



Title	A User-Friendly Web-Based Interface for Integrated Life-Cycle Cost Analysis and 3D Asset Visualisation in Real Estate Management
Authors(s)	Duncan, Sara, Bampoulas, Adamantios, Hoare, Cathal, Ali, Usman, O'Donnell, James, Mangina, Eleni
Publication date	2024-07-18
Publication information	Duncan, Sara, Adamantios Bampoulas, Cathal Hoare, Usman Ali, James O'Donnell, and Eleni Mangina. "A User-Friendly Web-Based Interface for Integrated Life-Cycle Cost Analysis and 3D Asset Visualisation in Real Estate Management." Springer, July 18, 2024. https://doi.org/10.1007/978-3-031-64359-0_34 .
Series	Communications in Computer and Information Science, 2102
Publisher	Springer
Item record/more information	http://hdl.handle.net/10197/28379
Publisher's version (DOI)	10.1007/978-3-031-64359-0_34

Downloaded 2026-05-01 02:52:05

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)



© Some rights reserved. For more information

A User-Friendly Web-Based Interface for Integrated Life-Cycle Cost Analysis and 3D Asset Visualisation in Real Estate Management

Sarah Duncan
School of Computer Science
University College
Dublin
Dublin, Ireland
sarah.duncan@ucdconnect.ie

Adamantios Bampoulas *
School of Computer Science
University College
Dublin
Dublin, Ireland
adamantios.bampoulas@ucd.ie
*Corresponding author

Cathal Hoare
School of Mechanical
& Materials Eng.
University College
Dublin
Dublin, Ireland
cathal.hoare@ucd.ie

Usman Ali
School of Mechanical &
Materials Eng.
University College Dublin
Dublin, Ireland
usman.ali@ucd.ie

James O'Donnell
School of Mechanical &
Materials Eng.
University College Dublin
Dublin, Ireland
cathal.hoare@ucd.ie

Eleni Mangina
School of Computer Science
University College
University College
Dublin
Dublin, Ireland
eleni.mangina@ucd.ie

Abstract— The operation and maintenance of buildings and facilities present multifaceted challenges, with significant impacts on environmental sustainability and operational costs. Asset managers require up-to-date, pertinent, and comprehensible information to facilitate data-driven decision-making aimed at enhancing the energy and financial efficiency of their facilities. This paper introduces a novel approach by applying life-cycle cost analysis (LCCA) through a web-based knowledge-graph database. This application enables facility managers to make data-driven decisions, expanding the use of LCCA beyond the initial design phase to the operational phase. Unlike existing methodologies, this approach provides a user-friendly web interface that accommodates professionals, prioritises reliability, and optimises performance metrics. The interface includes an asset manager, comparative LCCA, and a 3D model viewer, enhancing interactive exploration of assets and cost data representation. To ascertain the interface effectiveness, feedback was solicited from industry professionals, and telemetry data underwent a thorough assessment using a defined set of metrics. This study fills a critical gap in LCCA software and holds the potential to improve the environmental efficiency of building operations, addressing the global energy crisis.

Keywords— *asset management, data visualisation, energy efficiency, web-based interface, life cycle cost analysis*

I. INTRODUCTION

The operation and maintenance of buildings and facilities constitute a multifaceted and intricate challenge, necessitating meticulous oversight from facility managers. The decisions made by these managers wield significant influence over the environmental footprint and operational expenses of their

facilities [1]. While substantial research efforts have been directed towards sustainable construction techniques, the operational life stage of buildings has received comparatively limited attention. Approximately 70% of all resources consumed by buildings and infrastructure are allocated during their operational life cycle phase [2]. Considering that the built environment accounts for approximately 40% of global energy consumption [3], enhancing the environmental efficiency of building operational phase holds immense potential for addressing the global energy crisis.

However, the transition to digitalised building information has hindered facilities manager access to crucial data rather than facilitating it [4]). Vital data is often compartmentalised in disparate, expensive IT systems or even antiquated paper-based manuals [5]. This study aims to apply life-cycle cost analysis (LCCA) considering a web-based knowledge-graph database. By using the developed application, facility managers can access the information required for making informed, data-driven decisions. LCCA typically employed during the initial design phase, estimates the comprehensive costs of a project, encompassing construction, operational, energy, and eventual disposal expenses [6].

