Project: **URL Shortener**

Design Document

Version 1.1

Developer: Subhajit Das

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 16/02/2019 | 1.0 | Design Document of URL Shorten Project. | Subhajit Das |
| 16/02/2019 | 1.1 | Mention order of execution of tests. | Subhajit Das |

Table of Contents

1. Introduction of Design Document
2. Entity Relationship Diagram (ERD)
3. Sequence Diagrams
4. Architecture Design Diagram
5. Class Diagram
6. Interface Design
7. Test Cases

1. Introduction of Design Document

The detailed study about different operations that are performed by a system and the relationship between them within or outside of the system is called Analysis of a system or the given project.

**Objective of Design Document:**

The main objective of this design is to analyse and understanding the system in detail. In this way the features and constitute parts of at least feasible solution are identified and then documented.

**Road Map for Projects:**

The design activity provides a roadmap to progressively transform the requirements through a number on stages into the final product by describing the structure of the system to be implemented.

**Activities Used in This Design Document:**

Different activities that are used in this document are briefly explained below.

**Entity Relationship Diagram (ERD):**

It is graphical representation of entities and their relationships to each other.

**Sequence Diagrams:**

It is a most popular kind Interaction Diagram. It is helpful in both when checking requirements and when checking designs.

**Architecture Design Diagram:**

This describes the layered Architectural model for the software system.

**Class Diagram:**

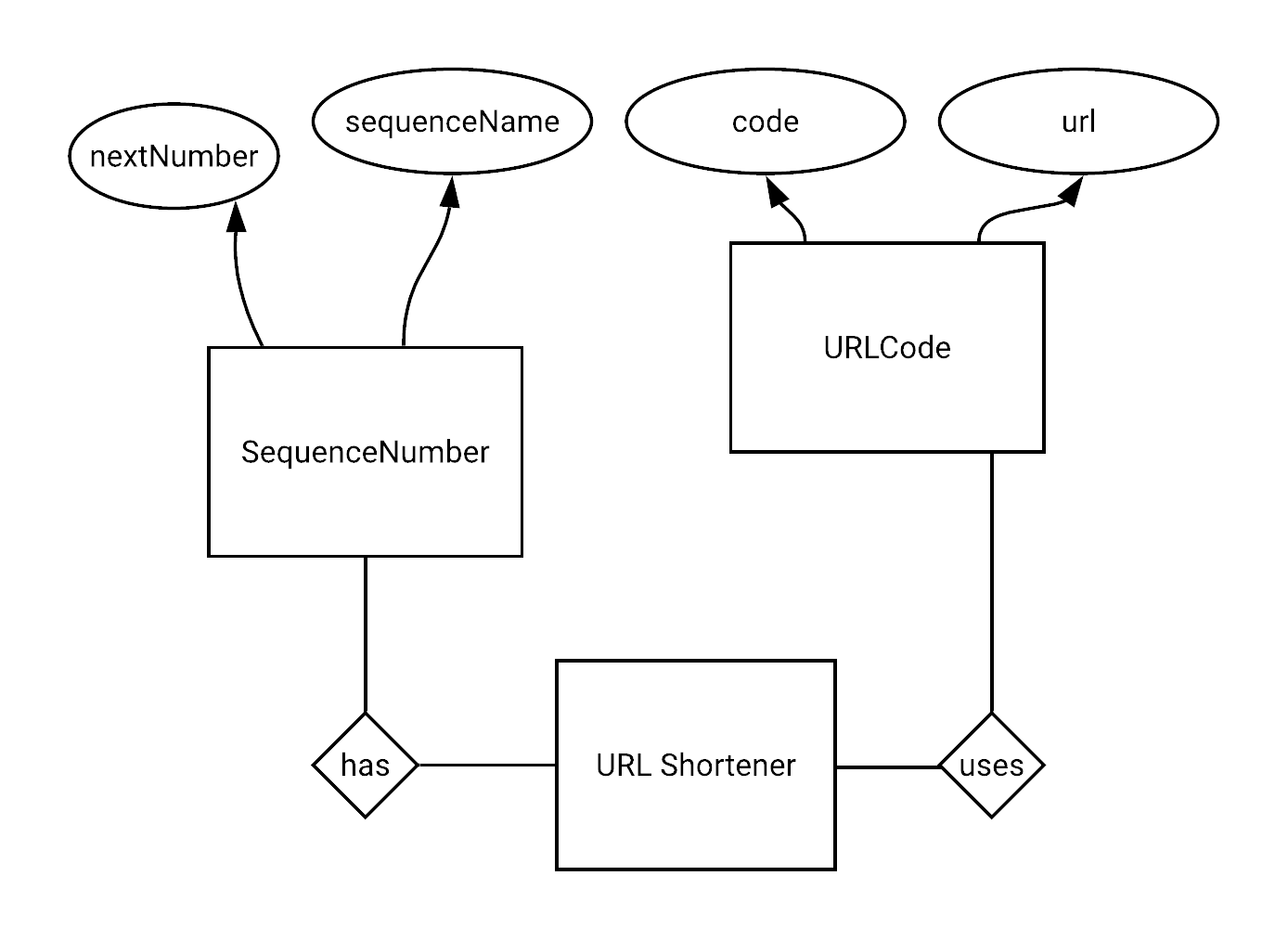
This describes the system by showing the classes, their attributes and relationships among objects etc.

**Interface Design:**

This design shows the important features of a product.

