

# Development of a Software Application for Calculating Cold-Water Pipe Sizes in Residential Buildings up to Two Floors

Parichatr Yankaew  
College of Arts, Media and Technology  
Chiang Mai University  
Chiang Mai, Thailand  
parichatr\_ya@cmu.ac.th

Siraprapa Wattanakul  
College of Arts, Media and Technology  
Chiang Mai University  
Chiang Mai, Thailand  
siraprapa.w@cmu.ac.th

**Abstract**— *Current cold water supply design practices for small residential buildings rely heavily on manual spreadsheet calculations and multiple document files, leading to accumulated errors, fragmented data, and limited traceability when design revisions occur repeatedly. This study addresses these issues by developing a web application that integrates automated hydraulic calculations and design version control into a single platform for low-rise residential buildings (up to two floors). The system was developed following the Software Development Life Cycle (SDLC), encompassing requirements elicitation and analysis, version-centric system architecture and database design, and the implementation of a hydraulic calculation module based on the Hazen–Williams equation in compliance with Engineering Institute of Thailand (EIT) standards. System validation was conducted across three testing levels, covering a total of 131 test cases. These levels included Unit Testing to verify the accuracy of the calculation functions, System Testing to evaluate overall operation, and User Acceptance Testing (UAT) with three practising plumbing design engineers. Performance was measured by comparing system-recommended pipe sizes against manual calculations, using an acceptance criterion of a maximum  $\pm 1$  commercial pipe size deviation, alongside assessments of version control efficiency and Time on Task measurements. The results showed that the system passed all criteria, accurately computing pipe sizes and hydraulic parameters in accordance with EIT standards. It successfully captured design histories, maintained an element-level audit log of pipe modifications, and facilitated side-by-side comparisons between versions. Furthermore, the system reduced task completion time by 38–47% across three residential case studies compared to manual methods, particularly during the design revision phase.*

**Keywords:** *cold-water pipe sizing; Hazen–Williams equation; version control; web application; residential building; plumbing design.*

## I. INTRODUCTION

Building design is an inherently multidisciplinary process that integrates structural engineering, architecture, and mechanical, electrical, and plumbing engineering. Among these systems, the cold-water pipe system (CWS) plays a fundamental role in ensuring adequate water supply to bathrooms, kitchens, and service areas. In contemporary practice, building layouts are frequently revised during the design development phase to satisfy spatial constraints, budgetary requirements, and client preferences. Each revision potentially alters fixture locations, pipe routing distances, and zone boundaries, thereby requiring

the responsible engineer to recalculate pipe sizes and regenerate all associated documentation.

The current cold-water pipe sizing process continues to rely heavily on multiple files and spreadsheets, primarily driven by manual procedures. These include redundant data entry across numerous files or worksheets, manual data transfer between documents, complex formula linking, cross-sheet data referencing, and adjusting parameters to meet specific project requirements. Without a clear version-control mechanism, these factors can generate cumulative errors, ultimately leading to inconsistencies among the calculation tables, design drawings, and specified pipe materials [1]. Furthermore, the fragmentation of data across multiple files significantly diminishes traceability, as the sources of numerical values, calculation assumptions, and key parameters are not consolidated within a single centralized data repository.

Current tools for handling data and information in building design management are Computer-Aided Design (CAD) and Building Information Modeling (BIM). CAD software generates two-dimensional layout drawings and is widely used for pipe routing because it is lightweight, familiar to practitioners, and requires no specialist training [2]. However, CAD functions solely as a drafting tool: it provides no mechanisms for integrated data management, automated hydraulic calculation, or version tracking. BIM, by contrast, is a process of creating an integrated project model that links multidisciplinary data—structural, architectural, and building-systems—enabling three-dimensional visualisation and serving as a shared information database for design and construction [3]. BIM-enabled workflows can link pipe sizing data directly to three-dimensional building models and facilitate automated clash detection [4]. However, full BIM adoption in small-scale construction projects is constrained by high software licensing costs, a shortage of trained BIM operators, and excessive model complexity relative to the scope of work, such as found in Nigerian construction projects [5]. Consequently, practising engineers in small-project contexts might not be capable of applying BIM's management capabilities.

