University Of Mumbai

URL Shortener

By

Amit Kesarwani & Suraj Raut

Under Guidance

of

Counsellor’s name

Submitted to the college of Vidyavardhini's College of Engineering and Technology

Vasai Road – 401202

in partial fulfilment of the requirements

for the degree of

Master’s

Of

Computer Applications

Mumbai, Maharashtra - 400032

INDEX

Introduction

"Welcome to our Short URL Generator! Our platform was designed to make it easy to share long, unwieldy URLs in a compact and easy-to-manage format. With just a few clicks, you can shorten any URL to a more manageable length, making it easier to share on social media, in emails, or anywhere else.

Our **URL Shortener** is perfect for anyone who needs to share links frequently, whether you're a blogger, social media manager, or just someone who wants to share interesting content with your friends and followers. Plus, our tool is completely free to use and doesn't require any complicated sign-up or verification process.

Using our **URL Shortener** is simple. Just paste in the long URL you want to shorten, click the '**Save**' button, and voila! You'll receive a compact, shortened URL that you can easily copy and share with others. Our tool also allows you to track the performance of your shortened URLs, so you can see how many clicks they receive and where those clicks are coming from.

We're excited to provide this powerful and convenient tool to help you make the most of your online content. Whether you're a marketer, blogger, or just someone who wants to make sharing links easier, our Short URL Generator is the perfect solution. Try it out today and see how it can streamline your online content sharing!"

Project Objectives

Simplify long URLs: The primary objective of a URL shortener is to simplify long, complicated URLs into shorter, more manageable links.

1. Save space: Shortened URLs take up less space in email messages, social media posts, and other online content, allowing users to share more information in a smaller amount of space.
2. Improve readability: Long URLs can be difficult to read and remember, especially when they contain a lot of random characters. Shortened URLs are easier to read and remember, making them more user-friendly.
3. Track link performance: URL shorteners can provide data on how many clicks each link receives, which can help website owners and marketers track the performance of their online content.
4. Branding and customization: Many URL shorteners allow users to customize the shortened links with their own brand or keywords, helping to create a consistent brand identity and improve recognition.
5. Security: URL shorteners can also provide additional security features, such as password protection, to prevent unauthorized access to sensitive information.

These are just a few possible objectives of a URL shortener project.

Project Scope

1. User Interface: A URL shortener should have a simple, user-friendly interface that allows users to quickly and easily shorten URLs. It should also provide clear feedback on the status of each shortened URL.
2. Shortening Algorithm: The URL shortener should use a reliable and secure algorithm to shorten URLs. This algorithm should be able to create unique, non-guessable short links that are resistant to hacking attempts.
3. Customization: The URL shortener should allow users to customize their shortened links, such as by adding a custom domain or a personalized keyword. This can help users establish a consistent brand identity and improve recognition.
4. Link Management: The URL shortener should provide a way for users to manage their shortened links, such as by allowing them to view click statistics, edit link details, and delete links as needed.
5. Security: The URL shortener should implement security features to prevent unauthorized access to sensitive information. This may include password protection for certain links, as well as measures to protect against spam and hacking attempts.
6. Integration: The URL shortener should be easy to integrate with other online platforms and tools, such as social media networks, email marketing tools, and analytics platforms.
7. Scalability: The URL shortener should be scalable, meaning it should be able to handle a large volume of requests without slowing down or crashing. This may require the use of cloud-based infrastructure or other advanced technologies

Business Case for Project

Long URLs are very tough to share & type. So Url-Shortner is one of the good idea where we can create a service that maps our short custom URLs to long URLs. Also, we can get some analytics on URLs like,

- How many times which URL is visited.
- Analytics on device usage like Android or Desktop with OS
- This service has scope in B2B, B2C as well as C2C.
- Short links can be used in campaigns as well to see diff sources & pre-link processing.

`https://github.com/AmitSureshChandra/Url-Shortner/blob/main/backendservice/src/main/kotlin/com/github/amitsureshchandra/urlshortner/controller/UrlController.kt`

The above given URL is much long so we can short it like `bit.ly/url-controller` which is much simpler to share & use.

The above service

- mainly redirects short URL to long URL.
- gives analytics over link like (visits, time based usage, platform from where link accessed)
- secures our original links by creating expiring short URLs.