Several studies have developed Life Cycle Cost Analysis (LCCA) methodologies, with a focus on diverse applications. One example is the development of a Building Information Modeling (BIM) integrated relational database management system in [7], which compiles and organises data from BIM models to compute building life-cycle costs. Varae et al. [8] present a probabilistic optimisation methodology for a 3D reinforced concrete structure, incorporating the FEMA-P-58

framework and innovative algorithms. Jalilzadehazhari et al. [9] combine BIM, experimental design, and the analytical hierarchy process for decision-making considering multiple criteria. Baldoni et al. [10] introduce a software tool assessing environmental and cost impacts in building retrofits, integrating life-cycle costing and assessment to evaluate economic and environmental trade-offs. Morales et al. [11] address service life model uncertainties, utilising statistical analysis to highlight variations and stress the importance of selecting suitable models for LCCA. These studies contribute to a broader understanding and application of LCCA methodologies across various domains.

In contrast to the aforementioned research efforts, this study aims to expand the application of LCCA to include operational facilities through a user-friendly web interface. Despite the growing interest in LCCA, there is no mention in the literature of a dedicated software for performing these tasks, with Microsoft Excel being the most commonly mentioned software. The prerequisites for the developed software encompass several critical aspects. Notably, the application should introduce features currently absent in other software or necessitating user-driven implementation. Furthermore, it must be user-friendly, accommodating both LCCA professionals and individuals with limited expertise. Additionally, the software should implement unit tests to guarantee component reliability. Furthermore, it should focus on optimising various performance metrics, such as First Contentful Paint [12], DOM Content Loaded, Time to Interactive [13], Total Blocking Time [14], Largest Contentful Paint [15], and Speed Index [16].

This interface enables end users with an asset manager interface featuring a comparative LCCA and a 3D model viewer for interactive asset exploration. It visually represents integrated cost data, aiding in the comparison of investment alternatives. Visualisation, demonstrated in [17] and [18], is crucial for analysing time series data, offering insights into changes over time. Incorporating data visualisation principles enhances the importance of combining design aesthetics and rigorous scientific approaches [19]. In this study, the user-friendly web interface development prioritises both functionality and visually intuitive representation of integrated cost data. Industry professional feedback through a comprehensive survey and telemetry data assessment ensures interface effectiveness.

II. METHODOLOGY

A. Life Cycle Cost Analysis Calculation

The developed application implements the net present value (NPV) formula as a form of LCCA. The NPV is written as such

$$NPV = \frac{R_t}{(i + i)^t}$$

Where R_t is the net cash flow at time t , i is the discount rate and t is the time of the cash flow. A function that returns a data frame of the NPV at each year of an assets life is written in Python and will enable the user of the application to determine the cost of the potential assets over time considering the discount rate as well as the inflation and maintenance costs.

B. Web Application Implementation

The developed web application is created using Flask. Flask is a lightweight WSGI (Web Server Gateway Interface) web application framework and is characterised by accessibility and scalability [20]. The Flask application contains a Plotly Dash dashboard [21]. Although Plotly Dash can be implemented without wrapping it in Flask, the addition of Flask will allow for easier implementation of any additional software, that is not a part of Plotly Dash. This dashboard is divided into two tabs; the net present value tab and the model tab.

1) *Architectural Overview:* As shown in Fig. 1, the application primarily operates within a Flask app, composed of two key sections: the Python Logic section, responsible for data preprocessing and NPV calculations, and the Plotly Dash section, which furnishes interactive UI components for the application. The Python Logic can be executed within the application or through a dedicated testing program, which facilitates unit tests for NPV calculations, ensuring that any changes to the main program consistently produce expected results. The choice of Unittest as the testing framework is informed by its inclusion in the Python standard library [17], and its suitability for the relatively straightforward unit tests. Plotly Dash is selected as the framework for crafting the dashboard due to its user-friendly data visualisation-oriented components, which are easily customisable using CSS. These components can be added, modified, or removed through the Plotly Dash callback system, offering a swift development process for simple elements while allowing for more intricate interactions through CSS and callbacks. The callback function developed clears graphical components and removes existing dataframes upon clicking the refresh button.

