

# Introduction to Software Engineering Assessment CS

## Group 8 TinyBridge Project Proposal

---

### Title

TinyBridge - A Lightweight URL Shortening Service designed for Efficiency and Branding

### Problem

In the age of online communication, how we present and deliver information has begun to play a significant role in effectiveness, professionalism and success. As we continue to market, educate and socialize via the digital age, the format in which we share links has gone from a technically irrelevant topic to one of extreme importance when it comes to user perception and communication.

Long URLs — with their often-meaningless alphabet soup of random characters and symbols — are hard to remember, take up excess space, reduce readability and in general just get in the way of the efficient transfer of information. These attributes make them a misfit for channels where clarity and brevity is key. Long URLs have no place in SMS notifications, email marketing or any social media post with a character limit.

By contrast, short links have become an essential tool to overcome these challenges. They offer a concise, structured, and visually appealing alternative that improves usability while strengthening brand image. Easily applied across diverse contexts — from social media and email marketing to printed flyers and professional presentations — short links not only enhance clarity and efficiency but also convey a polished and trustworthy impression. In this sense, they are no longer just a technical convenience but a strategic component of modern digital communication.

However, despite the benefits of short links, mainstream platforms frequently create barriers in cost, flexibility, and usability. The most notable issues include:

1. High costs due to subscription-based or usage-based pricing;
2. Limited options for customization and branding;
3. Inconvenient operation with reliance on third-party platform;
4. Scalability and performance instability under heavy use.

Therefore, there is an urgent need for a lightweight, cost-efficient, and customizable URL shortening solution that combines simplicity of use with professional branding capabilities. Such a system should not only reduce the friction of link sharing but also empower users to build a stronger and more consistent brand identity across digital channels, while ensuring stability, scalability, and full user control.

### Solution

We plan to develop a Web application to provide a lightweight, customizable, and cost-effective short link platform TinyBridge to all customers. In order to solve the above problems, we plan to implement the following basic functions on the application.

1. **Cost Control:** Offering a free basic version and flexible tiered packages, supporting private deployment to reduce long-term costs.
2. **Customization & Branding:** Allowing users to bind their own domain names, customize short link prefixes and paths, and display brand information on landing pages. Reading user data and analyzing it.
3. **Convenient Operation:** Creating an independent platform with APIs, supporting batch import/export of links, and enabling users to quickly generate or edit visual landing pages.
4. **Performance Assurance:** Adopting a distributed architecture and multi-layer caching mechanism (in-memory caching with optional Redis support), with real-time system performance monitoring and Bloom filter optimization.
5. **Function Integration:** Integrating URL shortening, QR code generation, visual landing page creation, and comprehensive data analysis.
6. **Platform Features:** Based on a multi-tenant architecture to achieve data isolation, supporting multi-role permission management and personalized settings.

By realizing the above functions, TinyBridge platform will be committed to building a more convenient and reliable platform, and we will also comprehensively adopt various technologies to optimize the user experience, including support for multiple database backends (SQLite, PostgreSQL, MySQL) to ensure flexibility in deployment scenarios.

## Objectives

Objectives 1: Implement user login system, URL shortening and redirection functionality, and build a basic web interface

The core of the goal of this project is to implement the URL shortening and redirection functionality. In order to achieve this goal, we will need to design and implement a web application that allows users to input long URLs and receive shortened versions, as well as redirect users from the shortened URLs to the original long URLs.

The first requirement is to design a secure and user-friendly login system, which will allow users to create accounts, log in, and manage their shortened URLs. This will involve implementing authentication and authorization mechanisms to ensure that only authorized users can access their own data. We plan to use Json Web Token (JWT) for authentication and authorization. JWT is stateless, cross-platform, and easy to use, which makes it a good choice for our web application, it solves the problem of statelessness in RESTful APIs, allowing for scalable and efficient user session management, compared to traditional session-based authentication.

In the process of user password storage, we will use Argon2 to hash passwords before storing them in the database. Argon2 is a modern and secure password hashing algorithm that provides strong protection against brute-force attacks and rainbow table attacks. By using Argon2, we can ensure that user passwords are stored securely and reduce the risk of unauthorized access to user accounts.

Why we need a user login system? The user login system is essential for providing a personalized experience for users, allowing them to manage their own shortened URLs and access additional features such as analytics and customization options. It also helps to ensure the security and privacy of user data, as only authorized users can access their own information.

