Design Document for URL Shortener

.

Omri Anidgar

26.09.2024

Version: 1.0

Table of Contents

[Introduction 3](#_Toc178196560)

[Project Overview 3](#_Toc178196561)

[Architecture Overview 3](#_Toc178196562)

[Technology Stack 4](#_Toc178196563)

[Component Design 4](#_Toc178196564)

[Data Flow 5](#_Toc178196565)

[User Interface Design 6](#_Toc178196566)

[Security Considerations 6](#_Toc178196567)

[Scalability and Performance 6](#_Toc178196568)

[Testing Strategy 7](#_Toc178196569)

[Deployment Strategy 7](#_Toc178196570)

[Conclusion 7](#_Toc178196571)

# Introduction

**Purpose**

This document outlines the design choices and architecture of the URL Shortener, serving as a guide for developers involved in the project.

**Scope**

The scope includes the design and implementation of a web-based URL shortening service, which encodes long URLs to shorter, shareable links.

# Project Overview

**Project Goals**

* Provide users with a simple interface to shorten URLs.
* Track the number of visits for each shortened URL.
* Ensure reliable performance and security for users.

**Target Audience**

Individuals and businesses looking for a convenient way to share and track URLs.

# Architecture Overview

**Architecture Style**

The project employs a RESTful API architecture, facilitating communication between the frontend and backend components.

**High-Level Architecture Diagram**



# Technology Stack

**Frontend**

* **Framework**: React for building a responsive user interface.

**Backend**

* **Runtime**: Node.js with Express for handling API requests.
* **Database**: Cloud MongoDB for storing URL data.

**Infrastructure**

* **Web Server**: Nginx for serving the frontend and acting as a reverse proxy for the backend.
* **CDN**: Cloudflare Workers for caching and request handling.
* **Hosting**:
  + **Backend**: Railway for deploying the Node.js application.
  + **Frontend**: Netlify for hosting the React application.
* **Domain**: porkbun for the domain ([ktzr.lol](https://ktzr.lol))

# Component Design

**Frontend Components**

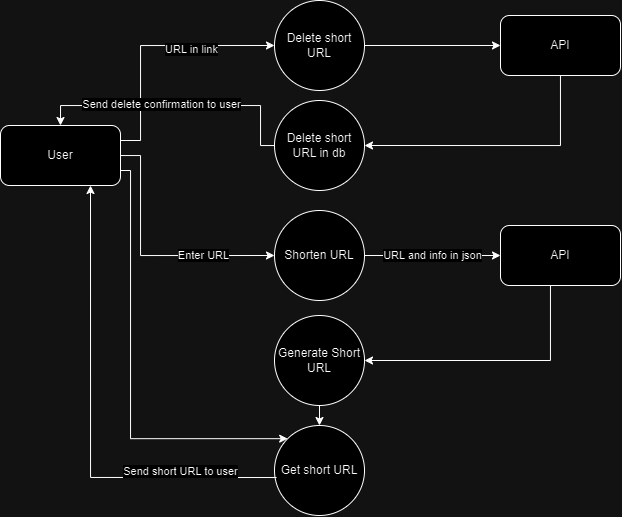
* **URL Input Form**: Allows users to submit long URLs for shortening.
* **Results Display**: Shows the shortened URL and click statistics.

**Backend Services**

* **URL Encoding Service**: Encodes URLs into base62 format.
* **Click Tracking Service**: Monitors and records the number of clicks on each shortened URL.