The application comprises two tabs within the Plotly Dash interface: the NPV tab, which interfaces with Python Logic and user input, and the Model Viewer tab, which engages with the IFC viewer. The IFC viewer encompasses a 3D model viewer, utilising the IFC.js library, and a treeview panel that reveals the structural components of the model and enables users to focus on specific structural elements when selected. From the viewer, users can seamlessly pan, rotate, and zoom in and out within the 3D model. It is worth noting that the literature review code is not directly linked to the application but is stored within the same GitLab repository. The application runs within the browser and can be activated by executing the corresponding Python file.

2) *Net Present Value Tab:* The NPV tab comprises two primary sections: a file upload area and an input field for specifying inflation and discount rates. File uploads are restricted to files possessing either the .csv or .xlsx extensions. Upon satisfying all three input requirements, the program dynamically updates to present an integrated table consolidating all uploaded cost data. In cases where multiple rows pertain to a single cost entry, aggregation is performed to yield single comprehensive costing. Beneath the aggregate table, individualised breakdown tables for each cost entry become visible. To the right of the screen, an interactive line chart illustrates the cumulative NPV over the projected lifespan of an item. Below the chart, a drilldown table offers users the choice to select from options such as Cumulative NPV, Cost,

Cost plus Inflation, or NPV Cashflow, in conjunction with the ability to specify the desired year. The table subsequently displays the pertinent figures corresponding to the selected value for each item. Furthermore, a refresh feature is incorporated, affording users the capability to eliminate the currently uploaded files and commence the data input process again. This multifaceted interface facilitates comprehensive financial analysis while maintaining user-friendly functionality.

The NPV Tab serves as the central dashboard of the application, acting as the initial point of interaction for users upon accessing it. This tab is the primary hub for conducting NPV calculations, constituting the core functionality of the application. As users first launch the application, their point of entry is Fig. 2, which includes key elements such as, the tab title, informative guidelines on utilising the project, the discount rate, specifically designed for numerical data entry, and the inflation rate, another crucial input relevant to both NPV calculations and cost plus inflation calculations, also designed for numerical input exclusively. The application also features a Refresh button, primarily intended for refreshing the page. It is worth noting that, when no files are uploaded, this button remains non-functional; nevertheless, it is retained in the layout for the sake of simplicity. The file upload section is designed to accept both CSV and Excel files, accommodating an unlimited number of uploads as long as they adhere to the specified input format. Additionally, clicking on this area triggers an open file prompt, or alternatively, users can drag and drop files into it. Lastly, the Model tab provides users with access to the model tab.

The user is instructed to complete the fields for the discount rate and inflation rate, as well as upload one or more files adhering to a specific format. The application includes a file upload system that accepts files in either comma-separated or Excel format. In these files, the first row serves as the header, comprising the following columns: Item ID, Item Type, CAPEX, Annual Energy, Annual Planned Maintenance Cost, Annual Corrective Maintenance Cost, End of Life Cost, and Lifetime. While Item ID and Item Type can be of any data type, the remaining columns must be interpretable as floating-point numbers or integers. Files can contain multiple rows, which are aggregated to create a single costing. Separate costings should be submitted in distinct files. Subsequently, the callback functions for NPV and figure generation are triggered, altering the application appearance to match the layout depicted in Fig. 3. This specific layout includes a discount rate of 0.08, an inflation rate of 1.03, and two costing files: sample.csv and sampleOptionA.csv.

The application introduces several notable changes to its components: First, the discount rate remains adjustable, allowing users to modify its value and observe the corresponding graph adjustments. Similarly, the inflation rate retains this flexibility. The Refresh button, equipped with the capability to clear any uploaded files, effectively erases their associated tables and figures from the screen. The file upload area remains functional, continuing to accept files and integrate them into the tables and graphs. The Aggregate table provides a comprehensive summary of all available costings, while individual tables for each costing break down the results. The Cumulative NPV Graph is a visual representation of NPV over time, aiding users in understanding cost dynamics. It is

interactive, permitting users to hover over points to view specific costs at a given time. Powered by Plotly Dash, this graph also offers functionalities such as panning, selection, zoom in, zoom out, and the ability to download it as a PNG image. Additionally, the Drill down area enables users to explore four calculations central to NPV: Cost, Cost Plus Inflation, NPV Cashflow, and Cumulative NPV for a chosen year (selected in the adjacent box). These results are presented in a table for user reference and analysis.

