Design and Development of a Web Application for One-Time Location Sharing via QR Code

Napat Theeranitichai[1] and Nuth Otanasap[2]
Master of Management of Digital Technology Innovation Program,
Faculty of Business Administration, Southeast AsiaUniversity
19/1Petchkasem Road, Nongkhangphlu, Nonghkaem, Bangkok 10160
Corresponding author: napat.theeranitichai@gmail.com

Abstract: This study presents the design and development of Song-Loma, a web-based application that enables users to share a temporary location using a one-time QR code. Traditional methods of location sharing, such as manually sending map links or verbally describing directions, often lead to communication errors, inefficiency, and privacy concerns. The objective of this research was to develop a secure, time-limited, and user-friendly system that allows precise location sharing while minimizing the risk of unintended data exposure.

The system was implemented using a full-stack architecture consisting of a Next.js frontend, a .NET 8 Web API backend, and a PostgreSQL database. To address privacy and security requirements, a tokenization mechanism was employed in which each generated QR code contains only a randomized reference token rather than personal or location data. Each token is configured to expire automatically after 15 minutes, ensuring that shared locations can be accessed only within the intended timeframe and cannot be reused after expiration. System functionality was evaluated through a series of technical tests designed to verify accurate token expiration, proper invalidation of expired QR codes, and the prevention of unauthorized reuse across multiple devices and browsers.

User satisfaction and acceptance were assessed through a Likert-scale survey of 57 participants, the Net Promoter Score (NPS), and qualitative feedback collected through open-ended questions. The results indicate a high level of user satisfaction, with an overall mean score of 4.58, and a positive NPS of +44, reflecting strong user acceptance and willingness to recommend the system. The findings confirm all three research hypotheses: the system performs reliably, users respond positively to its design and usability, and user data privacy is effectively preserved. This study demonstrates that temporary, token-based QR code location sharing can serve as a practical, privacy-conscious alternative to traditional location-sharing methods. Recommendations for system enhancement and directions for future research are also discussed.

***Keywords*:** One-Time QR Code, Privacy, Security, Location Sharing, Web Application.

1. Introduction

Sharing a location is now part of daily life, especially when giving directions to drivers. Nevertheless, as numerous studies indicate, even the most common solutions, such as sending map pins or sharing Google Maps links via chat applications, do not always work in the field. According to Church and Oliveira (2013) [1], in messaging applications, users must save the contact as a friend to send a location, which is an issue for taxi or delivery drivers. More recent studies have also demonstrated that Google Maps links can easily be broken when copied and pasted [2] and that long codes or nameless pins are commonly misinterpreted by drivers [3]. All these problems render manual location sharing ineffective, uncomfortable, and even disorienting. To address these issues, the present study proposes Song-Loma, a simple web-based application that enables location sharing via a temporary QR code. The system generates a random token that expires after 15 minutes and cannot be reused, rather than sending a permanent link to a map or Live Location. The aim of the study is to develop the system, conduct user experience testing, and explore the feasibility of a safe, more convenient alternative to conventional location-sharing procedures using temporary QR codes.

2. Related Work

The studies reviewed demonstrate that traditional location sharing is not a well-developed tool due to such issues as a challenging copying of links, incomprehensive long coordinates, usage of chat-apps, and untrustworthy opening of URLs, and none of the existing solutions is applicable to real-life cases. The research papers generally just name these issues and the limited systems suggested have limited contexts like driver navigation or indoor positioning.

Song-Loma fills these gaps by offering:

1. A 15-minute expiry system, which preserves privacy and ensures this data cannot be used again- again, something that all former systems have lacked.
2. A simple QR code that avoids copy/paste issues
3. Sharing without chat apps or contacts
4. A general-purpose design usable in any scenario.

Overall, Song-Loma extends prior work by providing a practical, privacy-focused, and easy-to-use alternative to existing methods.

