

CS Group 8 TinyBridge Process Update #1

Update

TinyBridge — A Lightweight URL Shortening Service designed for Efficiency and Branding.

1. **Clear requirements documentation ——— Completed:**

The team collaborated to identify and analyze the challenges of long URLs in digital communication (difficult to remember, reduce readability, hinder brand image).

Requirements for a lightweight, customizable, cost-efficient URL shortening service were documented.

2. **Define the technical plan and complete the initial technical document ——— Completed:**

We designed the initial technical plan covering URL shortening algorithms, user login system, customization options, analytics support, and scalability architecture.

3. **Initialization of the project ——— Completed:**

The project environment was set up and initialized. The team adopted Git for collaborative development, and the repository has been created.

Our repository is visible on Github

(<https://github.com/BugUniversity-2024/2025-ITSE-Shortlinker>).

4. **Project development and white paper writing ——— On-going:**

The backend (short link generation, user authentication, analytics) and the frontend (user-friendly web interface) are being developed in parallel.

A detailed white paper introducing the TinyBridge project is currently being drafted to help others better understand and evaluate the system.

Challenges

1. **UI design does not fully align with general user interaction habits — Needs improvement:**

The current UI design has several usability and consistency issues, which may prevent it from meeting modern user-centered design standards and could negatively affect user experience and adoption.

2. **Algorithm design requires further refinement:**

Redirection log recording and utilization analysis: There is still no clear strategy on how frequently to record user redirection logs and what content to include. An efficient approach is needed to balance data completeness with storage overhead. Hot link detection and scheduling: It remains unclear how to efficiently identify trending (hot) links in real time

and pre-allocate resources. Developing or adopting predictive scheduling algorithms will be critical to improving system responsiveness and scalability.

Solutions

1. Adopt systematic UI design using professional tools and references:

We plan to use Figma as the primary platform for UI prototyping, drawing inspiration from leading open-source projects and established design systems. This will help ensure a more consistent, user-friendly, and visually appealing interface.

2. Refine algorithm design with targeted strategies:

For redirection log recording, implement adaptive logging intervals based on access frequency, and include metadata such as timestamp, user agent, and storage utilization for richer analysis.

For hot link prediction, explore lightweight predictive models (e.g., moving average, time-series analysis, or ML-based traffic forecasting) to pre-allocate resources and optimize database writes for popular links.

Next Steps

1. Continue UI development with Figma prototypes and validate design consistency through testing.
2. Develop and evaluate algorithms for redirecting log analysis and hot-link prediction, targeting initial implementation in the next script.
3. Finalize the white paper to clearly explain TinyBridge's purpose, target users, cost estimation, and system architecture.