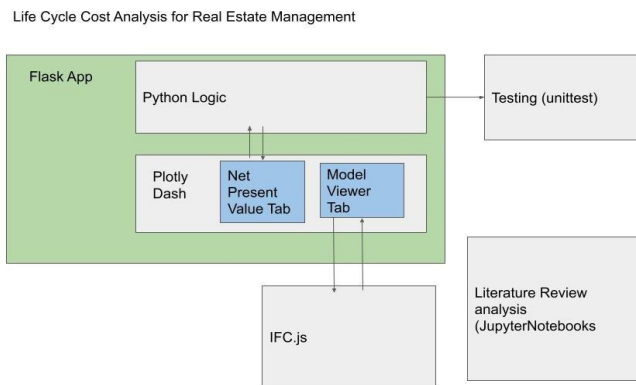


Fig. 1. Architectural diagram of the software.

3) **Model Tab:** The Model tab within the application offers a robust 3D *visualisation* tool, comprising two sections designed to facilitate the exploration of Industry Foundation Classes (IFC) files—standardised formats for building object import/export [22]. Upon selecting an IFC file, the IFC.js library processes it, presenting the 3D model in an embedded iFrame within the Plotly Dash dashboard. Users can navigate and manipulate the model using standard mouse controls, with the TreeView component unveiling the hierarchical structure of the IFC file. Activating buttons associated with components containing position data focuses the model on the relevant object. The Model Viewer, hosted in the second tab, leverages the IFC.js library to parse and render 3D models from IFC files. This dynamic 3D model instantly generates upon file upload, allowing users to manipulate it with standard controls and add/remove planes. To seamlessly integrate the JavaScript feature into the Flask app, the code is placed in a designated folder, ensuring harmonious blending. However, modifications require a separate compilation, posing occasional challenges in debugging.

The initial page features essential components: a tab title, NPV tab, file upload area, empty TreeView, and a 3D model plane. Upon uploading a file, its name appears in the upload area, and the TreeView and 3D model display in the viewing area. Only one IFC file can be viewed at a time, with the upload triggering changes in the file display and Treeview. Users can explore specific elements, like Basic Wall:Exterior-Brick on Block:143590, indicating versatility in inspecting entire buildings or specific aspects like plumbing or individual items. The Model tab provides a comprehensive visual exploration of building structures, facilitating detailed assessments at various levels. Users can delve into specific elements, empowering a nuanced understanding of projects akin to the capacity for detailed evaluations in the NPV tab. The dual sections of the Model tab, powered by the IFC.js library, showcase an

immersive 3D experience that enhances the application's functionality and offers users a powerful tool for inspecting, analysing, and understanding complex building structures.

4) *Testing and Documentation*: Unit testing for the web application is performed using Unittest. Unittest [23] is Python's inbuilt *unittest* testing framework. Unit tests are added to the NPV calculations to ensure that no additional changes to the project interfere with the correct results of these calculations. The project documentation is available on the associated gitlab repository.

C. Evaluation Metrics

The evaluation of the web application is conducted through a combination of methods, including a video demonstration, a survey distributed to professionals who incorporate LCCA into their work, and the collection of telemetry data. The software key features are showcased through a video demonstration, enabling survey respondents to assess the application performance. The report was circulated through the School of Mechanical and Materials Engineering, a social media post on LinkedIn and by reaching out to personal contacts in the industry. The survey comprised ten questions along with an additional comment box. These ten questions were strategically structured, featuring a mix of short-form inquiries with rating scales and long-form queries, offering respondents the flexibility to provide detailed input if they desired. Importantly, none of the questions were mandatory, a deliberate choice intended to accommodate respondents with time constraints, thereby encouraging a wider participation. Some of the questions were designed as ranking queries, employing a scale from 1 to 5, where 1 represented "strongly disagree," and 5 denoted "strongly agree." It is essential to note that participant emails were not collected, ensuring privacy and anonymity. The resulting feedback is instrumental in generating a comprehensive report that can inform post-project activities. The survey encompasses questions on aspects such as user interface readability, usability, and functionality, and it is finalised once the user interface is fully implemented.