Table 1. Summary of Related Work and Comparison with Song-Loma

Study / System Name	**Purpose**	**Limitation**	**How Song-Loma Differs**
Church & Oliveira – WhatsApp Messaging Behaviors. (2013) [1]	Study messaging limitations for sharing content	Does not target navigation or temporary links	Removes the need for contacts by allowing location sharing without chat apps.
Arora & Arora — Cross-Platform Location Sharing Issues. (2022) [2]	Examine issues when sharing map links via chat	Identifies problems but offers no solution	Avoids copy-paste errors by using a QR code instead of long URLs.
Balasingam & Prasad — Driver Navigation Difficulties. (2021) [3]	Investigate navigation errors when drivers receive location links	No privacy features; does not support temporary access	Provides an auto-redirect to Google Maps through a single QR scan, offering a clearer flow.
Sadeghi & Kamel — Link Reliability Study. (2023) [4]	Evaluate reliability of shared Google Maps URLs	No method to simplify sharing or protect privacy	Prevents broken or corrupted URLs through a secure server-side token mechanism.
Adusei & Zahadat – Usability Challenges in Mobile Map Apps. (2020) [5]	Evaluate common user errors in location sharing	Does not propose a solution; only analyzes user mistakes	Removes confusing manual steps by replacing link sharing with quick QR scanning.
Kunz et al. – Collaborative Location Sharing Problems. (2019) [6]	Study how users interpret shared map links	No private or temporary-sharing mechanism	Enhances privacy by hiding coordinates and using anonymous tokens.
Zhang et al. – Dynamic QR Navigation for Drivers (2020). [7]	Improve navigation via QR code for transportation	Focuses on drivers, not general users, no location-saving or survey	Improves security with one-time-use QR codes and a short expiration window.