Software Requirement Specification

❖ Functional Requirements

- ➢ User should be able to create short links.
  - ✓ With custom suffix or with auto generated
- ➢ User should be able to edit created links.
  - ✓ Depends on whether user has subscribed to premium plan.
- ➢ Short link analytics Dashboard.
  - ✓ For premium plans (limited details for free users)
- ➢ Able to add expiry time after that short link should expire.
- ➢ User can be able to enable/ disable link.
- ➢ User can choose custom subdomain for short URLs.
  - ✓ Included in premium plan.
- ➢ Authentication via Username & Password / Social Login

❖ Non-Functional Requirements

- ➢ SLA Uptime
- ➢ Reliability
- ➢ Authentication
- ➢ Security

❖ Interface Requirement

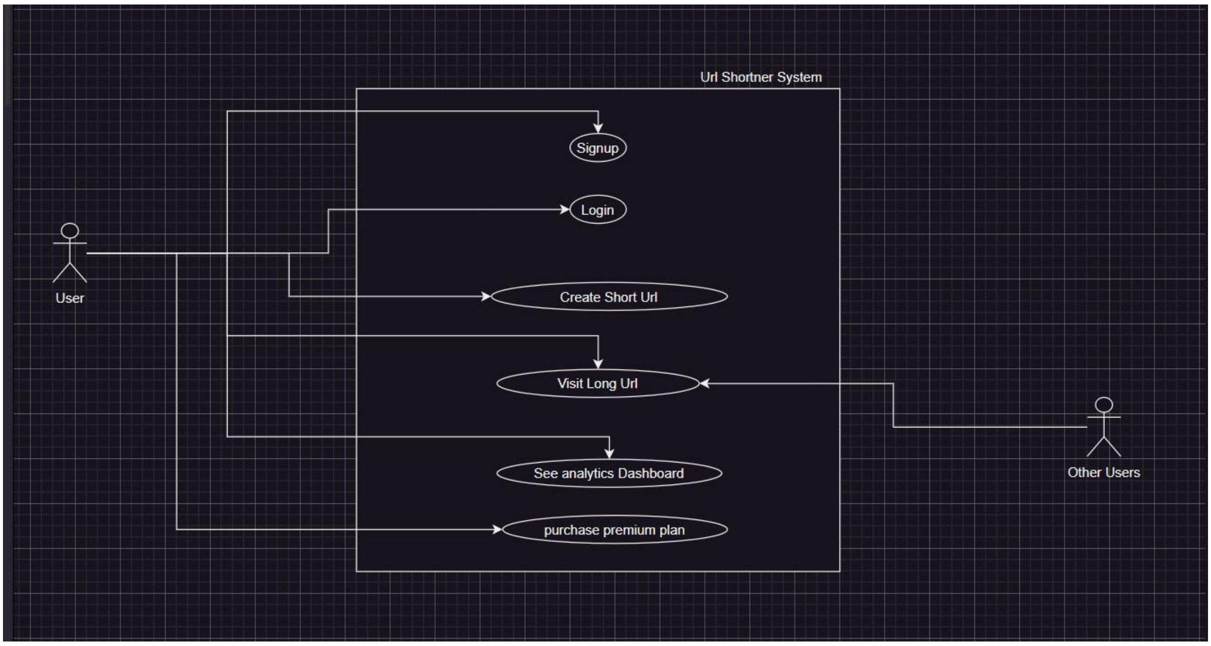
- ➢ Frontend Service
  - ▪ For forms to create/ edit short URLs
- ➢ Backend Service
  - ▪ For redirect & admin work

