initiate an appointment or chat service, access consultation history, order a laboratory test, and leave the application. The practical contribution is the consolidation of these functions with consultant verification, location support, and payment status in one role-based architecture. In contrast, the construction digitalization literature has largely emphasized organizational information models, project collaboration, and lifecycle data exchange [1,4–8]. Hello Engineer shifts that emphasis toward access to individual professional services.

The architecture also clarifies responsibility boundaries. Administrators verify consultants and manage platform data; consultants manage availability and deliver professional services; and clients initiate and track service transactions. Centralized server-side mediation can support consistent validation and data integrity, but it also creates a critical dependency on API availability and secure authorization. Architectural quality therefore depends not only on feature completeness but also on maintainability, performance efficiency, reliability, security, and scalability [16,17].

Several limitations constrain the findings. First, the nine test cases verify selected functions only; they do not measure usability, accessibility, response time, load capacity, GPS accuracy, transaction latency, reliability, or recovery behavior. Second, the study did not report participant-based evaluation or SUS scores. Third, consultant-verification speed and accuracy were not measured. Fourth, the security controls described above were not validated through code review or penetration testing. Consequently, the

prototype should undergo user-based usability testing, API and server-load testing, security assessment, payment failure and refund scenarios, location-accuracy evaluation, and longer-term reliability testing before production deployment.

CONCLUSION

This study developed Hello Engineer as an integrated web and Android platform for connecting clients with verified construction consultants. DSDM supported iterative clarification of requirements across administrator, consultant, and client roles and produced stakeholder requirements, interface prototypes, a relational data model, a server-mediated architecture, implemented applications, and a functional-test report. All nine black-box scenarios passed, corresponding to a 100% success rate for the functions tested. The main contribution is the integration of consultant verification, expertise-based discovery, chat consultation, appointment scheduling, laboratory-test ordering, location services, ratings, and payment processing within one service workflow. The result does not establish usability, performance, scalability, reliability, or security effectiveness. Future work should evaluate these qualities with representative users and operational workloads. Any future recommendation feature should rely on service-relevant and consented attributes rather than demographic proxies such as age or gender, and should include privacy, fairness, and explainability safeguards.

DECLARATIONS

Author Contributions: The first author conceptualized the study, designed the methodology, administered the project, and wrote the original draft of the manuscript. The second author conducted the requirements analysis and investigation and contributed to reviewing and editing the manuscript. The third author developed the software, implemented the system, and performed the formal analysis. The fourth author validated the results, supervised the research, and contributed to

reviewing and editing the manuscript. All authors contributed to the development and refinement of the manuscript, read and approved the final version, and agreed to be accountable for the work presented in this study.

Funding: The authors would like to express their sincere gratitude to the Institute for Research and Community Service (LPPM), Institut Teknologi Sumatera, South Lampung Regency, Indonesia, for providing research grant funding and supporting the successful completion of this study.

Conflict of Interest: The authors declare no conflict of interest.

Generative Artificial Intelligence Use Statement: No generative AI tools or large language models were used to generate the scientific content, research data, analysis, interpretation, or conclusions of this manuscript.

Ethics Approval and Informed Consent: Formal ethical approval was not required for this study, which focused on software development and functional testing. Requirements interviews with administrators were conducted voluntarily, and verbal consent was obtained from the participants.

Data Availability: The data supporting the findings of this study, including system requirements documentation, the database schema, and black-box functional testing scenarios, are available from the corresponding author upon reasonable request.

ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to the Institute for Research and Community Service (LPPM), Institut Teknologi Sumatera, South Lampung Regency, for providing research grant funding to support this research.

