

Assignment Brief: Build a URL Shortener with .NET Core

Author: Vinh Hoang (VinHN7@fe.edu.vn)

Overview

A URL shortener is a tool that converts long URLs into more manageable, shorter versions. This is useful for sharing links on platforms with character limits or improving user experience by reducing clutter. Two popular URL shorteners are Bitly and TinyURL.

In this assignment, you will design and implement a URL shortener using .NET Core. You will learn how to apply the principles of system design and API design to create a scalable, performant, and user-friendly application.

Scenario

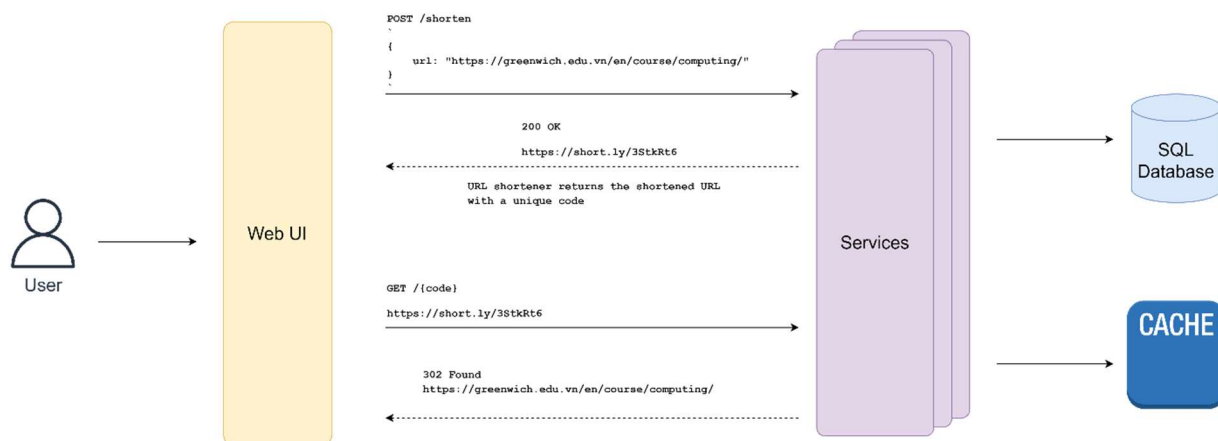
You are a team of software engineers (3 members/team maximum) working for a startup company that wants to launch a new URL-shortening service. Your task is to design and develop the backend system and API for the service. You will also create a simple front-end web application to demonstrate the functionality of your service.

Requirements

Your URL shortener service should have the following features:

- Generate a unique code for a given URL
- Redirect users who access the short link to the original URL
- Validate the input URL and handle errors gracefully
- Store the shortened URLs and their metadata in a database
- Expose a RESTful API for creating and retrieving shortened URLs
- Create a web application that allows users to interact with your service

You can consider the following architecture for the starting point



Deliverables

You will submit the following deliverables:

- **Group Presentation (70%):** You will present your system design and implementation to the class. You will explain the architecture, technologies, and trade-offs of your solution. You will also demonstrate the functionality of your web application. Your presentation should include **slides** and **source code**.
- **Individual Report (30%):** You will write a **self-evaluation report** of the project. You will reflect on your contribution, challenges, and learning outcomes. You will also provide feedback on your team members and the assignment.

Assessment

You will be assessed based on the following criteria:

- **Pass: 5 – 6.5 points**
 - You have completed the basic design and implementation of the URL shortener service. Your service can generate and redirect shortened URLs. Your code is clean, readable, and follows the .NET conventions. Your application must contain Unit Testing and contain a simple web UI (Vue – React) to interact with your system. Your presentation and report are clear and concise.
- **Merit: 7 – 8.5 points**
 - You have added caching to your service to improve the read performance. You have used a distributed cache like Redis to store the frequently accessed URLs. You have explained the benefits and drawbacks of caching in your presentation and report. Your project must have Integration Testing.
- **Distinction: 9 – 10 points**
 - You have implemented microservices architecture for your service. You have separated the users' service from the URL shortening service. You have used a message broker like RabbitMQ to communicate between the services. You have discussed the advantages and challenges of microservices in your presentation and report.