The second requirement is to implement the URL shortening and redirection functionality. This will involve designing a database schema to store the mapping between long URLs and their corresponding shortened versions and other relevant metadata, as well as implementing the logic to generate unique short codes for each long URL. We plan to use an 8-character random string generation algorithm to create unique short codes for each long URL. This approach uses a character set consisting of alphanumeric characters (a-z, A-Z, 0-9), providing  $62^8$  possible combinations, which ensures uniqueness while maintaining a reasonable short code length. The 8-character length strikes an optimal balance between uniqueness, security, and memorability, making it significantly more secure than shorter alternatives while remaining practical for sharing. By using this random generation approach with collision detection, we can ensure that each shortened URL is unique and cannot be easily predicted or manipulated by malicious users.

The third requirement is to build a basic web interface that allows users to input long URLs, view their shortened URLs, and manage their accounts. This will involve designing a user-friendly interface that is easy to navigate and visually appealing, as well as implementing the necessary front-end functionality to interact with the back-end API. We plan to use Vue.js for building the front-end of our web application. Vue.js is a popular and lightweight JavaScript framework that provides a simple and flexible way to build user interfaces. It is easy to learn and has a large community of developers, which makes it a good choice for our project. The framework offers excellent performance, reactive data binding, and component-based architecture, allowing us to create a responsive and dynamic web interface that provides a seamless user experience.

Overall, the successful implementation of these objectives will require a combination of technical skills, including web development, database design, and security best practices. By achieving these objectives, we will be able to provide a valuable service to users who need a simple and effective way to shorten and manage their URLs.

## Objectives 2: Develop a user-friendly web interface for URL shortening and redirection, implement basic analytics features and improve the performance of the URL redirection service

The second objective is to develop a user-friendly web interface for the URL shortening and redirection service. This will involve designing and implementing a responsive and intuitive front-end that allows users to easily input long URLs, view their shortened URLs, and manage their accounts.

For the front-end pages, we will process them based on the first version of the UI/UX design prototype. The main pages will include a homepage, a user registration and login page, a dashboard for managing shortened URLs, and a settings page for customizing user preferences. This requires a good understanding of UI/UX design principles, as well as proficiency in front-end development technologies such as HTML, CSS, and JavaScript.

For the data analytics features, we will implement basic analytics functionality that allows users to track the performance of their shortened URLs. This will involve designing and implementing a database schema to store analytics data, as well as developing the necessary back-end logic to collect and analyze this data. We plan to support multiple database options including SQLite for development and testing environments, PostgreSQL for production deployments requiring high performance, and MySQL for compatibility with existing infrastructure. This multi-database approach ensures flexibility in deployment scenarios while maintaining consistent functionality across different environments. SQLite provides simplicity for local development, while PostgreSQL offers robust support for data integrity, scalability, and performance in production environments.

For the performance improvement of the URL redirection service, we will implement a sophisticated multi-layer caching mechanism to reduce latency and improve response times. This will involve using an in-memory caching layer as the primary cache, with optional Redis support for distributed caching scenarios, as well as optimizing database queries. Bloom filter is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set. It is particularly useful in scenarios where the set is large and memory usage needs to be minimized. In our URL shortening service, we can use a Bloom filter to quickly check if a shortened URL exists in our database before performing a more expensive database lookup. This can help to reduce latency and improve the overall performance of the URL redirection service by eliminating unnecessary database queries for non-existent URLs.

### Objectives 3: Enhance the web interface with advanced features, including user account management, and detailed analytics

The third objective is to enhance the web interface with advanced features, including user account management, detailed analytics, and landing page customization. This will involve designing and implementing additional front-end functionality that allows users to manage their accounts, view detailed analytics reports, and customize their preferences, as well as implementing the necessary back-end logic to support these features.

For user account management, we will implement features such as team collaboration, role-based access control, and account settings. It also requires the back-end to support these features, including the necessary database schema and API endpoints. This will require a good understanding of user management best practices, as well as proficiency in front-end and back-end development technologies. The team collaboration feature will allow multiple users to work together on link management, with different permission levels ensuring data security and proper access control.

