

Java programming Project 2

project :- Link shortener development

duration :- 7 days

date assigned :- 28/12/2023

## Project Overview

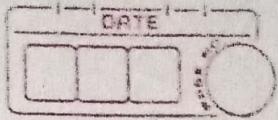
As part of your my ongoing Java programming internship, we are excited to introduce the week 2 project, "Link Shortener Development". This project will provide you with valuable hands-on experience in Java programming & web development.

The main goal of this project is to create a simple link shortener application using Java. A linkshortener is a tool that takes a long URL and generates a short, unique identifier, allowing users to share concise links. This project will give you insights into data structures, algorithms, and basic web application development.

## Project Objectives

To accurately outline the scope of work required for a project, it is crucial to first identify its objectives. Pinpointing what the project hopes to accomplish will assist in determining its inclusion & limitations.

- Implement a Link Shortener System in Java
- Develop a function to shorten long URLs and expand short URLs to their original form.
- Ensure the uniqueness of short URLs to handle potential collisions.
- Explore basic error handling and user feedback.
- Gain practical experience in Java programming, data structures, and algorithmic thinking.



## Requirements and features

- 1) Use Java as the primary programming language.
- 2) Create a class or set of classes that manage the short URLs.
- 3) Implement a basic hash function for generating short URLs.
- 4) Include error handling to address scenarios like duplicate long URLs and invalid short URLs.
- 5) Optionally, consider persisting data to maintain link mapping between sessions.
- 6) Develop simple command line interface or web-based interface for user interface.

## Working of Project

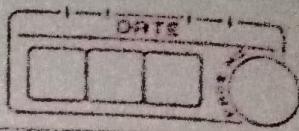
### 1) Initialization

When the program starts an instance of the 'Linkshortener' class is created. This instance initializes two hash maps ('shortToLongMap' and 'longToShortMap') to store mapping between short and long URLs.

```
private Map<String, String> shortToLongMap;  
private Map<String, String> longToShortMap;  
  
public Linkshortener() {  
    this.shortToLongMap = new HashMap<>();  
    this.longToShortMap = new HashMap<>();  
}
```

### 2) User Interaction

The 'main' method sets up a continuous loop for user interaction. It displays a menu allowing users to choose between shortening a URL, expanding a URL, or exiting the program.



(while c < true) {

// display menu options

System.out.println("choose an option:");

System.out.println("1. shorten URL");

System.out.println("2. expand URL");

System.out.println("3. Exit");

// switch condition.

int choice = Scanner.nextInt();

Scanner.nextLine();

switch (choice) {

case 1:

// Shorten URL

case 2:

// Expand URL

case 3:

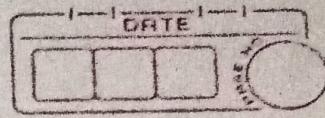
// Exit.

default :

// Invalid choice

}

}

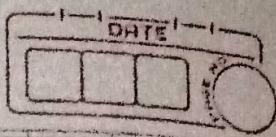


### 3) Shorten URL

When the user chooses to shorten a URL, the program prompts the user to enter a long URL. It then generates a short URL using the 'shortenUrl' method of the 'LinkShortener' class.

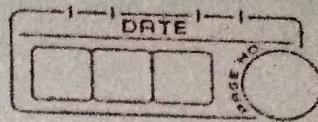
```
System.out.print("Enter the long URL: ");
String longUrl = scanner.nextLine();
String shortUrl = linkShortener.shortenUrl(longUrl);
System.out.println("shortened URL: " + shortUrl);
```

- Generates a short URL using the 'generateShortUrl' method.
- Checks if the generated short URL is already in use. If not, it adds mappings to both hashmaps ('shortToLongMap' and 'longToShortMap').
- Handles collisions by recursively generating a new short URL if needed.



## Error Handling :-

- The project includes basic error handling to address scenarios such as collisions (duplicate short URLs) and invalid short URLs.
- If a collision occurs during URL shortening, the program attempts to generate a new short URL.



Conclusion :-

The link shortener project has successfully realized its core objectives of URL shortening and expansion through a well-designed & modular Java application.

The project's key achievements include the implementation of a simple hash function for generating short URLs, user-friendly command-line interface, and basic error handling mechanisms.

The project serves as a solid foundation for URL management, with the potential for ongoing improvements to meet evolving requirements and user expectations.