❖ Performance Requirement

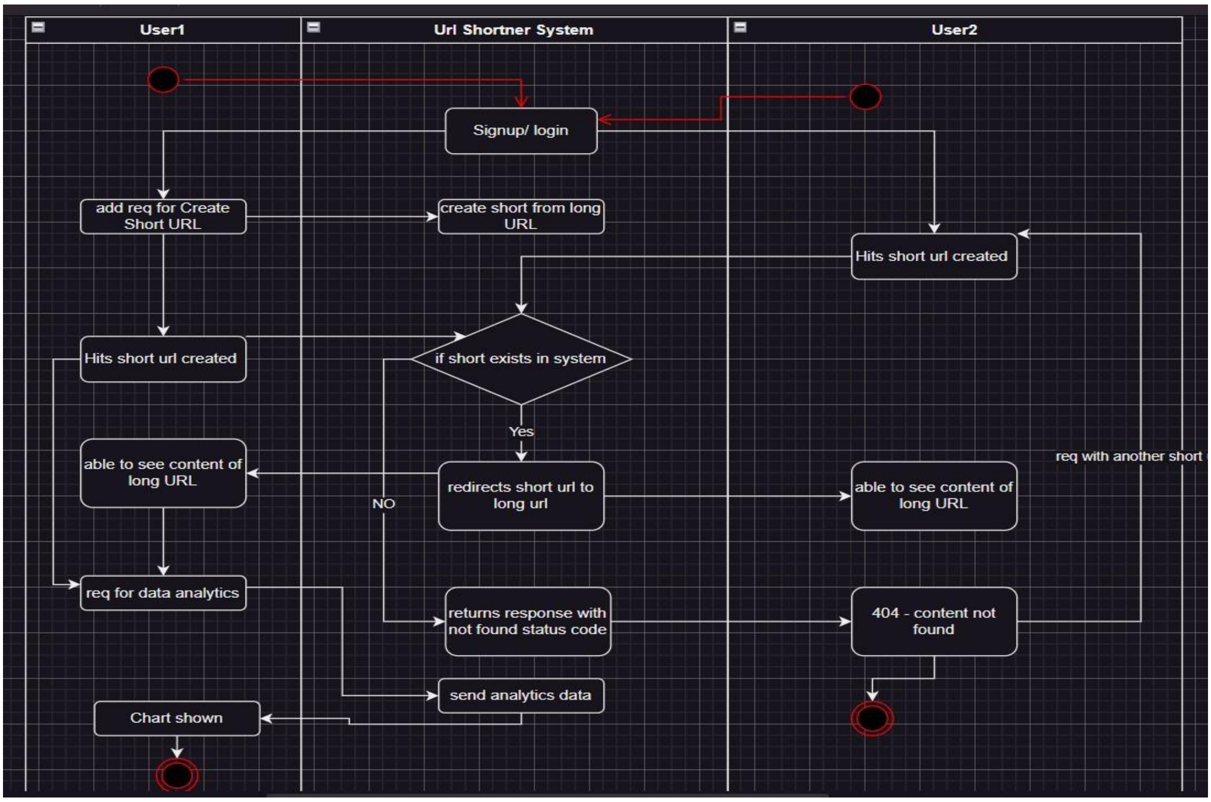
- ➢ Service should be able to handle at least 1000 users/second.