REFERENCES

- [1] Oesterreich TD, Teuteberg F. Understanding the implications of digitisation and automation in the context of Industry 4.0: A triangulation approach and elements of a research agenda for the construction industry. *Computers in Industry*. 2016;83:121–139.
- [2] Alaloul WS, Liew MS, Zawawi NAWA, et al. Industrial Revolution 4.0 in the construction industry: Challenges and opportunities for stakeholders. *Ain Shams Engineering Journal*. 2020;11(1):225–230.
- [3] Maskuriy R, Selamat A, Ali KN, et al. Industry 4.0 for the construction industry: Review of management perspective. *Economies*. 2019;7(3):68.
- [4] Turk Z. Ten questions concerning building information modelling. *Building and Environment*. 2016;107:274–284.
- [5] Oraee M, Hossein Rezazadeh Azar E, Golparvar-Fard M, et al. Collaboration in BIM-based construction networks: A bibliometric-qualitative literature review. *International Journal of Project Management*. 2017;35(7):1288–1301.
- [6] Succar B. Building information modelling framework: A research and delivery foundation for industry stakeholders. *Automation in Construction*. 2009;18(3):357–375.
- [7] Sacks R, Eastman C, Lee G, et al. *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers*. 3rd ed. Wiley; 2018.
- [8] Sawhney A, Riley M, Irizarry J, editors. *Construction 4.0: An Innovation Platform for the Built Environment*. Routledge; 2020.
- [9] Zulfikar RA, Supianto AA. Rancang Bangun Aplikasi Antrian Poliklinik Berbasis Mobile. *J Teknol Inf dan Ilmu Komputer*. 2018;5(3).
- [10] DSDM Consortium. *The DSDM Agile Project Framework*. DSDM Consortium; 2014.
- [11] Beck K, Beedle M, van Bennekum A, et al. *Manifesto for Agile Software Development* [Internet]. 2001. Available from:

- <https://agilemanifesto.org/>.
- [12] Conboy K. Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*. 2009;20(3):329–354.
 - [13] Highsmith J, Cockburn A. Agile software development: The business of innovation. *Computer*. 2001;34(9):120–127.
 - [14] Abrahamsson P, Salo O, Ronkainen J, et al. Agile Software Development Methods: Review and Analysis. VTT Technical Research Centre of Finland; 2002. Report No.: 478. .
 - [15] ISO/IEC/IEEE. ISO/IEC/IEEE 29148:2018 Systems and Software Engineering: Life Cycle Processes: Requirements Engineering. International Organization for Standardization; 2018.
 - [16] ISO/IEC. ISO/IEC 25010:2011 Systems and Software Engineering: Systems and Software Quality Requirements and Evaluation: System and Software Quality Models. International Organization for Standardization; 2011.
 - [17] Bass L, Clements P, Kazman R. Software Architecture in Practice. 4th ed. Addison-Wesley; 2021.
 - [18] Fielding RT. Architectural Styles and the Design of Network-Based Software Architectures [Doctoral dissertation]. University of California, Irvine; 2000.
 - [19] Fielding R, Nottingham M, Reschke J. HTTP Semantics. Internet Engineering Task Force; 2022. Report No.: RFC 9110. .
 - [20] Schiller J, Voisard A, editors. Location-Based Services. Morgan Kaufmann; 2004.
 - [21] Hardt D. The OAuth 2.0 Authorization Framework. Internet Engineering Task Force; 2012. Report No.: RFC 6749. .
 - [22] OWASP Foundation. Application Security Verification Standard 4.0.3. OWASP Foundation; 2021.
 - [23] Grassi PA, Garcia ME, Fenton JL. Digital Identity Guidelines: Authentication and Lifecycle Management. National Institute of Standards and Technology; 2017. Report No.: NIST SP 800-63B. .
 - [24] PCI Security Standards Council. Payment Card Industry Data Security Standard: Requirements and Testing Procedures, Version 4.0.1. PCI Security Standards Council; 2024.
 - [25] ISO. ISO 9241-11:2018 Ergonomics of Human-System Interaction: Usability: Definitions and Concepts. International Organization for Standardization; 2018.
 - [26] Brooke J. SUS: A Quick and Dirty Usability Scale. In: Jordan PW, Thomas B, McClelland IL, et al., editors. Usability Evaluation in Industry. Taylor & Francis; 1996. p. 189–194.
 - [27] Bangor A, Kortum PT, Miller JT. An empirical evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*. 2008;24(6):574–594.
 - [28] Lewis JR. The System Usability Scale: Past, present, and future. *International Journal of Human-Computer Interaction*. 2018;34(7):577–590.