Another limitation of BIM and CAD lies in their version control systems (VCS). These tools typically track changes at the whole-file level, whereas practising engineers implement modifications at the individual element level [6], [2]. For

example, when a pipe diameter is revised, the entire model file is saved as a new version. There is no built-in mechanism to record granular changes—such as a specific pipe segment changing from 25 mm to 32 mm—nor does it capture the design rationale behind the modification or its subsequent impact on the hydraulic system. Furthermore, any alteration in the pipe network directly affects the hydraulic parameters that support design decisions. Such element-level changes and their cascading effects cannot be effectively tracked by the conventional VCS of BIM and CAD platforms [6], [2]. Cheng et al. [7] further observed that practitioners compensate with informal file copies as ad-hoc version branches, generating naming confusion and loss of design history without providing any structured comparison capability.

This research addresses the identified gaps by developing a lightweight, purpose-specific web application that automates the EIT six-step cold-water pipe-sizing procedure and incorporates a snapshot-based version control system. The application targets residential buildings up to two floors—the most common residential building category in Thailand—and follows both EIT plumbing design standards and the ISO 29110 software development life-cycle framework for very small enterprises. The remainder of this paper is organised as follows: Section II is background; Section III describes the system design and implementation methodology; Section IV presents and discusses results; and Section V concludes with directions for future work.

## II. BACKGROUND

The theoretical foundation for cold-water pipe sizing is well established. Oungpakorn [1] formalised a six-step procedure for domestic plumbing design: (i) collect fixture unit (FU) values for every pipe segment; (ii) compute peak water demand from the total FU; (iii) apply a Water Factor correction when the total FU exceeds 400 to account for demand diversity; (iv) size main and branch pipes so that flow velocity remains within 3.0 m/s (with a recommended range of 1.2 to 2.4 m/s); (v) determine sub-branch sizes based on individual fixture demand; and (vi) iteratively adjust pipe sizes if total pressure loss is unacceptable. Fig. 1 summarises this workflow.

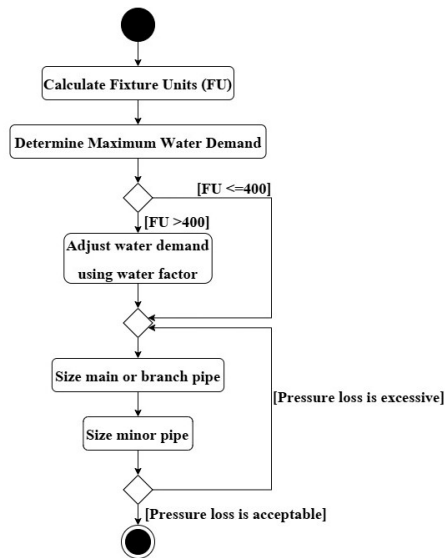


Fig. 1. Six-step cold-water pipe sizing process [1].

Pipe sizing according to engineering standards must be completed before BIM model generation; otherwise the resulting model will not reflect a valid design. Shin et al. [4] confirmed this through an automated CAD-to-BIM conversion pipeline for restroom sanitary plumbing systems, demonstrating that their approach—comprising pipe processing, connectivity analysis, elevation setting, fitting insertion, and clash resolution modules—can reduce data-transfer time and minimise manual-transcription errors. This finding directly motivates the need for a reliable standalone pipe-sizing tool that can serve as the upstream input to a BIM workflow.

For small-scale building projects, BIM adoption faces three primary barriers: resource constraints, limited human-resource capacity (trained BIM operators), and excessive software complexity relative to project scope. Agwa and Çelik [5] systematically catalogued these obstacles, as found in Nigerian construction projects, highlighting that small-project practitioners are particularly constrained by BIM's overhead.