3. Materials and Methods

3.1 System Architecture

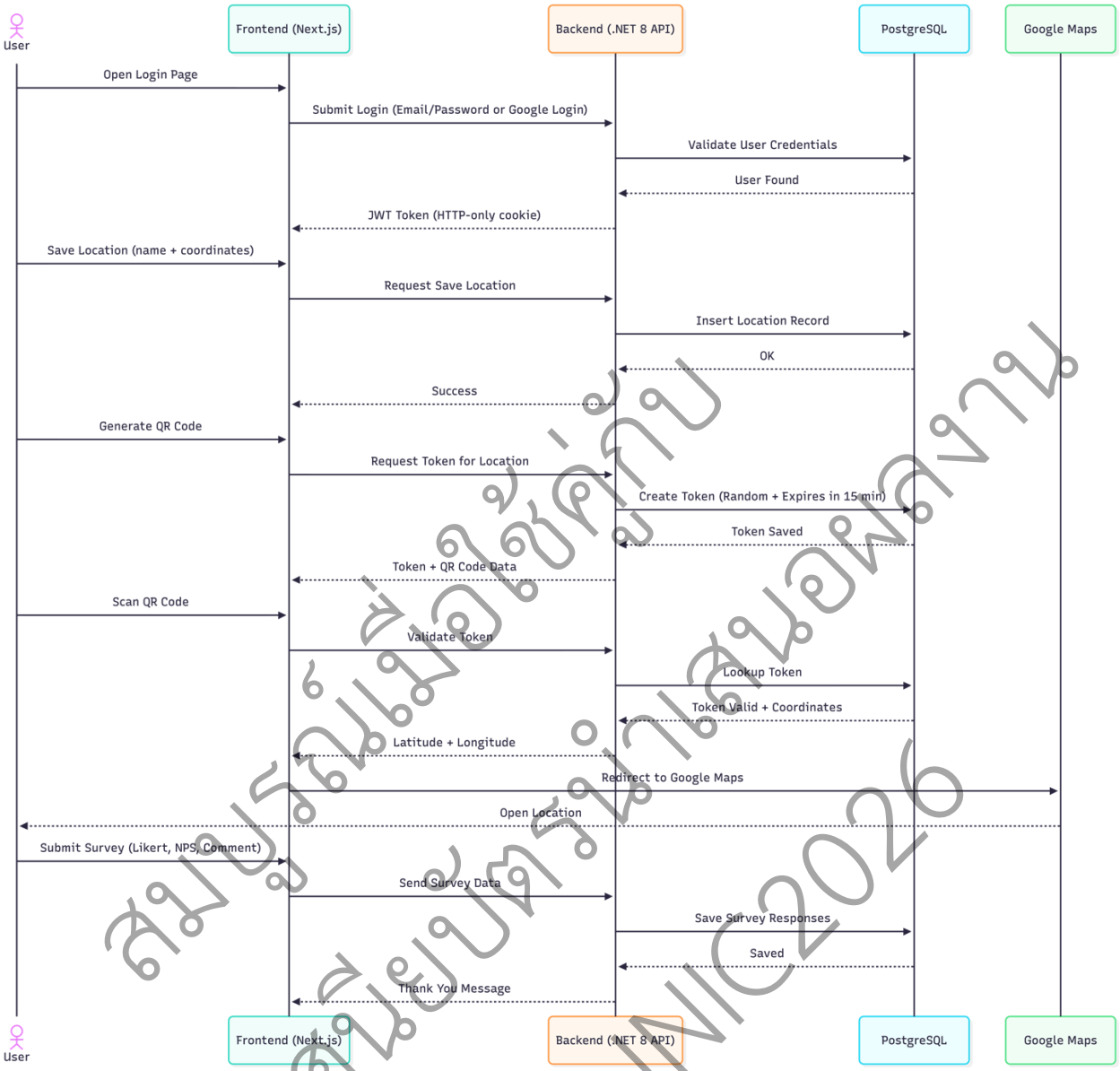


Figure 1.System Architecture Sequence Flow of Song-Loma

This paper has created a complete web application that can create temporary and one-time QR codes to share location securely. The system was created to allow sharing the exact locations of the users without revealing their personal data, without stationary links to the map, or tracking them over time. Each QR code loads a randomized token that is out of band rather than in-band with coordinates, and this prevents access to QR codes before the expiry of 15 minutes.

The architecture of the system is a client–server system. The frontend, developed using Next.js, will manage the choice of location, the creation of QRs, and communication with the user. Business logic, token creation, expiration, authentication, and secure access control are handled in the backend created with the help of .NET 8 Web API. PostgreSQL is the database layer that is highly reliable and integrating in terms of integrity and storage of locations, tokens, expiration timestamps, and survey data.

It is a design that divides presentation, logic, and data layers, making it easy to develop in a modular manner and also secure the communication using RESTful APIs. Good validation, indexing and structured schema design is better in performance and maintainability. A combination of these technologies offers a large-scale, reliable, and effective platform of sharing temporary locations.

3.1.1 Frontend Implementation

Table 2.Frontend Requirements Overview

Category	Features	Requirement Description
Functional Requirements (FR)	FR1 Homepage	Display main actions such as login and location functions.
	FR2 Login	Allow users to log in with email/password or Google Login.
	FR3 Manage Locations	View, create, and update saved locations.
	FR4 Map Selection	Let users pick coordinates using an interactive Google Map.
	FR5 Generate QR	Show QR code with expiration timer.
	FR6 Redirect	After scanning, validate the token and redirect to Google Maps.
	FR7 Survey	Allow users to submit a short satisfaction survey.
	FR8 Error Messages	Display clear messages for errors (expired token, invalid login, etc.).
Non-Functional Requirements (NFR)	NFR1 Responsiveness	Must work smoothly on mobile and desktop.
	NFR2 Usability	Interface must be simple, clear, and easy to understand.
	NFR3 Performance	Pages and maps should load quickly.
	NFR4 Compatibility	Support major browsers and mobile devices.
	NFR5 Security Awareness	No sensitive data stored on the frontend; rely on secure backend validation.

Next.js was selected as the frontend framework due to its high performance, server-side rendering (SSR), and the latest App Router architecture, which facilitates high-speed routing and page switching. Next.js is used to develop production-ready, responsive, and custom applications that include built-in rendering, routing, and API integrations [8].

In the case of Song-Loma, Next.JS is compatible with the Google Maps JavaScript API, which will enable visitors to search and choose the correct position with the help of an integrated map. QR code generation is fairly simple and only requires a few steps that are easy to adhere to, and simple interface elements, such as buttons and forms, ensure that the process is not complicated. The client-side validation makes sure that obligatory fields are filled in correctly and avoids invalid coordinates.

The frontend is designed in accordance with the principles of mobile-first design, and TailwindCSS provides the same layout across devices. There is also clear typography, visual hierarchy, and a few steps of interaction, making the user experience easy and intuitive to technical and non-technical users alike.

3.1.2 UX/UI

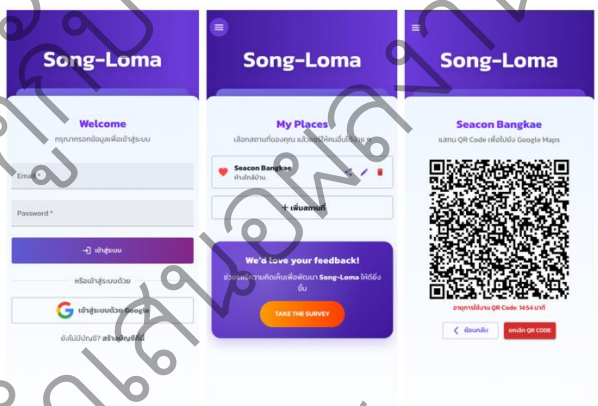


Figure 2. Screenshots of Key System Pages

The interface design and the user experience of the system were also taken into consideration. The interface was maintained to be simple and neat, and actions are labeled easily to minimize the cognitive load and enable customers to decide quickly. The visual hierarchy and spacing allowed users to follow the main steps as choosing a location, creating a QR code, and scanning the shared link. The navigation was simple and needed no instruction because of a familiar pattern of interaction, such as map pin selection and responsive buttons. The behavior of components was consistent, and the design was mobile-first, thus making the smartphone to be used as a common device to scan the QR. These UX and UI choices helped reach these high usability and user satisfaction rates that were reported in the study.

