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 versions2. This is useful for sharing links on platforms with character limits or improving user experience by reducing clutter3. Two popular URL shorteners are Bitly and TinyURL4.

In this assignment, you will design and implement a URL shortener using .NET Core1. You will learn how to apply the principles of system design and API design to create a scalable, performant, and user-friendly application.

Scenario

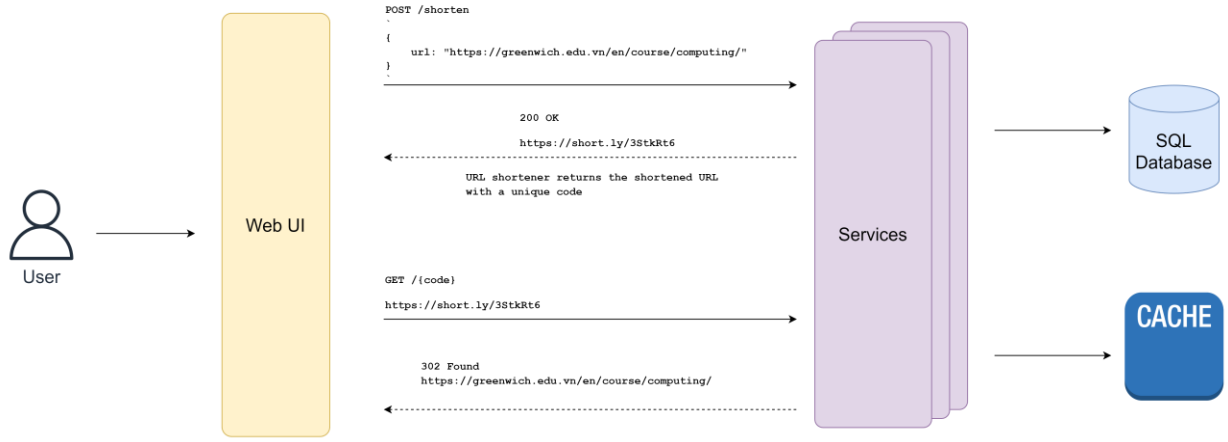
You are a team of software engineers (3 members/team maximum) working for a startup company that wants to launch a new URL-shortening service. Your task is to design and develop the backend system and API for the service. You will also create a simple front-end web application to demonstrate the functionality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

To pass this course, you must achieve **at least 5 on average.**

You will be assessed based on the following criteria:

Group Presentation

MARK	CRITERIAS
9 - 10	Exceptional presentation with a thorough understanding, highly organized structure, and seamless transitions. Technical depth is exceptional, showcasing mastery. The demonstration of functionality is flawless, showcasing all aspects effectively. The source code exhibits excellent practices, being clean, well-documented, and following best practices.
7 – 8.5	The presentation reflects a solid understanding with clear organization and logical flow. Technical depth is evident with clear explanations, and the demonstration of functionality is successful with minor issues. The source code adheres to coding standards with few minor bugs.
5 – 6.5	The presentation demonstrates a basic understanding but with significant gaps. The structure is somewhat organized, but clarity issues persist. Technical details are basic, with inconsistencies. The demonstration of functionality is limited, and the source code shows adherence to basic standards but with some issues.
< 5	The presentation lacks understanding, coherence, and technical depth. Content is superficial, structure is disorganized, and technical explanations are flawed. The demonstration of functionality is ineffective, and the source code quality is poor.

Individual Report

MARK	CRITERIAS
9 - 10	An exceptional report with a thorough and insightful reflection on personal contribution, challenges, and learning outcomes. Feedback on team members and the assignment is detailed, constructive, and specific, displaying a high level of self-awareness and critical evaluation.
7 – 8.5	The report reflects a solid understanding with a reasonable reflection on personal contribution, challenges, and learning outcomes. Feedback on team members and the assignment is constructive and provides some specificity.

5 – 6.5	The report shows a basic understanding but with significant gaps. Reflection on personal contribution, challenges, and learning outcomes is limited and lacks depth. Feedback on team members and the assignment is basic and lacks specificity.
< 5	The individual report lacks depth and understanding. Reflection on personal contribution, challenges, and learning outcomes is superficial. Limited or no feedback is provided on team members and the assignment.