Version control in design software has received growing research attention. Esser [6] analysed BIM model versioning and found that current systems store whole-file snapshots, making it impossible to identify which specific element changed between versions without manual inspection. He proposed a graph-based approach in which each model element is represented as a node and changes are recorded as edge events, enabling granular, element-level difference reporting. This graph-based approach represents the type of element-level tracking that integrated BIM platforms such as Revit should incorporate to support meaningful design change management. Frazelle [2] examined CAD file formats and noted that binary representations prevent meaningful semantic differencing—it is possible to detect changed bytes but not to interpret that a particular pipe's nominal diameter increased. Cheng et al. [7] surveyed engineers about branching behaviour in CAD workflows and found that practitioners commonly create file copies as informal version branches, leading to naming confusion and data loss. Cristie and Joyce [8] implemented GHShot, a snapshot-based version control plug-in for the parametric design tool Grasshopper, and demonstrated that storing design snapshots at meaningful decision points significantly improves process transparency and facilitates retrospective design analysis. A meaningful graphical-based version control has the potential to improve design information change tracking and facilitate process streamlining.

Existing commercial pipe-sizing tools—including Pipe Flow Calculations, Pipe Sizer, and PipeFlow Expert—provide hydraulic computation capabilities but offer no project-level data management, no automatic Water Factor adjustment, no integration with version tracking, and no output tailored to EIT standards or Thai commercial pipe sizes. This gap motivates the present work.

## III. METHODOLOGY

### A. System Scope

The system targets residential buildings up to two floors with cold-water supply only. Calculations reference EIT plumbing standards, specifically: fixture unit tables, the Hazen–Williams equation for friction loss, a maximum allowable velocity of 3.0

m/s (with a preferred range of 1.2 to 2.4 m/s), and commercially available Thai pipe sizes (nominal diameter). Development followed ISO 29110 SDLC covering requirements elicitation, architecture design, implementation, and testing.

### B. Functional Modules

The application is partitioned into three functional modules. The Project Data Management module supports creation and storage of project metadata (name, type, floor count), central design criteria, and floor-plan image import for visual reference. The Hydraulic Calculation Module automates the complete six-step EIT pipe-sizing procedure: fixture unit aggregation per zone; peak-flow computation; automatic Water Factor adjustment when total FU exceeds 400; main, branch, and sub-branch diameter selection using the Hazen–Williams equation; velocity and head-loss verification; and auto-suggest resizing with user-approval. The Version Control System records the full design state network topology, fixture schedule, and calculation results as a JSON snapshot per version, enabling duplication, history browsing, and side-by-side engineering comparison. An element-level audit log additionally records each pipe-segment modification with before and after values, timestamp, and user identity. The overall system architecture is illustrated in Fig. 2.