3.1.3 Backend Implementation

Table 3. Backend Requirements Overview

Category	Features	Requirement Description
Functional Requirements (FR)	FR1 Authentication	Support login via email/password or Google.
	FR2 Location Management	Save, view, and delete user locations.
	FR3 QR Token Generation	Generate a one-time QR with a randomized token.
	FR4 Token Expiration	Token must expire after 15 minutes.
	FR5 – QR Redirect	Validate the token and redirect to Google Maps.
	FR6 – Survey Submission	Store Likert, NPS, and comment responses.
Non-Functional Requirements (NFR)	NFR1 – Security	Use HTTPS and JWT for secure access.
	NFR2 – Privacy	QR/URL must contain no personal data.
	NFR3 – Performance	API should respond within an acceptable time.
	NFR4 – Reliability	Expiration and validation must be consistent.
	NFR5 – Usability	QR creation must be simple and quick.
	NFR6 – Compatibility	Work smoothly on mobile and desktop browsers.
	NFR7 – Maintainability	Architecture must be modular and easy to update.

The backend was built on .NET 8 Web API, a modern, high-performance framework for building secure and scalable web services [9]. where all the operations of the system core were performed, which are the generation of QR codes, validation of the token, authentication, the storage of location, the processing of surveys, and the flow of redirects after scanning the QR code. In the production of a QR code, the backend generated a randomized token that was associated with a stored location, and the token expired after 15 minutes to avoid resilient usage of the token.

Authentication was done through email, password, and Google Login. When validated, the backend returned a secure JWT token, which was checked with each request to allow only users with authorization to control their locations or use it to create QR codes.

In the case of QR scanning, the redirect API checked out the token and sent the saved coordinates back, and automatically opened Google Maps to the user. Reusable locations and survey submissions that were acceptable were also stored on the backend. In general, it was the principal controller between authentication, location management, token lifecycle, redirects, and survey handling to provide the secure and unproblematic functioning of the system.

3.1.4 Database

The database used in the system was PostgreSQL, a reliable open-source relational database known for strong data integrity (PostgreSQL Global Development Group, 2024) [10]. Indexing and constraints enabled fast lookups and accurate expiration checks for one-time tokens.

The backend applied validation rules to reject expired or invalid tokens, and PostgreSQL processed these operations consistently across all devices. Overall, it provided a good balance of performance, usability, and data privacy for the system's temporary location-sharing design.

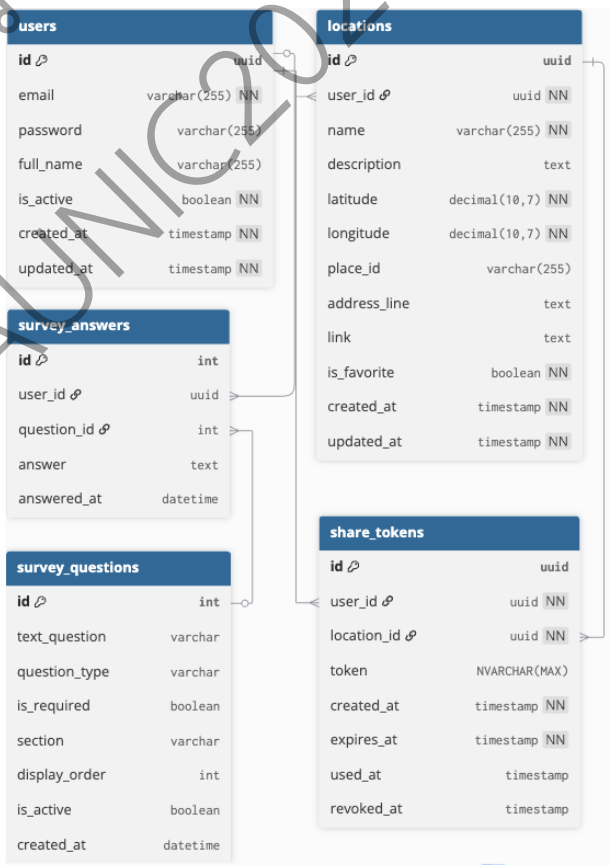


Figure 3. Database Diagram

3.1.5 Deployment Environment

A frontend was deployed on Vercel, a global edge network designed for fast performance and automatic scaling. Continuous deployment was seamless because new updates were automatically built and released after each repository push.