The telemetry data collected encompasses various metrics crucial for assessing web page performance. First Contentful Paint (FCP) marks the initial point when users can see content, with an optimal FCP below 1.8 seconds. Time to Interactive measures the time for full interactivity, with an ideal duration below 3.8 seconds. Largest Contentful Paint (LCP) gauges when the largest content becomes visible, aiming for a favourable LCP under 2.5 seconds. Total Blocking Time (TBT) calculates the cumulative duration of page unresponsiveness, with an efficient TBT under 200 milliseconds. Speed Index evaluates the visual content display speed during loading, with a satisfactory index below 3.4 seconds.



Fig. 2. Initial View of The Application.



Fig. 3. Application with input.

III. RESULTS

The evaluation of the application encompassed two primary methods: first, through a survey targeting professionals who employ LCCA in academic or industrial settings, and second, by employing the six metrics detailed in the Methodology section.

A. Survey

The survey conducted for this report specifically targeted professionals who actively incorporate LCCA within their work. This approach offered the distinct advantage of eliciting well-informed feedback from individuals with domain expertise. However, this approach also presented the limitation of a potentially restricted pool of respondents. Currently, the survey includes responses from a total of five participants.

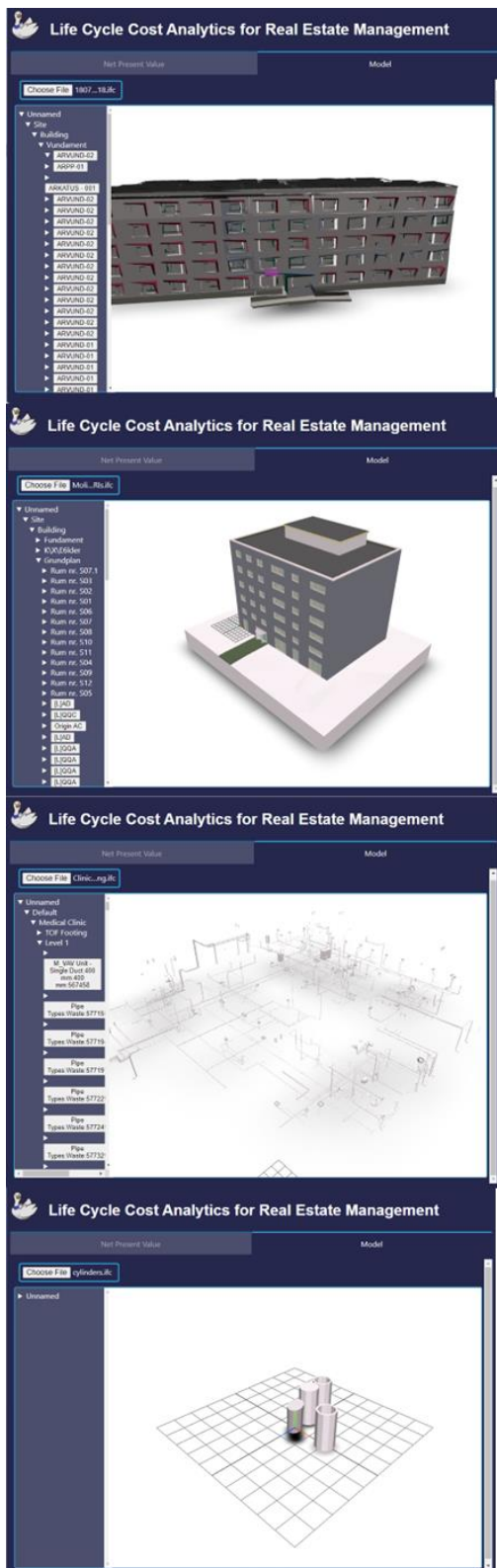


Fig. 4. A Sample of the Variety of Models that the Model Viewer is capable of rendering. a) and b) Show Conventional Models of Buildings, whereas c) shows the plumbing of a clinic and d) is a simple rendering of cylinders at various heights and angles.

1) *Survey Questions:* The survey conducted encompassed ten questions and an open-ended comment section to gather feedback on the developed web application. The respondents, including one professor and three quantity surveyors, primarily represented diverse professional backgrounds, acknowledging the influence of their professions on LCCA utilisation. This contextual information provides a foundation for understanding their perspectives. The survey employed a mix of ranking and open-ended questions to comprehensively assess various aspects of the web application. Notably, the respondents expressed favourable sentiments regarding the application's utility, with 60% agreeing and 40% strongly agreeing that it would be useful to them. The user interface's intuitiveness was evaluated using a ranking scale, despite the initial intention to gather responses from individuals who interacted directly with the software. Despite this limitation, 60% of respondents agreed with the UI's intuitiveness, and 40% strongly concurred. A follow-up question sought to identify any confusing or unintuitive components, revealing that three respondents found the application clear, while one praised its clarity compared to other tools, and another expressed uncertainty about formatting Excel or CSV files.