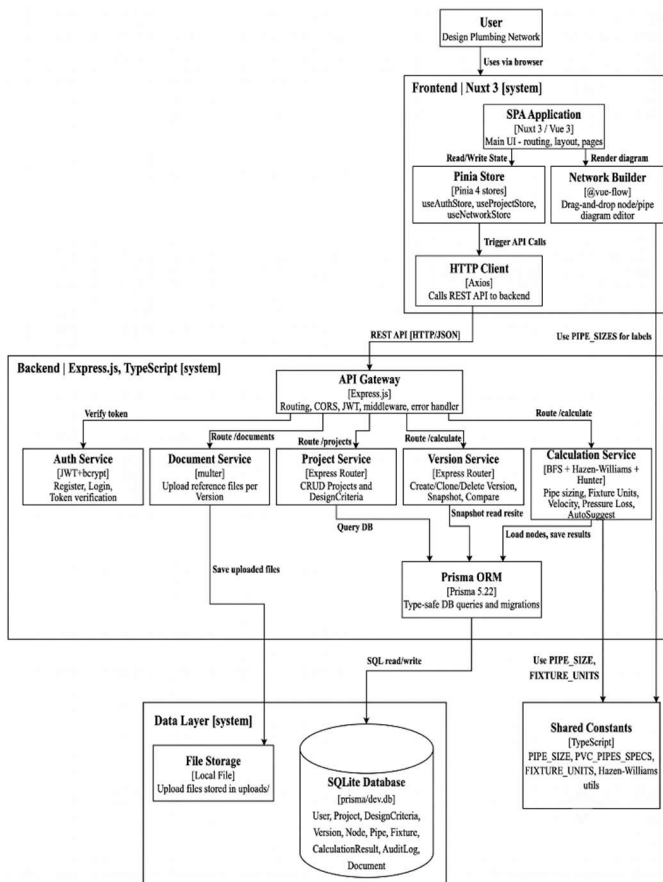


Fig. 2. System architecture

### C. Technology Stack

The front-end is built with Nuxt.js (Vue 3 Composition API) and Pinia state management, providing real-time

reactivity when pipe parameters change. Vuelidate enforces input constraints (positive lengths, valid FU values) before data submission. The back-end uses Node.js with Express.js for RESTful API endpoints; Express-validator guards against malformed JSON payloads. Prisma ORM manages SQLite persistence and prevents SQL injection. Version snapshots store network topology (nodes and pipe segments) and fixture lists as JSON TEXT fields, ensuring each version is self-contained and independently retrievable. Vitest provides unit testing of hydraulic functions; Supertest covers API integration; and Playwright performs end-to-end interaction testing on the canvas-based network editor.

### D. Testing Strategy

Three testing levels were executed: Unit Testing (81 cases) covering FU conversion, Hazen–Williams pipe selection, Water Factor edge cases, velocity boundary checks, and version snapshot save/load operations; System Testing (40 cases) verifying end-to-end integration across all functional modules including multi-version creation and side-by-side comparison; and User Acceptance Testing (UAT: 10 cases) conducted with three practising plumbing design engineers in a staging environment, each completing the same ten realistic design scenarios.

## IV. RESULTS AND DISCUSSION

### A. Calculation Accuracy

Unit testing produced 81 passes out of 81 cases with no failures. All functions were verified including FU conversion, Hazen–Williams pipe selection, Water Factor edge cases, velocity boundary checks, and version snapshot save/load operations.

System testing was conducted across 40 scenarios to verify end-to-end operational behaviour. Through an iterative testing process, minor interface and rendering adjustments were implemented, ultimately resulting in the system successfully passing all 40 test cases. Calculation accuracy was further validated using three representative building scenarios also employed in UAT: a 2-story townhouse (C1), a 1-story single-detached house (C2), and a 2-story single-detached house (C3). These cases were selected to span the range of building complexity (floor count and fixture count). System outputs were compared against pre-computed manual reference solutions derived from the EIT six-step procedure; three licensed engineers reviewed both sets of results and confirmed that all outputs fell within  $\pm 1$  commercial pipe size. Table I presents this comparison.

TABLE I. PIPE SIZE COMPARISON: MANUAL VS. SYSTEM (MAIN PIPE)

Case	Floors	Fixtures	Manual Size (mm)	System Size (mm)	Match
C1	2	11	32	32	✓
C2	1	20	40	40	✓
C3	2	18	40	40	✓

### B. Version Control Capability

The version control module successfully stored independent snapshots for each design iteration. In a controlled test, a project was revised two times (V1–V2), each time changing the fixture count in one zone. The system correctly maintained two separate, independently retrievable snapshots, each containing the full input–output state. Side-by-side comparison of V1 and V2 displayed the differences in peak flow rate, main-pipe diameter, and total head loss, confirming design traceability and version comparison capability. The snapshot states captured for V1 and V2 are shown in Fig. 3. The main calculation interface displaying fixture input and pipe-size results for each version is illustrated in Fig. 4.