# Data Flow

**Data Flow Diagram**



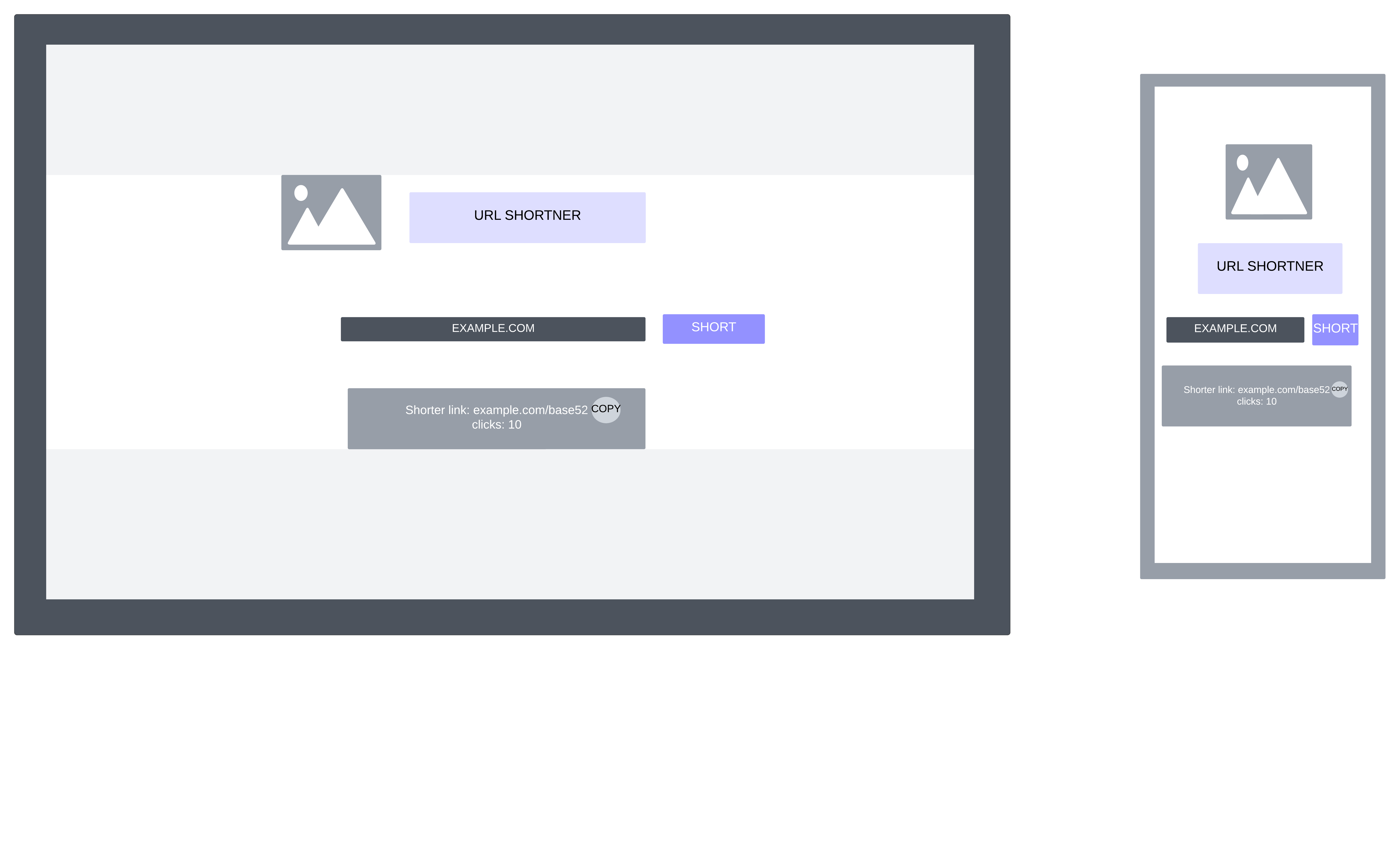
**Database Design**

* **Urls Table**:

|  |  |
| --- | --- |
| \_id | ObjectId (auto generated) |
| originalUrl | string |
| clicks | number |
| date | string |
| shortUrl | string |

# User Interface Design

**Wireframe Mockups**



**User Experience Considerations**

The interface is designed for simplicity and ease of use, with clear prompts for input and intuitive navigation.

# Security Considerations

* **CORS Middleware**: Implemented to handle cross-origin requests securely.
* **Data Protection**: Measures will be taken to ensure user data is secure and not exposed via the API.

# Scalability and Performance

**Scalability Strategies**

* The application can scale horizontally by deploying multiple backend instances and using a load balancer.

**Performance Optimization**

* Caching strategies via Cloudflare Workers will be used to reduce server load and improve response times.

# Testing Strategy

**Testing Types**

* Unit tests for backend services.
* Integration tests for API endpoints.

**Tools and Frameworks**

* Jest for backend testing.
* React Testing Library for frontend testing.

# Deployment Strategy

**Deployment Pipeline**

A CI/CD process will be established using GitHub Actions to automate testing and deployment to Railway and Netlify.

**Hosting Details**

* The backend will be hosted on Railway, enabling easy deployment and scalability.
* The frontend will be hosted on Netlify for efficient delivery.

# Conclusion

**Summary**

The URL Shortener will deliver a straightforward and effective service for encoding URLs, tracking usage, and providing an enjoyable user experience.

**Future Considerations**

Potential enhancements may include user accounts for personalized tracking, analytics dashboards for URL performance.