The backend was hosted on an Ubuntu server, a widely used Linux-based operating system known for stability and security in server environments (Canonical, 2024). The .NET 8 Web API ran inside Docker containers, which provide isolated, consistent runtime environments across development and production (Docker, 2023). A reverse proxy was configured to enforce HTTPS and manage incoming API traffic.

PostgreSQL was installed either in the same server or an adjacent secure instance and provided a confident access to data with a strict control of permissions. This deployment configuration was in general an optimized balance strike of performance, security, and maintainability that can be well-suited to a lightweight production-ready application.

3.1.6 CI/CD Pipeline

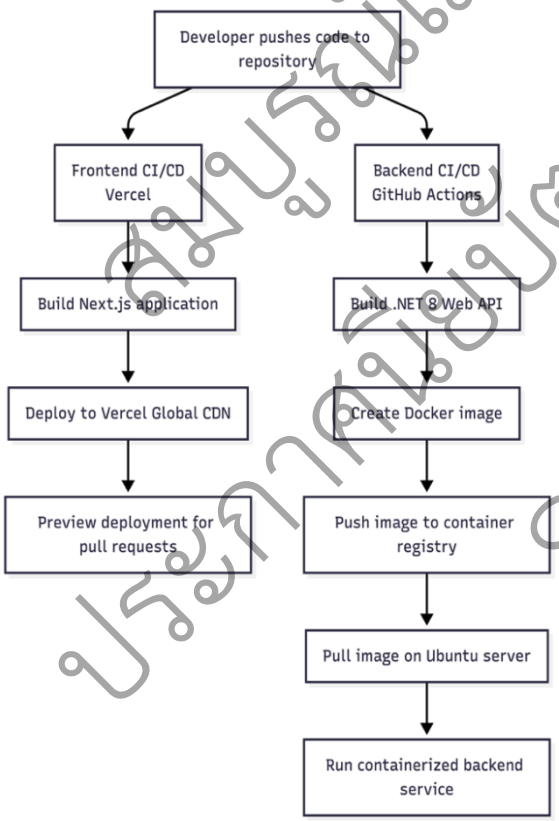


Figure 4. CI/CD Pipeline Flow

Vercel was completely in charge of the frontend CI/CD pipeline. Whenever a new code was sent to the main or deployment branch, Vercel would build the Next.js application automatically and deployment a new version to its global CDN. Each pull request was also created as preview builds, which could be tested and modified in isolated environments before release. This workflow was automated to allow rapid updates without downtime.

In this case, as the backend, CI/CD has been applied through GitHub Actions along with Docker. Every workflow was initiated by a code commit, and it created a Docker image with the.NET 8 Web API and its dependencies and pushed it to a secure container registry. The update image was then able to be pulled via the Ubuntu server to reconstruct the container either manually or automatically. The Docker usage made the environments in development and production similar and minimized the number of problems that are tied to configuration.

3.1.7 Google Maps

The frontend integrates the Google Maps JavaScript API, allowing users to visually select and confirm their location before generating a QR code. For privacy, the system uses a randomized token instead of embedding real coordinates in the QR. Each token automatically expires after 15 minutes, preventing reuse and reducing the risk of unauthorized access.

3.1.8 System Testing Procedures

Table 4. Results of Core System Tests

Test Type	Result
Expiration Accuracy	Passed
Token Invalidation	Passed
Token Reuse Protection	Passed
Privacy Validation	Passed

The core features of Song-Loma were evaluated through system testing to verify correct behavior in real-world conditions. The results confirmed that QR codes expired exactly after 15 minutes, expired tokens were consistently rejected, and tokens could not be reused across different devices or browsers. Privacy validation showed that QR codes contained only randomized tokens without any personal or location data. Overall, the system effectively enforces expiration, prevents reuse, and preserves user privacy as intended.