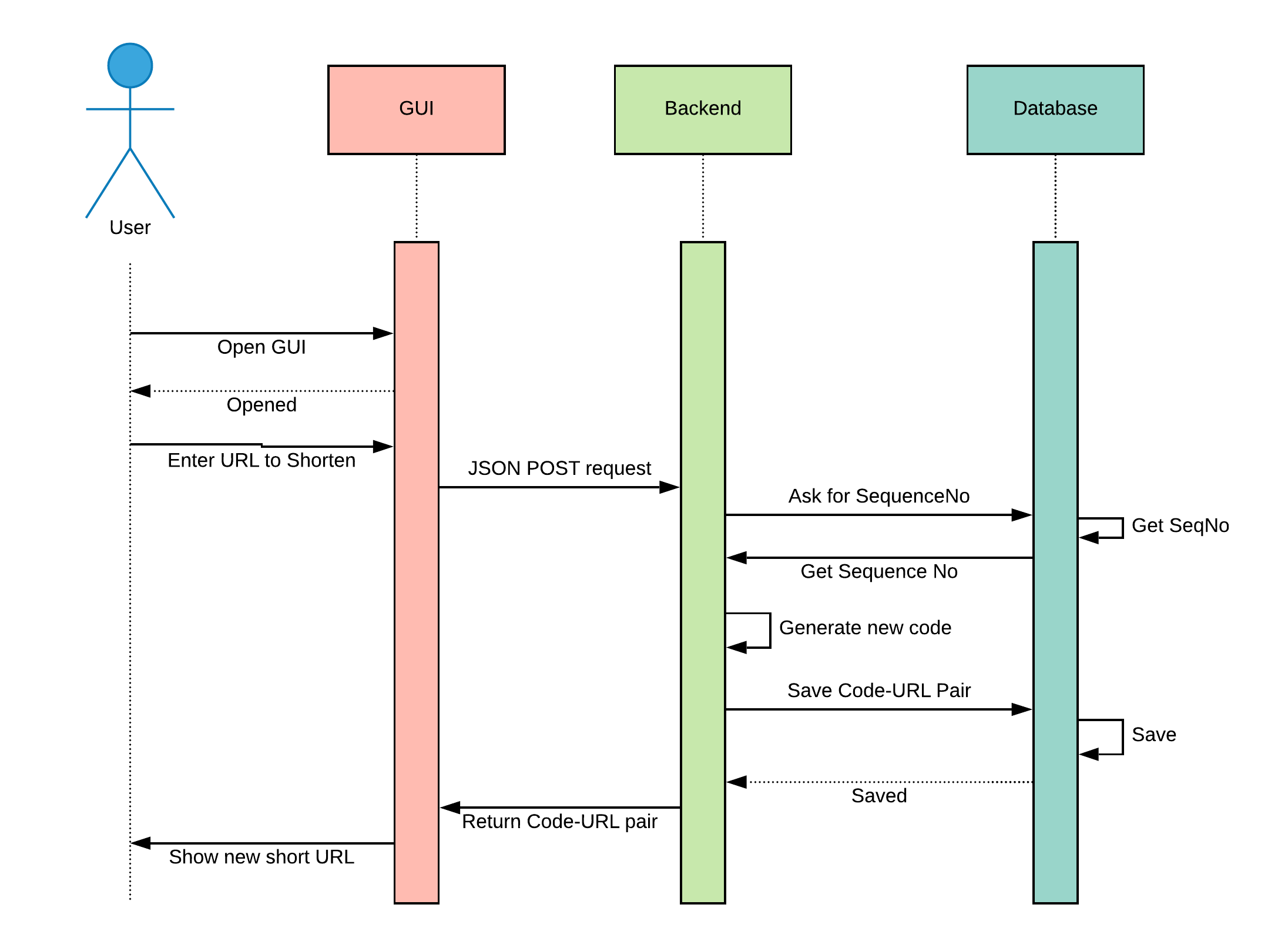
**Test Cases:** Finally test cases are provided in detail.