For detailed analytics, we will implement advanced analytics features that allow users to view detailed reports on the performance of their shortened URLs and data visualization features that allow users to easily interpret and analyze their analytics data, including metrics such as click-through rates, geographic distribution of clicks, device type analysis, browser information, and referral sources. Also, it will be easy for users to export their analytics data in various formats, such as CSV or Excel. This comprehensive analytics system will provide valuable insights into user behavior patterns, helping users make data-driven decisions about their link sharing strategies. This will require a good understanding of data analysis and visualization techniques, as well as proficiency in front-end and back-end development technologies.

For landing page customization, we will provide users with practical tools to modify the style and content of their landing pages. Users can upload their own logos, choose different layouts, and edit text or images to match their brand. These features will help users create landing pages that are not only visually attractive, but also consistent with their brand image and marketing needs.

## Benefits

### 1 Click Behavior-Based Recommendation and Enhancing Resource Access Efficiency and User Retention

By mining the correlation of users' click behaviors, the system accurately pushes highly relevant short links and resources (such as frequently accessed links by classmates in the same course) on the homepage. This significantly reduces the time cost for users to find target resources and improves resource discovery

efficiency. Meanwhile, it leverages group behavior data to enhance recommendation accuracy and strengthen users' reliance on the system. Adapts to the resource sharing needs in course scenarios, helping users quickly obtain high-quality content within the same community, reducing ineffective browsing, and improving the practicality of the tool in learning or work scenarios.

2 Convenient and Efficient Social Sharing

The generated web links are compatible with daily social scenarios like WeChat and Douyin. Users can share with one click, easily realizing cross-platform dissemination and interaction of content.

3 Short Code Recommendation

After users input long URLs, the system automatically generates readable 8-character aliases that balance security and memorability. This solves the pain point that original short codes are meaningless and hard to memorize. Meanwhile, it ensures the uniqueness of aliases through advanced collision detection and Bloom filter optimization, reducing the operation cost and memory burden for users to customize aliases.

4 Accurate Data-Driven Decision-Making

Equipped with comprehensive data analysis functions, it can accurately count key dimensional information such as users' geographical distribution, device types, browser preferences, and referral source composition, providing reliable data support for enterprises' market positioning, product iteration, and marketing strategy formulation.

5 Efficient Dissemination Drives Growth

Leveraging the link sharing capabilities of mainstream social ecosystems such as WeChat and Douyin, it significantly lowers the threshold for content to reach target audiences, helping enterprises achieve rapid dissemination of brand information and business content.

Through the benefits, the shortlinkers gives a more convenient and cheap platform for campus and companies in order to organize their links, promote their updating efficiency and save remarkable resources.

Timeline

*Note: Specific dates will be updated based on actual project progress and course requirements*

Phase	Description	End Date
Phase1	Website architecture completed and project initialization	TBD
Phase2	Research on shortening and redirection algorithms	TBD
Phase3	Front and backend separate development	TBD
Phase4	Front and backend separate testing	TBD
Phase5	Front and backend interconnection and software testing	TBD
Phase6	Software deployment and launch	TBD

Action plan

Objectives 1: Implement URL shortening and redirection functionality, build a basic web interface, and user login system.

Action	Assigned to	Deadline	Progress
Requirement analysis and requirement documentation		TBD	Completed
Research on 8-character random string generation algorithms		TBD	In-progress
Design of data model and database table structure		TBD	In-progress
API interface design and documentation		TBD	In-progress
Website architecture design complete		TBD	In-progress

Objectives 2: Develop a user-friendly web interface for URL shortening and redirection, and implement basic analytics features.

Action	Assigned to	Deadline	Progress
Frontend UI/UX design and prototype development		TBD	In-progress
Multi-database support implementation research		TBD	In-progress
Caching mechanism and Bloom filter design		TBD	In-progress
Basic analytics data collection implementation		TBD	In-progress
Performance optimization and testing		TBD	In-progress

Objectives 3: Enhance the web interface with advanced features, including user account management, and detailed analytics.

Action	Assigned to	Deadline	Progress
Project Proposal writing		TBD	Completed
White paper writing		TBD	In-progress
Team collaboration features backend development		TBD	In-progress
Advanced analytics and visualization frontend development		TBD	In-progress
Front and backend interconnection and software testing		TBD	In-progress
Technical document writing		TBD	In-progress
Proof-of-concept software report writing		TBD	In-progress

Action	Assigned to	Deadline	Progress
Software deployment and final presentation		TBD	In-progress