3.1.9 System Performance Evaluation

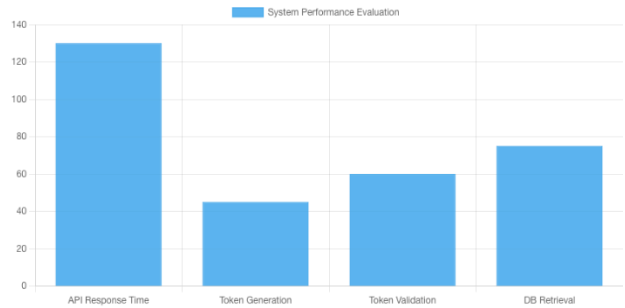


Figure 5. System Performance Evaluation

Song-Loma was performance-tested to ensure QR generation, token validation, and database access were fast and reliable in real use. Grafana K6 [13], a modern open-source load-testing tool, was used to simulate real users and measure backend speed under different traffic levels.

K6 scripts invoked the important API endpoints (QR generation, token validation, and location retrieval) 30 times each to obtain consistent averages of response time, throughput, and errors. The performance was found to be low in terms of latency on all the operations: tokens were generated and validated within very short periods, and database lookups were always fast because of the optimized indexing. On the whole, K6 affirmed that the Song-Loma works quite well, with low response times and stable backend processing, which is appropriate in real-time sharing of location.

3.1.10 Security Configuration

The Song-Loma system involved fundamental security controls to keep sensitive location information and provisional access tokens. Three primary security controls were employed, namely, HTTPS encryption, secure environment variable and strong random token generation. Both frontend and backend were put on HTTPS to avoid data interception and avoid man-in-the-middle attacks. Sources of sensitive values like the database credentials, API keys, and JWT secrets were represented in the protected environment variables rather than in the source code itself. QR codes had cryptographically secure random tokens containing no personal data or coordinates on the inside. All of the tokens expired automatically after 15 minutes, and the backend rejected all the expired, reused, and invalid tokens, which assisted in avoiding unauthorized access.

3.1.11 User Evaluation

Table 5. Age Distribution of Participants

Age Group	Number	Percentage
Below 20 years	0	0.00%
20–29 years	27	47.37%
30–39 years	19	33.33%
40–49 years	5	8.77%
50–59 years	2	3.51%
60 years and above	4	7.02%
Total	57	100%

The user assessment entailed 57 users with friends (10), family members (4), work colleagues (33), and other acquaintances (10). The sample was representative of the age groups as the majority of the users were aged between 20 and 39 years old, with a few aged 40 and above. The participants were put to test the system by creating and reading QR codes under real-use conditions and given structured and unstructured feedback regarding the experience.

The system understanding, ease of use, perceived security, and confidence in the privacy design were evaluated based on a Likert-scale questionnaire. Besides that, the Net Promoter Score (NPS) was also gathered to assess the likelihood of the participants recommending the system to other users, providing a general assessment of the system's acceptance.

3.1.13 Likert-scale questionnaire

A Likert-scale questionnaire [14] is a rating-scale method used to measure a respondent's level of agreement with specific statements. Participants select a score on a fixed scale (commonly 1–5), ranging from strong disagreement to strong agreement, allowing subjective opinions to be converted into numerical data for analysis. To interpret the results, this study applied a standard five-level criterion:

- 1.00–1.80 = Very Low,
- 1.81–2.60 = Low,
- 2.61–3.40 = Moderate,
- 3.41–4.20 = High,
- 4.21–5.00 = Very High.

This interpretation framework helps classify user satisfaction and understanding based on the mean scores obtained.

3.1.14 Net Promoter Score (NPS) Method

The Net Promoter Score (NPS) [15] is the rating of the probability of the users recommending the system to others on a 0-10 scale.

Responses are grouped into:

- Promoters (9-10): Very satisfied and likely to recommend.
- Passives (7-8): Satisfied but not strongly supportive.
- Detractors (0-6): Unsatisfied and unlikely to recommend.

NPS is calculated by subtracting the percentage of detractors from promoters (NPS=%Promoters−%Detractors), giving a simple measure of overall user acceptance.

3.1.14 Qualitative Data Collection

Qualitative information was gathered by using open-ended questions that gave an opportunity to the participants to explain their experiences, concerns, and suggestions. These responses further gave a more insight into the expectations of users and areas of improvement that could not be achieved by quantifying the user satisfaction, and a more comprehensive picture of the user satisfaction and system usability was formed.

4. Results and Discussion

4.1 System Effectiveness Evaluation

The success of the Song-Loma system was measured according to the success to meet the primary objectives of the project, such as secure location sharing one time, strong token expiration, the use of the system in daily life, and user privacy safety.

System testing ensured functional effectiveness because it revealed that tokens would always expire after 15 minutes, were not reusable, and validated by the backend. This proved that the system was dependable to provide temporary and secure location sharing based on QR.