The presentation of data was assessed positively, with 60% agreeing and 40% strongly agreeing that the data was presented in an easy-to-read manner. Aesthetics were also considered, with 80% agreeing and 20% strongly agreeing that the colour scheme and styling were pleasant. Predictability of component behaviour, a crucial aspect of usability, was explored, and 60% agreed, while 40% strongly concurred. Two respondents reported no surprising behaviours. In terms of an overall rating, 60% gave the web application four stars, and 40% rated it five stars. The respondents' preferences for additional features were also solicited, yielding suggestions such as Extended Reality (XR) integration, overlaying and comparing interest rates, and incorporating features for external factors and civil works. The comment section provided valuable insights, including a suggestion to make the model tab draggable for dual-screen viewing and a recommendation to enhance the sound quality in the video demonstration.

2) *Feedback Analysis:* The feedback received from the survey was largely positive, although a more diverse group of respondents from varied professions would have been ideal. Nonetheless, the survey yielded valuable, actionable feedback. Restricting the survey to those who use LCCA in their professions did limit the size of the survey population, but it resulted in more in-depth and pertinent feedback. Additionally, the survey confirmed the utility of the application and demonstrated a significant level of interest in its development. While feedback regarding the styling was positive, it lagged behind other aspects such as predictability and overall usefulness, possibly signaling a potential area for improvement.

Table 1 summarises the results of the telemetry study. While the majority of metrics fall within the Good category, Largest Contentful Paint is in the Fine category between Good and Poor. Largest Contentful Paint can be improved by determining which component of the application is the Largest Contentful Paint and running diagnostics on the four sub parts of the Largest Contentful Paint: Time to first byte, Load delay, Load time and

Render delay and using this to determine which of these parts requires optimisation.

TABLE I. RESULTS FOR VARIOUS TELEMETRY METRICS

Metric	Poor Threshold	Good Threshold	Result	Classif.
First Contentful Paint	> 3s	< 1.8s	0.178s	Good
Time To Interactive	> 7.3s	< 3.8s	0.51s	Good
Largest Contentful Paint	> 4s	< 2.5s	3.4s	Fine
Total Blocking Time	> 600ms	< 200ms	0s	Good
Speed Index	> 5.8s	< 3.4s	0.6s	Good

IV. DISCUSSION AND FUTURE WORK

This study focused on executing LCCA and visualising IFC models within a dedicated application. The development, coupled with user feedback, revealed opportunities for improvements and future features. The project, currently run locally, could enhance user access by being hosted on a web server. Accessibility improvements are suggested, including making the application screen reader-friendly and incorporating features for keyboard-only usage and captions. While the backend has unit tests, adding frontend GUI testing is recommended for increased reliability and streamlined development. The study advocates for live user testing to better assess usability based on actual user actions, as current feedback relies on video demonstrations and post-survey responses.

User feedback also generated a number of future improvements with varying degrees of complexity. Suggestions for enhancing the LCCA web application include providing detailed instructions for Excel and CSV formatting within the instructional area to improve user-friendliness. While the colour scheme and styling received no specific complaints, its lower ranking suggests a potential need for review. Proposals for integration of Extended Reality with the BIM for immersive 3D exploration and layering different interest rates on the same graph were made, highlighting potential usability improvements. Additionally, enabling the Model tab to be draggable to a separate browser tab was suggested to enhance functionality within the web-based platform.

V. CONCLUSIONS

This study develops a specialised life-cycle cost analysis application, simplifying complex calculations and providing asset managers with enhanced access to their building-generated data. Despite the growing research interest in LCCA in the literature, there is an absence of a dedicated application for its practical use. The resulting application is a web-based tool, offering users the capability to conduct Net Present Value calculations and explore building information models via IFC files. By using this web application, users can upload multiple CSV and Excel files, with their NPV calculations executed automatically. The application generates detailed breakdown tables, interactive graphs, and informative drilldown tables. Additionally, the Model tab enables users to upload IFC files, facilitating visual exploration and navigation through hierarchical tree views. To validate the application utility and

gather valuable feedback, a video demonstration and a ten-question survey were employed. These efforts not only reinforced the necessity for such an application but also provided constructive insights into the accomplished work. Finally, the survey yielded suggestions for future enhancements and developments.