1. Entity Relationship Diagram (ERD)

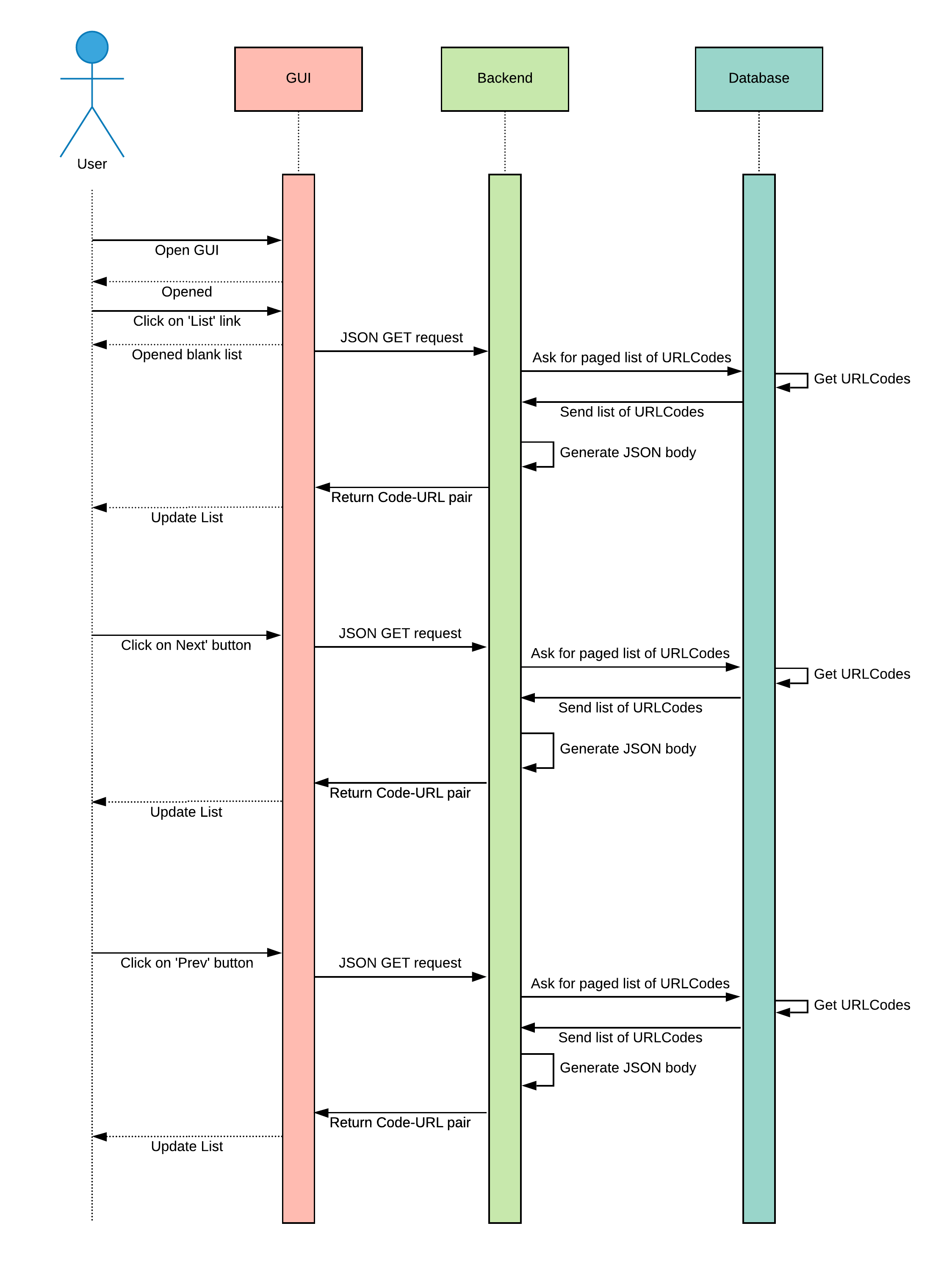


1. Sequence Diagrams

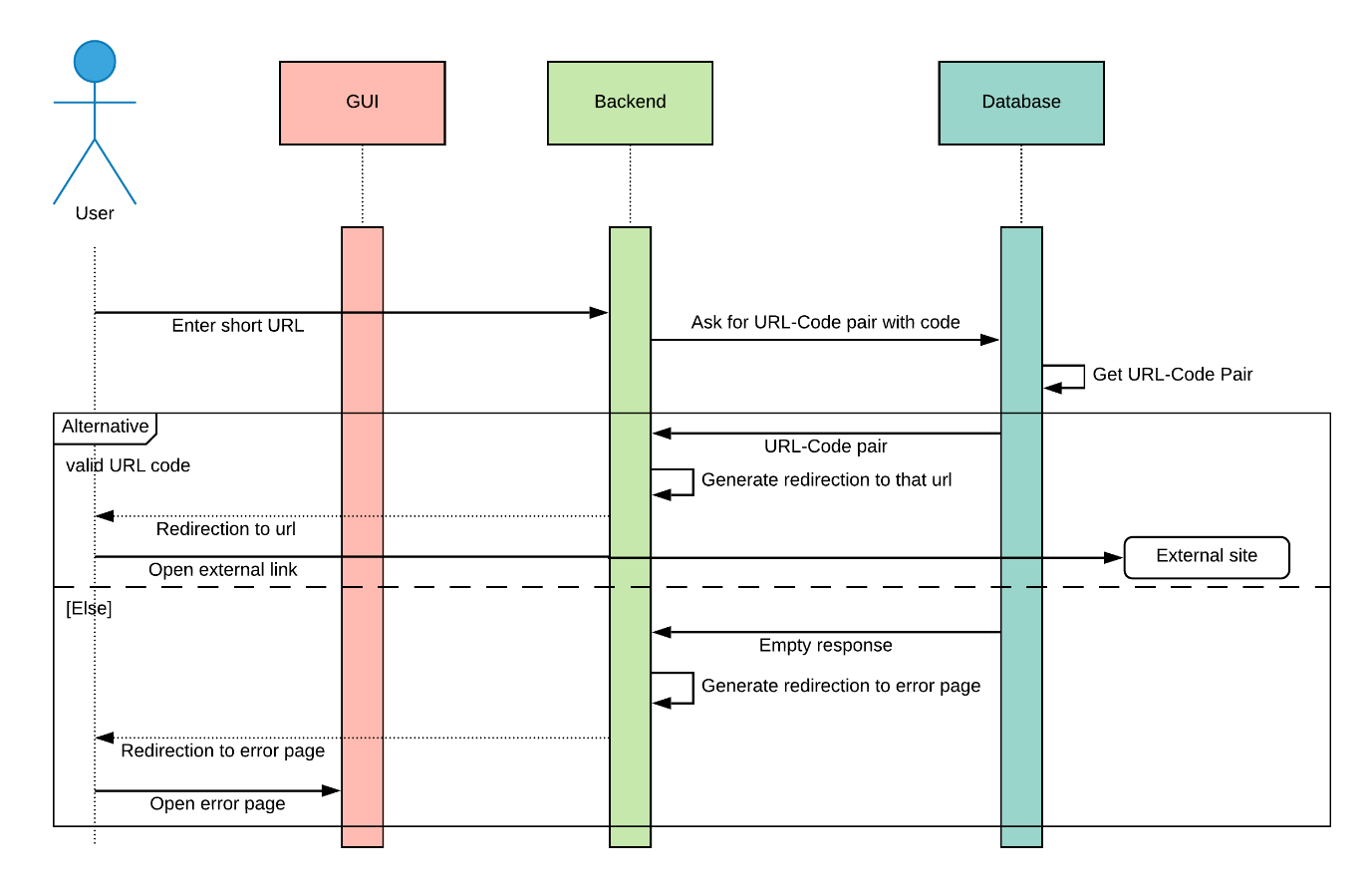
1. **Add/Shorten URL:**



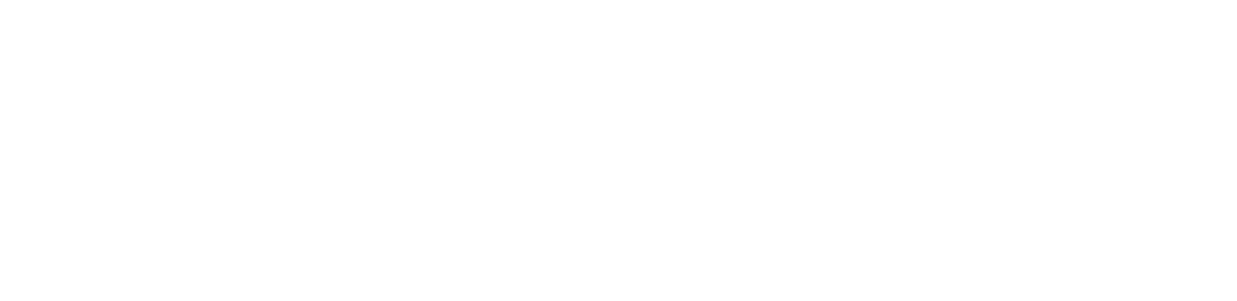
1. **List All URLs:**



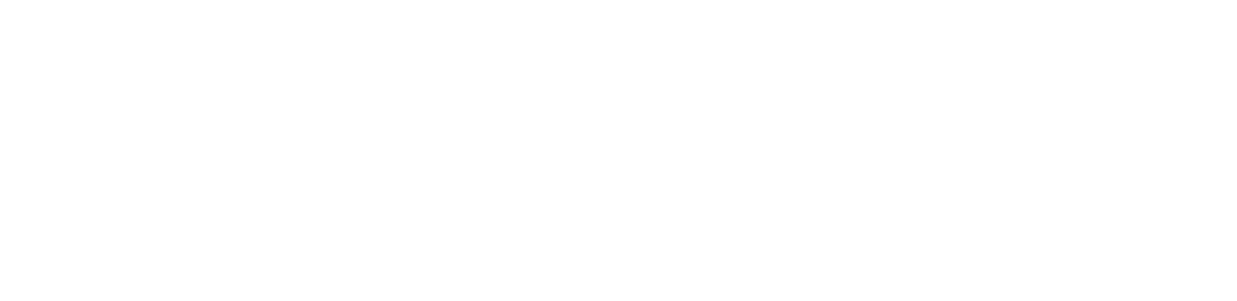
1. **Redirection:**



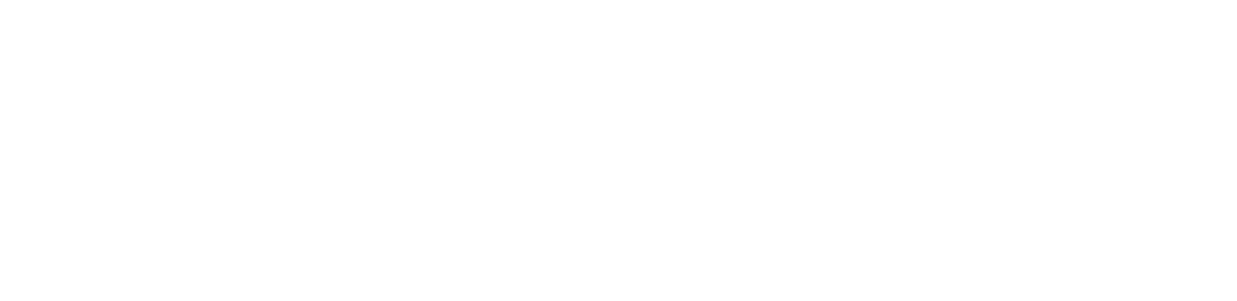
1. Architecture Design Diagram



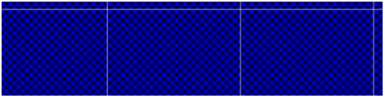
**Presentation Layer**



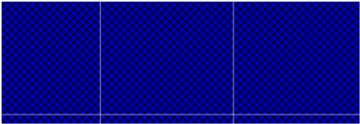
**Application Layer**



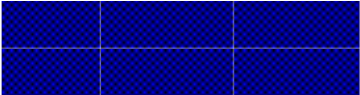
**Database Layer**



**Web GUI**

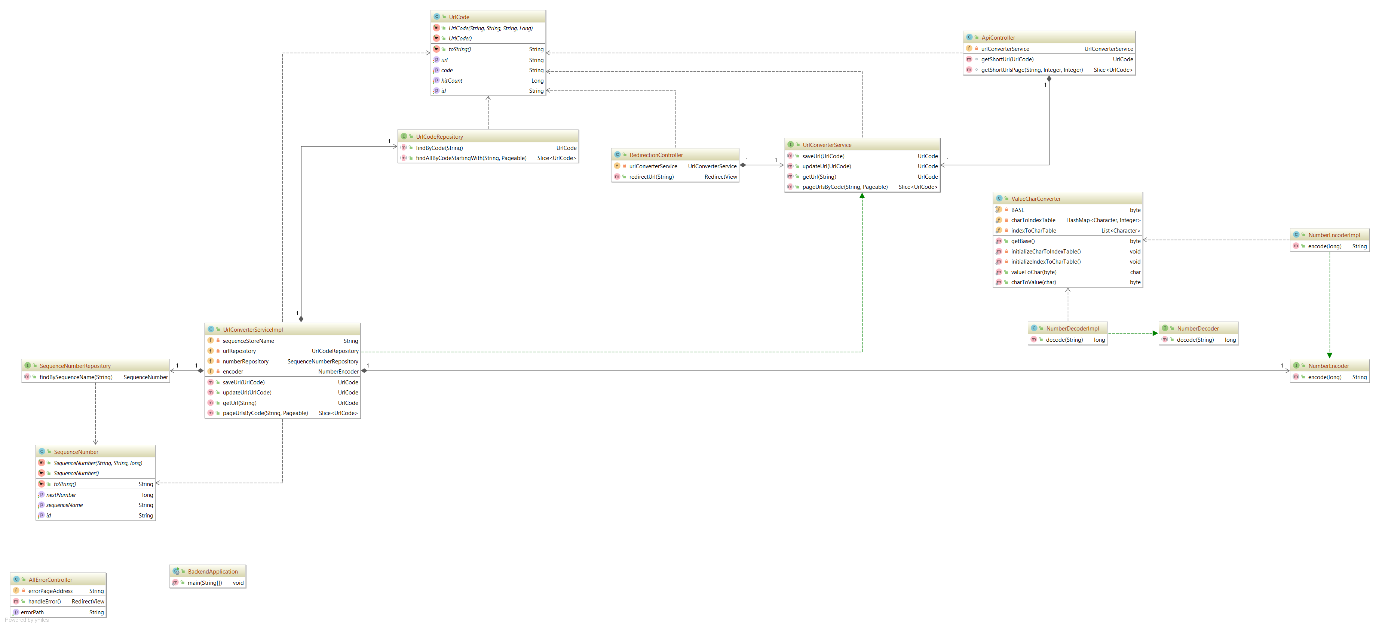