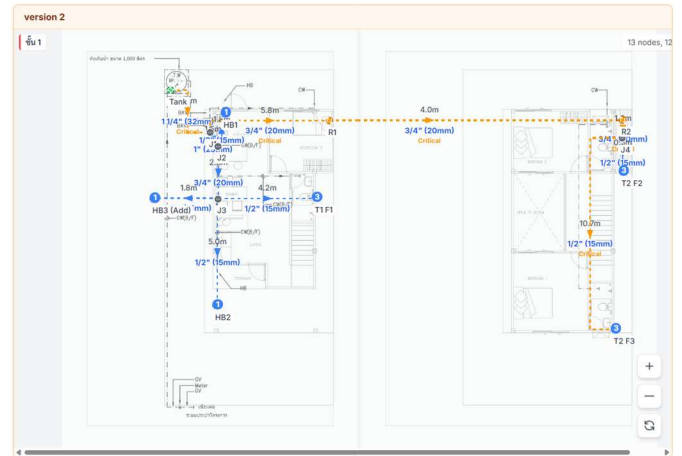
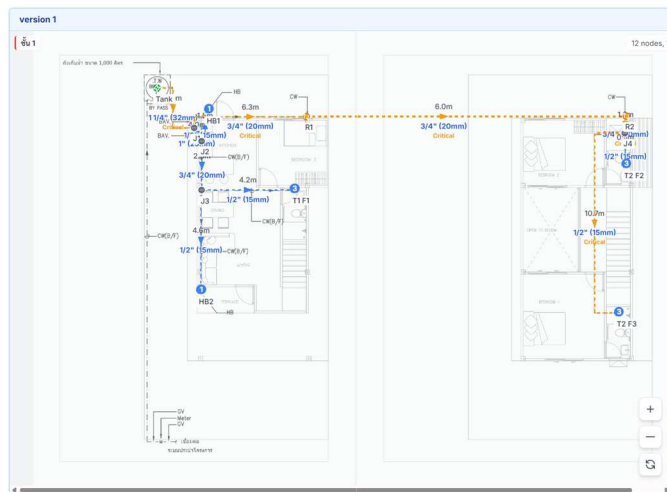
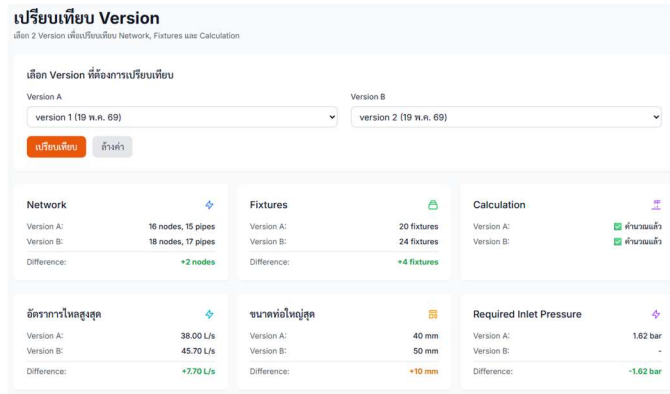


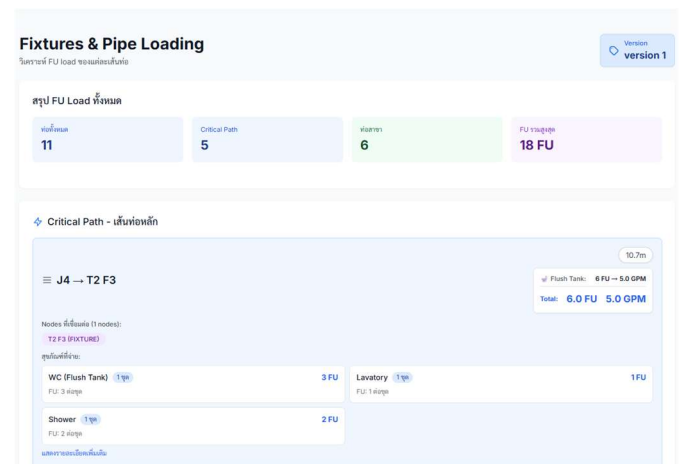
Fig. 3. Side-by-side version comparison: hydraulic parameter differences between V1 and V2.