The usability was evaluated with the help of a Likert-scale questionnaire and Net Promoter Score (NPS). The interface was rated as very clear, with a high level of satisfaction with how easily it produces QR codes and the convenience against the traditional way of doing things. These findings showed that users would be able to accomplish tasks fast and with less confusion.

Both the technical checks and user perception supported privacy effectiveness. It was found that the QR codes and URLs included only randomized tokens, no coordinates and personal information, so the intercepted links could not disclose any sensitive information. There was also a rating of improved privacy friendliness of the system compared to live location sharing or fixed map links.

In general, the results indicate that Song-Loma achieves its desired objectives. It is trustworthy, convenient, and has a high level of privacy as it operates in a token-based design that offers superior security and is more practical as compared to the current location sharing solutions.

4.2 User Understanding and Usability

The findings indicate that users were able to grasp the key features of the system, particularly the 15-minute expiration and one-time usage. The temporary QR concept was easily grasped by the majority of participants and they felt safer knowing that the shared location would be removed upon expiration. It was also deemed as a user-friendly interface, simple to use, and the user is able to create and scan QR codes within a few steps. A number of users stated that the system was quick and efficient whenever they were providing directions to drivers or visitors. All in all, users found the simplicity of the system, its convenience, and the design with privacy orientations to be very useful and indicative of how Song-Loma fits in real-life scenarios when one needs to share location quite fast and securely.

4.3 Quantitative Results

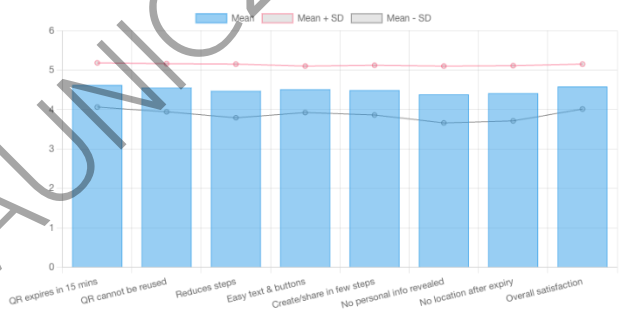


Figure 6. User Response Scores

Quantitative results showed strong positive feedback across all evaluation areas. The average score of the overall satisfaction was 4.58 which means high user approval. Awareness of the 15-minute expiration option had the highest score of 4.62 indicating that users had a clear understanding of the temporary access idea. The Net Promoter Score (NPS) was +44, which implied that the majority of the participants were ready to suggest the system. These results affirm that Song-Loma provides a concise, easy to use and safe user experience that meets the expectations of the user.

Table 6. User Response Scores

Item	Mean ($\bar{x}$)	SD	Interpretation
1. I clearly understand that the QR expires within 15 minutes.	4.62	0.56	Very High
2. The QR code cannot be reused after scanning.	4.55	0.61	Very High
3. The system reduces steps compared to other methods.	4.47	0.68	Very High
4. The text and buttons are easy to understand.	4.51	0.59	Very High
5. I can create and share a QR code within a few steps.	4.49	0.63	Very High
6. The generated link does not reveal personal information.	4.38	0.72	High
7. I am confident the system does not disclose my location after expiration.	4.41	0.70	High
8. Overall, I am satisfied with the system.	4.58	0.57	Very High

Table 7. Net Promoter Score (NPS)

Score Group	Score Range	Number	Percentage
Promoters	9–10	32	56.14%
Passives	7–8	18	31.58%
Detractors	0–6	7	12.28%
Total	–	57	100%

4.4 Quantitative Feedback

Table 8. Summary of Qualitative Feedback

Theme	Example Responses
Issues or Obstacles	“I wasn’t sure at first where the QR would lead.” / “I had a small delay loading the map.” / “There was an error when I tried to delete a saved place.”
Suggestions for Improvement	“It would be helpful to choose the expiration time.” / “I want a preview of the location before sharing.” / “It would be great if the system could pull my saved places from Google Maps.”

The qualitative feedback of the participants was also used to support quantitative results. The majority of the users liked the system as it became convenient, and they liked the time limit of 15 minutes, whereby they felt that their privacy was intact. Another thing that they liked was that the place could be shared without revealing personal data or fixed map links.

Table 7 included common issues and suggestions. There were some minor barriers as described by some users who were either confused about where they would go after scanning the QR, or the slowness of the maps to load, or difficulties when they deleted a saved location. Other suggestions were made, such as support of multiple languages, the ability to change the time to expire, a preview of the location to share it, and the possibility to extract saved places from Google Maps. Such remarks give a realistic advice on how the system can be improved in the future.