**REST backend**

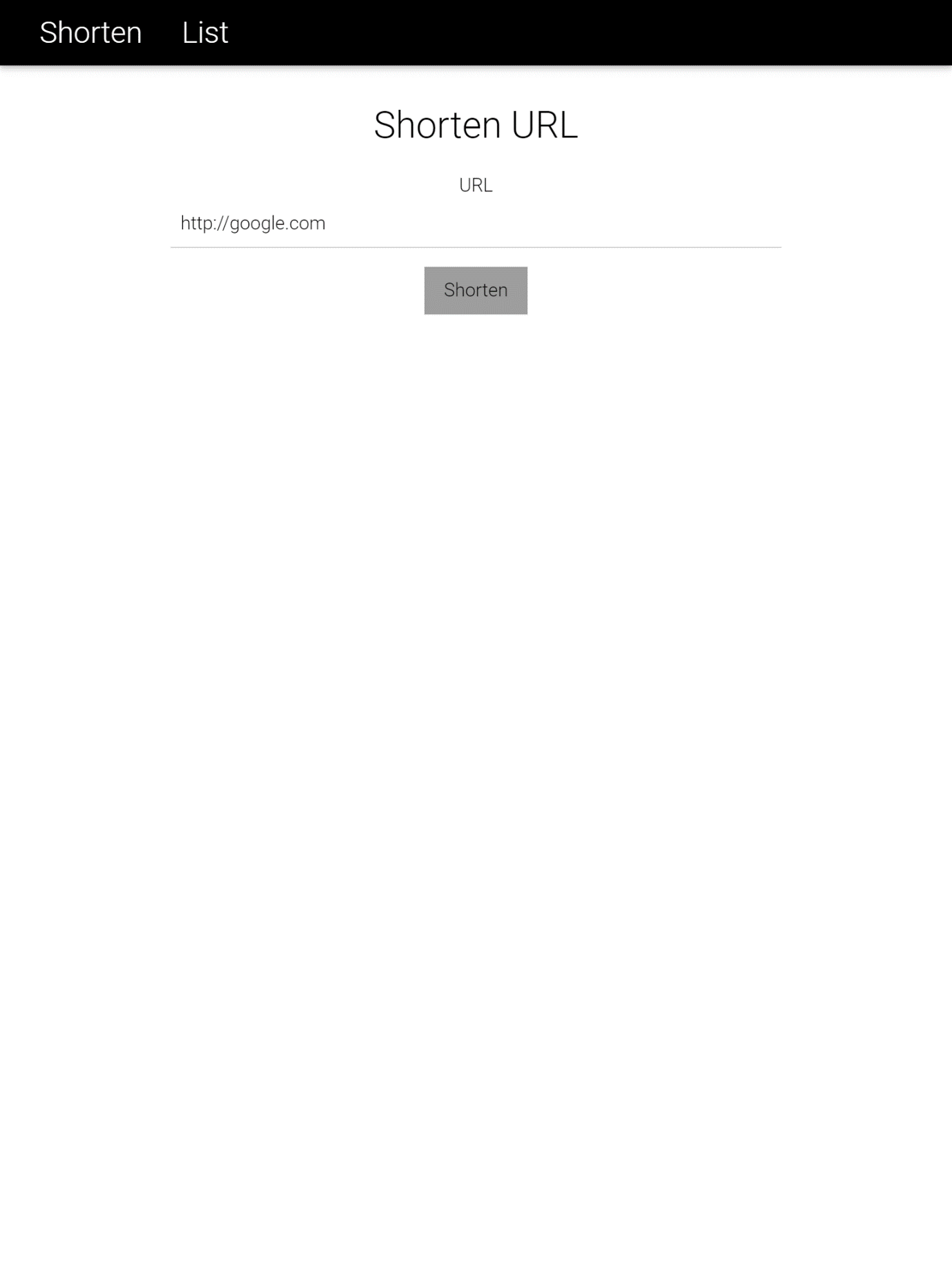


**Eatable DB**

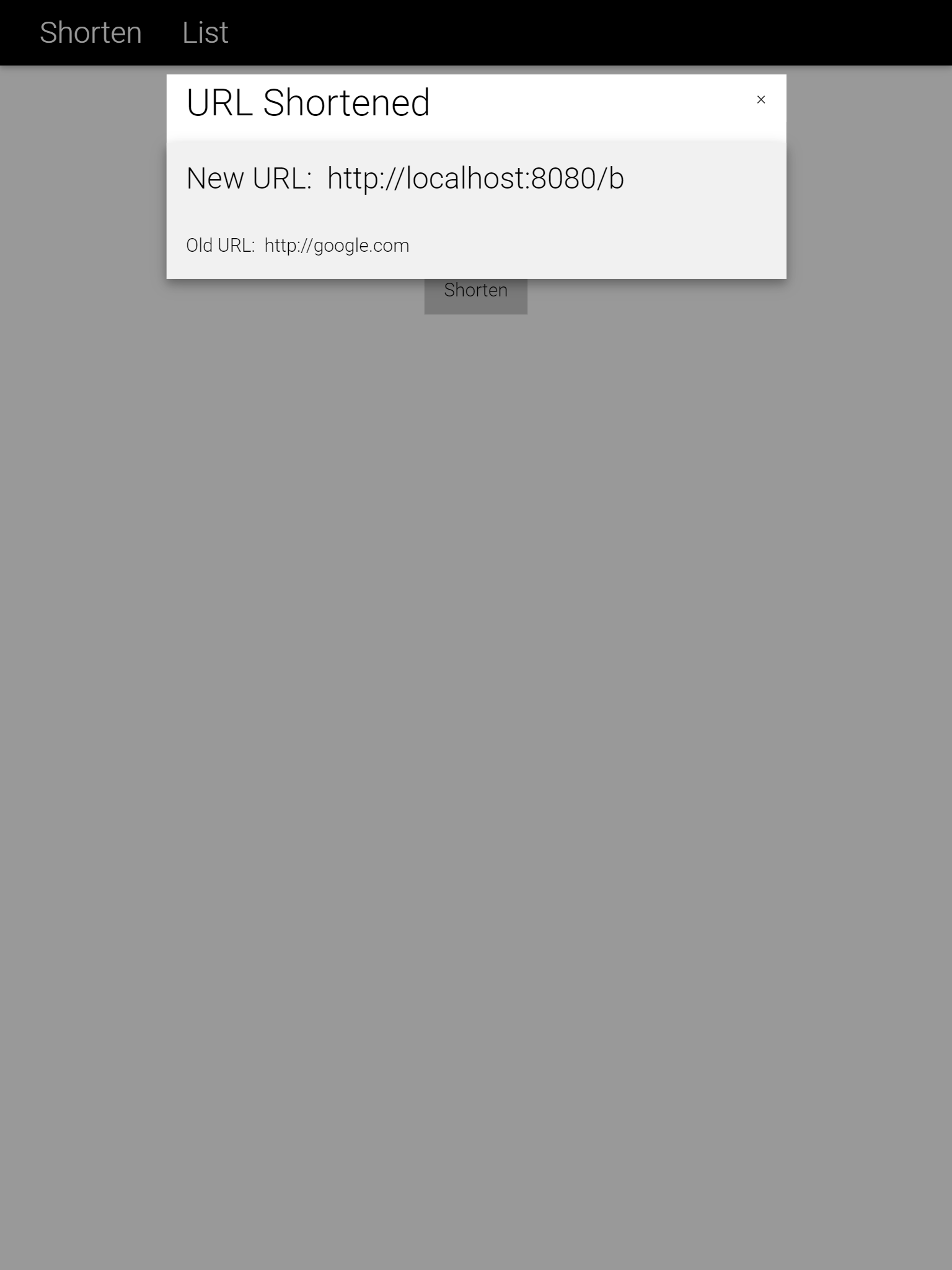
1. Class Diagrams



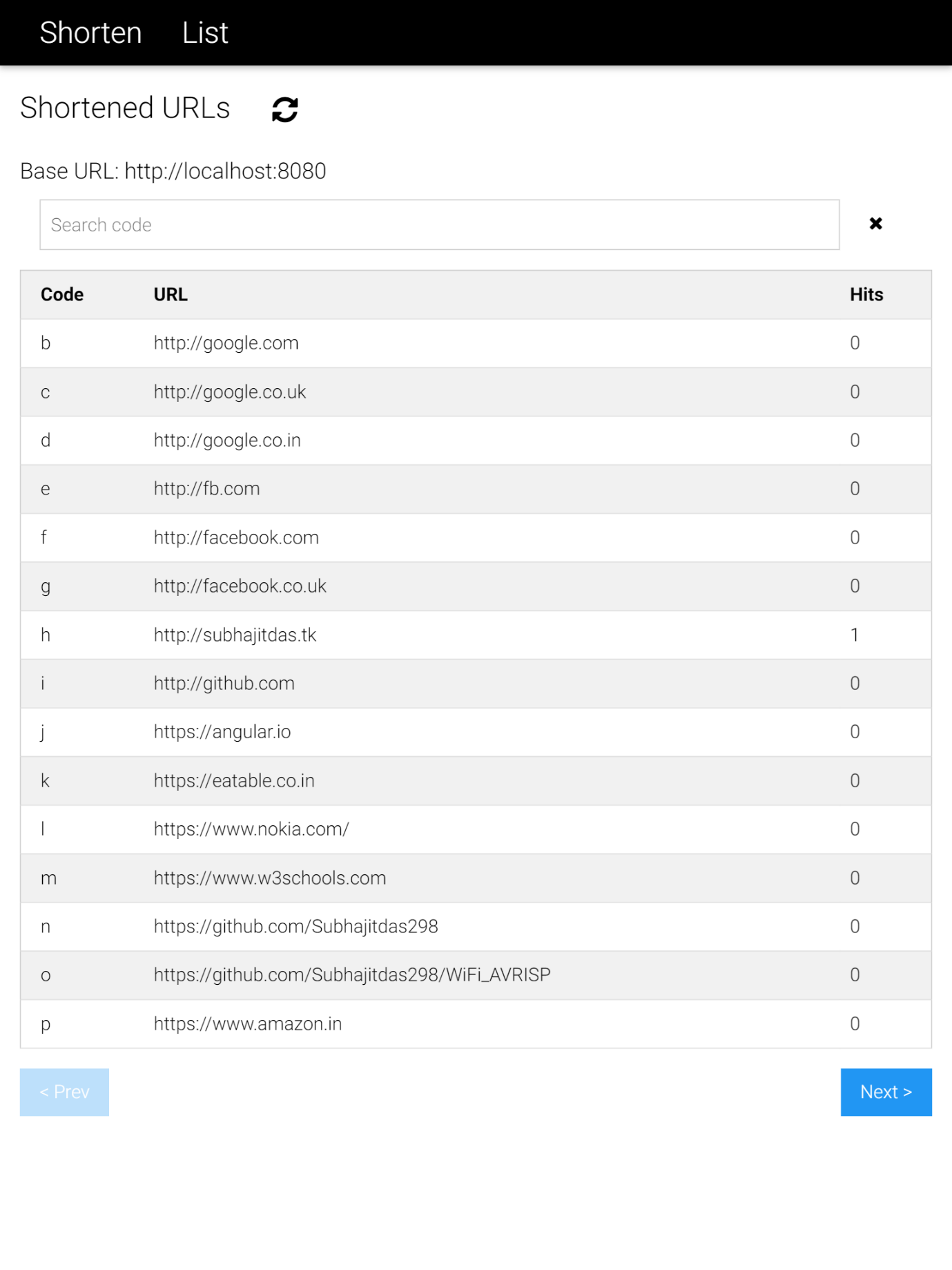
1. Interface Design



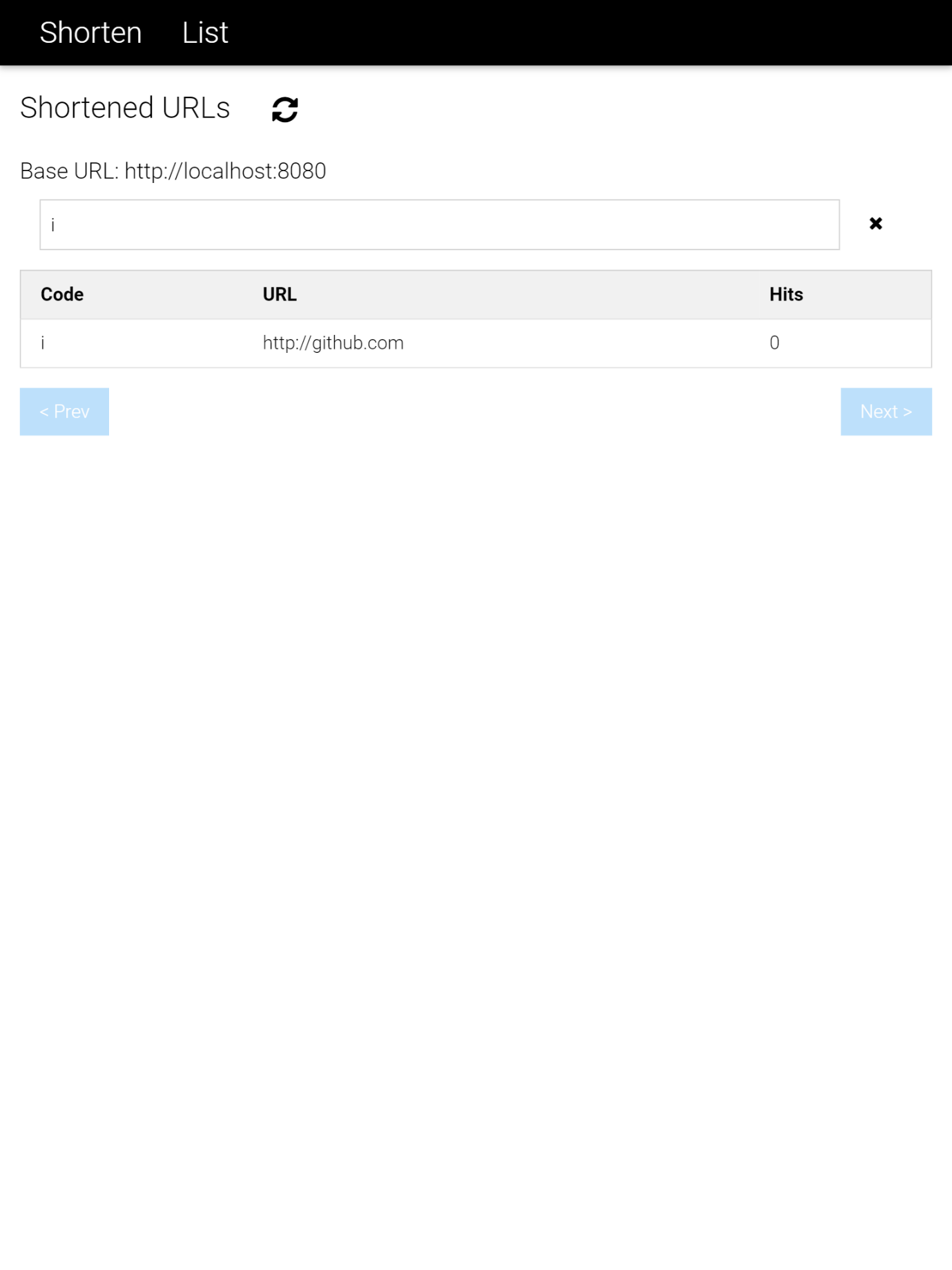
Shorten/Add URL



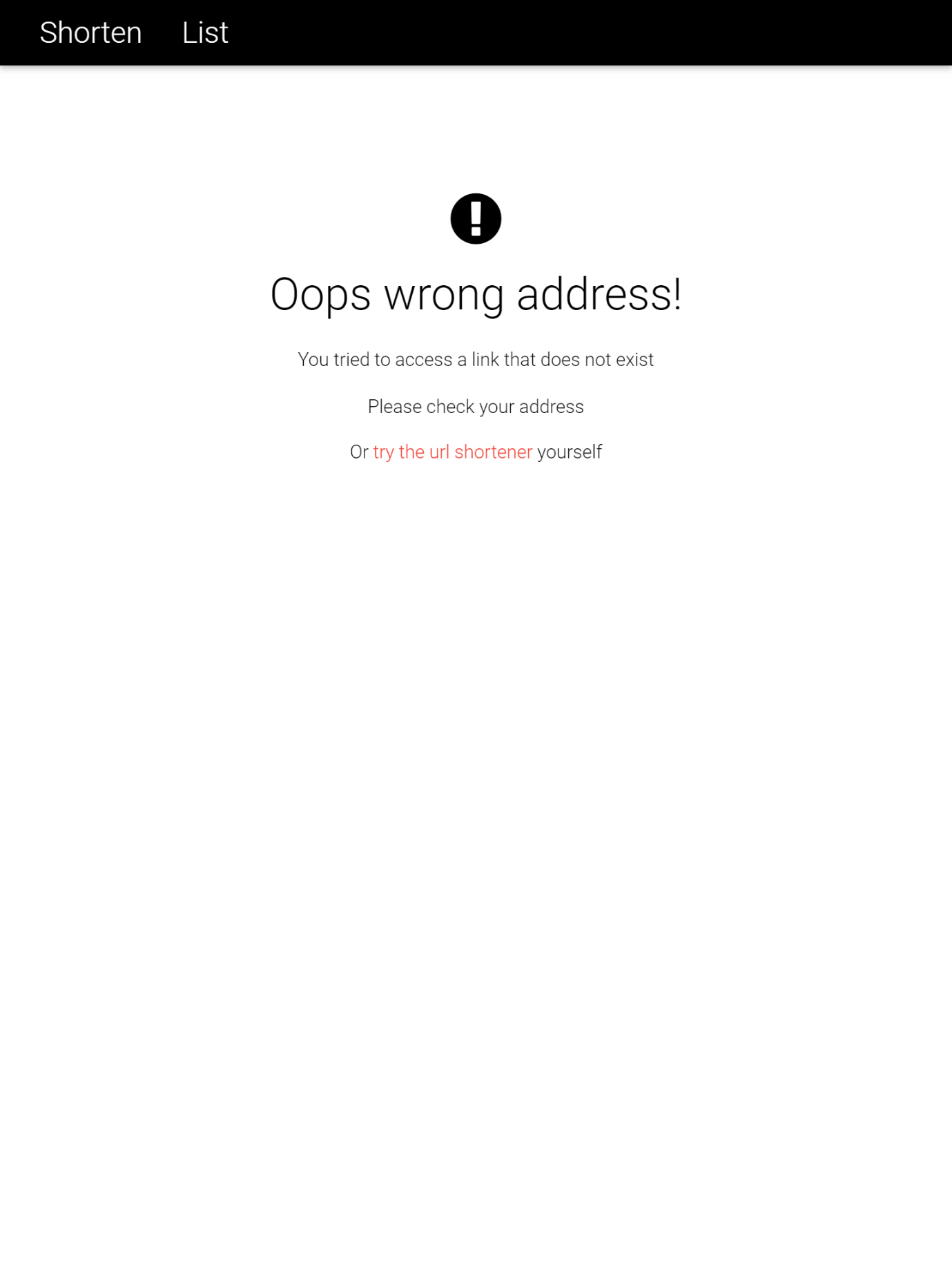
Generated short URL



List URLs



Search List URLs



Error screen

1. Test Cases

**Note:** It is to be noted that the backend and frontend have their own automated testing set up.

For backend, a fresh build or test tasks of Gradle will trigger those automated tests.  
For frontend, we are required to run karma with angular test. This will run all automated tests for the front end.