### C. User Acceptance Testing

Three independent environmental/sanitary engineers participated in UAT (Table II). All testers had at least five years of professional plumbing design experience and had no prior exposure to the system, ensuring independent evaluation. Each tester completed ten test scenarios covering project creation, fixture input, pipe calculation, version saving, and version comparison. UAT found no functional defects; all ten test scenarios were completed successfully on the first run by all three testers. Post-session satisfaction, measured on a five-point Likert scale, averaged 4 out of 5, exceeding the acceptance threshold of 3 out of 5. Engineers suggested adding middle-mouse-button panning and node-copy functionality, both of which were subsequently implemented.

TABLE II. UAT PARTICIPANT DETAILS

No.	Role	Gender	Age	Prior System Use
1	Sanitary Engineer	F	34	None
2	Sanitary Engineer	M	27	None
3	Sanitary Engineer	F	32	None



## Calculation

คำนวณขนาดท่อด้วยระบบ Auto Suggest และ Hybrid Sizing

### Auto Suggest Sizing

**Auto-Suggest Sizing**  
วิเคราะห์และแนะนำขนาดท่อที่เหมาะสม - Water Factor + Auto Selection

11 ท่อทั้งหมด    11 ปกติดี    0 ต้องตรวจสอบ    0 ต้องปรับปรุง

❖ **Critical Path (5 ท่อ)** เริ่มจากปลายทาง → ต้นทาง

✓ J4 → T2 F3    15mm (1/2")  
ปกติดี

Fixture Unit (FU)	Flow Rate (Hunter + HB)	Water Factor (100%)
6 FU	5.00 GPM	(5.00 × 100%) → 5.00 GPM

Total GPM	Total LPS	Total m <sup>3</sup> /s
5.00	0.32	0.00032

Nominal Diameter (mm)	Nominal Diameter (inches)	Internal Diameter (m)	ความเร็วท่อ
15	1/2	0.0163	10.7 m

C-Factor  
150

🔍 ลองขนาดท่ออื่น

ความเร็วน้ำ (v)	Friction Loss Rate	Major Loss
1.50 m/s ควร 1.2-2.4	16.67 m/100m	1.784 m.wg

⚠ WARNING: Velocity below minimum - may cause sedimentation

### Hybrid Pipe Sizing

**Hybrid Pipe Sizing**  
เลือกขนาดท่อโดยใช้ตาราง Fixture Units (Private Use) สำหรับห้องในห้องน้ำและห้องครัว

🔍 ท่อเข้าสู่ทุกชั้นแต่ละห้อง (Fixture Branch Pipes by Node)

- HB1 1 สุขภัณฑ์
  - ก๊อกสนาม (Hose Bibb) 15mm (1/2")
- T1 F1 3 สุขภัณฑ์
  - ชักโครกแบบ Flush Tank (WC Flush Tank) 15mm (1/2")
  - อ่างล้างหน้า (LAV) 15mm (1/2")
  - ฝักบัว (Shower) 15mm (1/2")
- HB2 1 สุขภัณฑ์
  - ก๊อกสนาม (Hose Bibb) 15mm (1/2")
- T2 F3 3 สุขภัณฑ์
  - ชักโครกแบบ Flush Tank (WC Flush Tank) 15mm (1/2")
  - อ่างล้างหน้า (LAV) 15mm (1/2")
  - ฝักบัว (Shower) 15mm (1/2")

Fig. 4. Main calculation interface showing fixture input and pipe-size results.