5. Conclusion

The results of this study clearly demonstrate that Song-Loma achieved all three objectives of the project. First, for the system development objective, the application successfully generated one-time QR codes that expired within 15 minutes and could not be reused. Technical testing confirmed that the expiration and validation functions worked reliably every time, and the token system ensured temporary and controlled access to shared locations.

Second, regarding user satisfaction, participants showed strong approval of the system. Survey results highlighted that users found the interface simple, easy to understand, and convenient

to use in real situations. The high satisfaction scores and positive NPS further strengthened this result, showing that most users were willing to recommend the system to others.

Third, with respect to data privacy, both testing and user feedback confirmed that the system protected personal information. All QR codes contained only randomized tokens, and no identifying data were exposed. Users also expressed increased confidence knowing that their shared locations disappeared after the time limit.

Overall, Song-Loma succeeded in meeting its goals by providing a fast, secure, and easy method for sharing locations. It proved useful in real scenarios such as directing drivers or helping visitors find a destination. With future enhancements like multilingual support or adjustable expiration settings, the system can become even more practical and widely adopted.

6. Acknowledgement

The author would also like to acknowledge the academic advisor who provided useful advice and support during this Independent Study. The instructors and members of the faculty are also appreciated for the contributions they make to academics. All participants that tried the system and gave useful feedback are also given special thanks as they made a significant contribution to the analysis. Finally, one appreciates colleagues, friends, and family who provided encouragement and support throughout the research process.

7. References

[1] Church, K., & Oliveira, R. (2013). *What's up with WhatsApp? Comparing mobile instant messaging behaviors with traditional SMS.* https://dl.acm.org/doi/10.1145/2493190.2493225

[2] Arora, A., & Arora, S. (2022). *Evaluating cross-platform reliability of shared location URLs in mobile messaging applications.* https://www.igi-global.com/article/evaluating-cross-platform-reliability-of-shared-location-urls/297961

[3] Balasingam, B., & Prasad, K. (2021). *Driver navigation performance using shared map links and coordinate-based locations.* https://www.scirp.org/journal/paperinformation.aspx?paperid=111546

[4] Sadeghi, M., & Kamel, M. (2023). *Reliability analysis of shared Google Maps URLs on mobile communication platforms.* https://www.hindawi.com/journals/misy/2023/5598712/

[5] Adusei, K., & Zahadat, N. (2020). *Usability challenges in mobile map applications: An evaluation of common user errors in location sharing.* International Journal of Human–Computer Interaction, 36(12), 1150–1163. https://www.sciencedirect.com/science/article/pii/S1071581920300124

[6] Kunz, A., Müller, J., & Wichmann, A. (2019). *Challenges in collaborative mobile location sharing: Understanding user errors with map links and coordinates.* Energies (MDPI). https://dl.acm.org/doi/10.1145/3290605.3300294

[7] Zhang, H., Zhao, Y., Li, Z., & Wang, J. (2020). *Dynamic QR codes for secure one-time navigation in transportation scenarios.* IEEE Access, 8, 128920–128932. https://ieeexplore.ieee.org/document/9101879

[8] Vercel. (2025). *Next.js documentation: The React framework for production.* https://nextjs.org/docs

[9] Microsoft. (2024). *Introduction to .NET.* https://learn.microsoft.com/dotnet

[10] PostgreSQL Global Development Group. (2024). *About PostgreSQL.* https://www.postgresql.org/docs/

[11] Canonical. (2024). *Ubuntu Server: Overview and documentation.* https://ubuntu.com/server

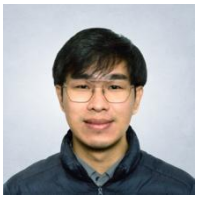
[12] Docker, Inc. (2023). *What is Docker?* https://www.docker.com/resources/what-container

[13] Grafana Labs. (2023). *k6: Open-source load testing tool for performance testing.* https://k6.io/

[14] Likert, R. (1932). *A technique for the measurement of attitudes.* Archives of Psychology, 22(140), 1–55.

[15] Reichheld, F. F. (2003). *The one number you need to grow.* Harvard Business Review. https://hbr.org/2003/12/the-one-number-you-need-to-grow

Authors' Biography:



Napat Theeranitichai
Student of the Master of Management of Digital Technology Innovation Program, Faculty of Business Administration, Southeast Asia University, Bangkok, Thailand.

Asst. Prof. Nuth Otanasap, Ph.D., Management of Digital Technology Innovation Master Program, Southeast Asia University, Bangkok, Thailand.
https://orcid.org/0000-0002-8713-3951