REFERENCES

- [1] N. Wang, K. Wei, K. Sun, "Whole Life Project Management Approach to Sustainability.," *Journal of Management in Engineering*, vol. 30, pp. 246–255, 2014.
- [2] D. Geekiyange, T. Ramachandra, "Significant Factors Influencing Operational and Maintenance (OM) Costs of Commercial Buildings.," *The 7th World Construction Symposium*, Colombo, Sri Lanka, 2018.
- [3] Climate Group, "Built environment.," <https://www.theclimategroup.org/built-environment>
- [4] C. Quinn, A. Z. Shabestari, T. Mistic, S. Gilani, M. Litoiu J.J. McArthur, "Building automation system - BIM integration using a linked data structure.," *Automation in Construction*, vol. 118, pp. 103257, 2020.
- [5] A. G. Mohamed, M. R. Abdallah, M. B. Marzouk, "BIM and semantic web-based maintenance information for existing buildings.," *Automation in Construction*, vol. 116, pp. 103209, 2020.
- [6] S. Fuller, "Life-Cycle Cost Analysis," *Whole Building Design Guide* <https://www.wbdg.org/resources/life-cycle-cost-analysis-lcca>
- [7] H. T. T. Le, V. Likhitrungsilp, N. Yabuki, "A BIM-Integrated Relational Database Management System for Evaluating Building Life-Cycle Costs.," *Engineering Journal*, vol. 24 (2), 2020.
- [8] H. Varaee, A. Shishegaran, M. R. Ghasemi, "The life-cycle cost analysis based on probabilistic optimization using a novel algorithm.," *Journal of Building Engineering*, vol. 43, 2021.
- [9] E. Jalilzadehazhari, A. Vadiiee, P. Johansson, "Achieving a Trade-Off Construction Solution Using BIM, an Optimization Algorithm, and a Multi-Criteria Decision-Making Method.," *Buildings*, 9(4), 81, 2019.
- [10] E. Baldoni, S. Coderoni, E. Di Giuseppe, M. D'Orazio, R. Esposti, G. Maracchini, "A Software Tool for a Stochastic Life Cycle Assessment and Costing of Buildings' Energy Efficiency Measures.," *Sustainability*, 13(14), 7975, 2021.
- [11] M. F. D. Morales, A. Passuello, A. P. Kirchheim, R. J. Ries, "Monte Carlo parameters in modeling service life: Influence on life-cycle assessment.," *Journal of Building Engineering*, vol. 44, 2021.
- [12] First Contentful Paint <https://web.dev/articles/fcp#:text=First>
- [13] Time to Interactive <https://developer.chrome.com/en/docs/lighthouse/performance/interactive/>
- [14] Total Blocking Time <https://developer.chrome.com/docs/lighthouse/performance/lighthouse-total-blocking-time/>
- [15] Largest Contentful Paint <https://developer.chrome.com/docs/lighthouse/performance/lighthouse-largest-contentful-paint/>
- [16] Speed Index <https://developer.chrome.com/docs/lighthouse/performance/speed-index/>
- [17] Y. Shirota, A. Presekal, Riri Fitri Sari, "Visualization of Time Series Data by Statistical Shape Analysis on Fertility Rate and Education in Indonesia.," *Journal of Advances in Information Technology*, 10(2), 2019.
- [18] C. H. Chen, "Research on the Application of Function-Technology-Aesthetics Framework in the Design Knowledge Modelling of Data Visualization.," *Journal of Advances in Information Technology*, 11(1), 2020.
- [19] B. Alkouz Z. Al Aghbari, "Detection and Visualization of Bilingual Trending Topics.," *Journal of Advances in Information Technology*, 11(2), 2020.
- [20] Flask <https://palletsprojects.com/p/flask/>
- [21] Plotly <https://dash.plotly.com/>
- [22] IFC <https://docs.fileformat.com/cad/ifc/>
- [23] Unit testing framework <https://docs.python.org/3/library/unittest.html>