### D. Process Improvement

By consolidating project management, hydraulic calculation, and version control in one platform, the system

eliminates inter-file data transfer. Time on Task measurements across three case studies showed task completion time reductions of 38–47% compared to manual spreadsheet methods (Table III). Engineers no longer need to maintain parallel Excel workbooks for different design iterations, reducing the risk of referencing an outdated sheet. The platform's single-source-of-truth model also improves auditability: any stakeholder can open any saved version and verify the assumptions and results that underpinned a specific design decision — a capability not achievable with the conventional multi-file approach.

TABLE III. TIME ON TASK: MANUAL VS. SYSTEM (MINUTES)

Case	Building Type	Manual (avg)	System (avg)	Reduction
C1	2F Townhouse	53	28	-47%
C2	1F Detached	41	22	-46%
C3	2F Detached	71	44	-38%

## V. CONCLUSION

This paper presented a web application for automated cold-water pipe sizing in residential buildings up to two floors, featuring integrated version control aligned with EIT plumbing standards. The system automates Oungpakorn's six-step pipe-sizing procedure [1], records independently auditable design snapshots per version, and provides side-by-side hydraulic comparison across versions. Testing across 131 cases (81 unit, 40 system, 10 UAT) confirmed calculation accuracy and functional completeness. UAT with three independent sanitary engineers (average satisfaction: 4 out of 5) demonstrated practical usability; Time on Task measurements showed task completion time reductions of 38–47% across three case studies compared to manual methods.

The current system is limited to cold-water supply in two-floor residential buildings. Future work will extend the scope to drainage and hot-water systems, accommodate larger multi-floor buildings, and develop a data-exchange interface with BIM authoring tools (Revit, ArchiCAD) so that validated pipe-sizing results can seed an accurate BIM model automatically, directly addressing the prerequisite identified by Shin et al. [4].

## REFERENCES

- [1] W. Oungpakorn, *Design of Plumbing Systems*, 19th ed. Bangkok: Engineering Institute of Thailand, 2013.
- [2] J. Frazelle, "A new era for mechanical CAD: Time to move forward from decades-old design," *Commun. ACM*, vol. 64, no. 10, pp. 5–16, Sep. 2021, doi: 10.1145/3464909.
- [3] BIMSpaces, "What is BIM & how important is the real estate industry?" *BIMSpaces*, 2021. [Online]. Available: <https://bimspaces.com/blog/what-is-bim-how-important-is-the-real-estate-industry/>
- [4] M. Shin, S. Park, B. Koo, and T. W. Kim, "Automated CAD-to-BIM generation of restroom sanitary plumbing system," *J. Comput. Des. Eng.*, vol. 11, no. 2, pp. 70–84, Feb. 2024, doi: 10.1093/jcde/qwae021.
- [5] T. C. Agwa and T. Çelik, "From barriers to breakthroughs: A deep dive into BIM integration challenge," *Buildings*, vol. 15, no. 7, Art. no. 1116, Mar. 2025, doi: 10.3390/buildings15071116.

- [6] S. Esser, "Graph-based version control of BIM models in an event-driven collaboration environment," *Eng. Proc.*, vol. 17, Art. no. 7, 2022, doi: 10.3390/engproc2022017007.
- [7] K. Cheng, P. Cuvin, A. Olechowski, and S. Zhou, "User perspectives on branching in computer-aided design," *Proc. ACM Hum.-Comput. Interact.*, vol. 7, no. CSCW2, Art. no. 371, Oct. 2023, doi: 10.1145/3610220.
- [8] V. Cristie and S. C. Joyce, "Versioning for parametric design exploration process," *Autom. Constr.*, vol. 129, Art. no. 103802, Sep. 2021, doi: 10.1016/j.autcon.2021.103802