To run automatic tests, the order of initialization is fixed:

Start and load database -> Run backend -> Run frontend

The test cases mentioned in this document are for manual testing of running system. This test are to be run at end.

**Test Case 7.1**

**Title: Shorten URL**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-Conditions**  User has opened the Web UI. | | |  |  |
| **Steps** | **Action** | **System Response**  **Expected** | **Result** | **Tested by** |
| **1.** | **User** clicks “Shorten” button. | It will show error as input is less than 4 characters. | **Pass/Fail** | Subhajit Das |
| **2.** | **User** fills the URL input field and press “Shorten” button. | UI will show a success message with newly generated short URL and entered URL. | **Pass/Fail** |

**Test Case 7.2**

**Title: List URL**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-Conditions**   * At least 11 to maximum 20 URLs are shortened. * User in in shorten page. | | | |  |
| **Steps** | **Action** | **System Response Expected** | **Result** | **Tested by** |
| **1.** | **User** click on “List” link. | A list of all previously shortened URLs are shown. “Next” button should be active, while “Prev” button should be disabled. | **Pass/Fail** | Subhajit Das |
| **2.** | **User** presses the “Next” button. | List screen will be updated with newer entries. “Next” button will be disabled. “Prev” button will be active. | **Pass/Fail** |
| **3.** | **User** presses the “Prev” button. | List screen will be updated with previous entries. “Prev” button will be disabled. “Next” button will be active. | **Pass/Fail** |
| **4.** | **User** enters search term in search box. | The list will automatically update with code items, starting with that term. | **Pass/Fail** |  |
| **5.** | **User** clears search box by manually or “X” button. | List restores to previously all items. |  |  |

**Test Case 7.3**

**Title: Redirection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-Conditions**   * At least 1 URLs are shortened. * User must have the shortened URL. | | | | |
| **Steps** | **Action** | **System Response Expected** | **Result** | **Tested by** |
| **1.** | **User** enters a previously shortened URL into browser address bar directly. | User will be redirected to actual URL. | **Pass/Fail** | Subhajit Das |
| **2.** | **User** enters an URL with invalid code, such as (0) or code which was not generated previously. | User will see the error page. | **Pass/Fail** |