UML Diagrams

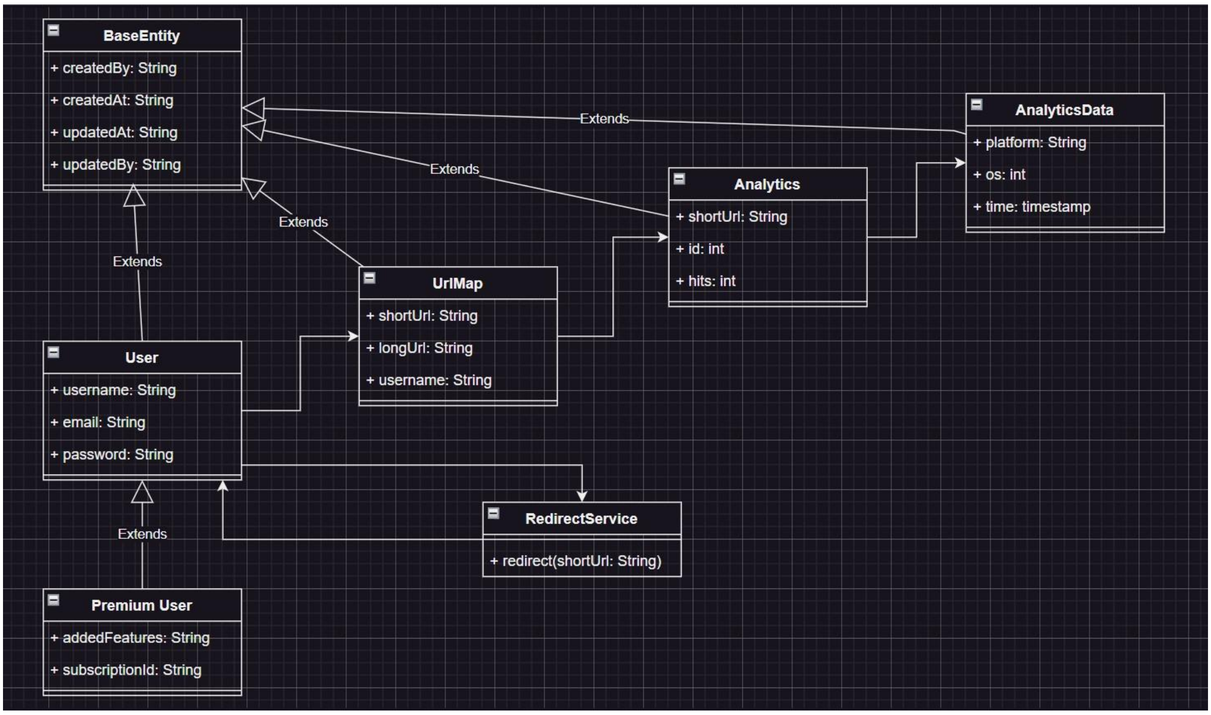
Use Case Diagram



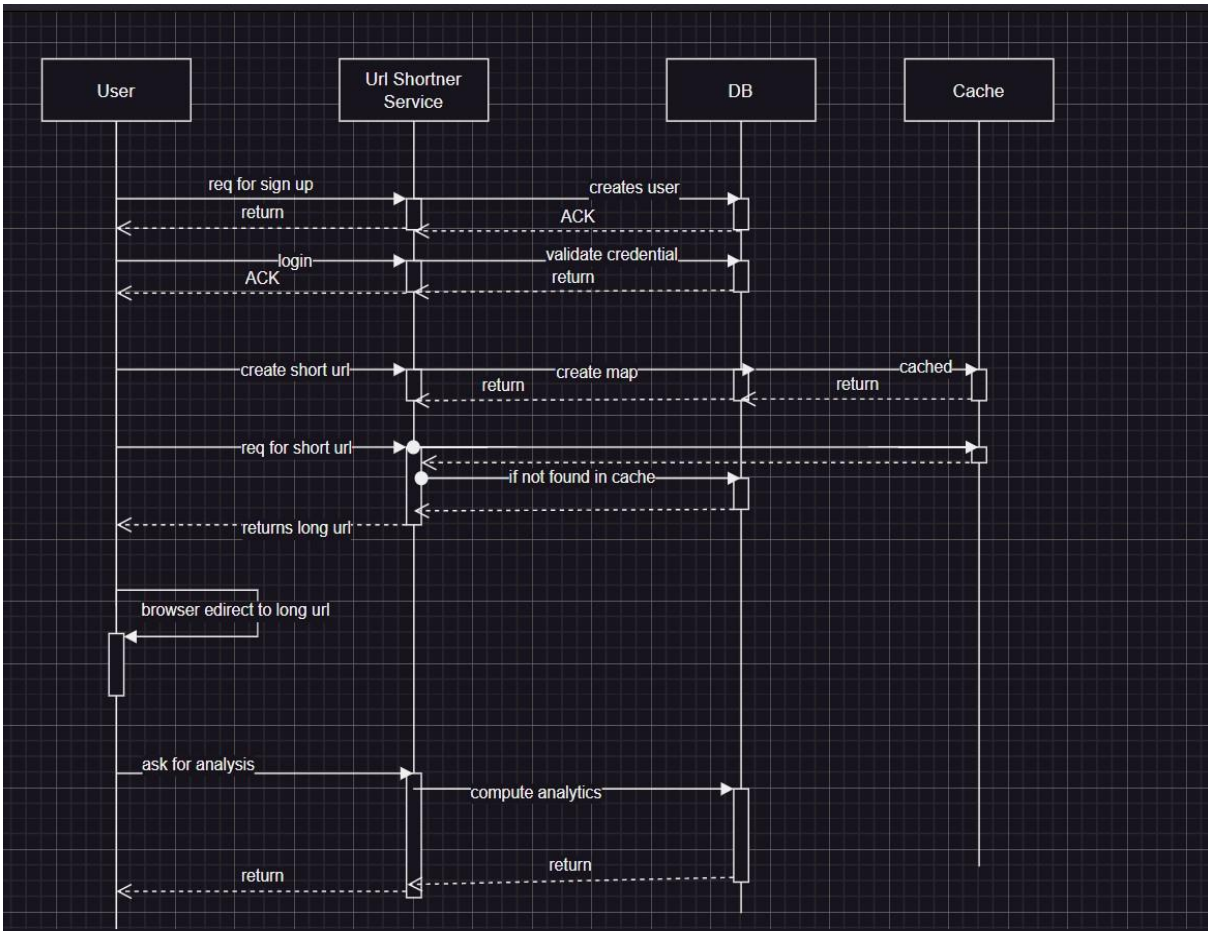
Activity Diagram


User1
Url Shortner System
User2
Signup/ login
add req for Create Short URL
create short from long URL
Hits short url created
Hits short url created
if short exists in system
Yes
req with another short
able to see content of long URL
redirects short url to long url
able to see content of long URL
NO
req for data analytics
returns response with not found status code
404 - content not found
send analytics data
Chart shown

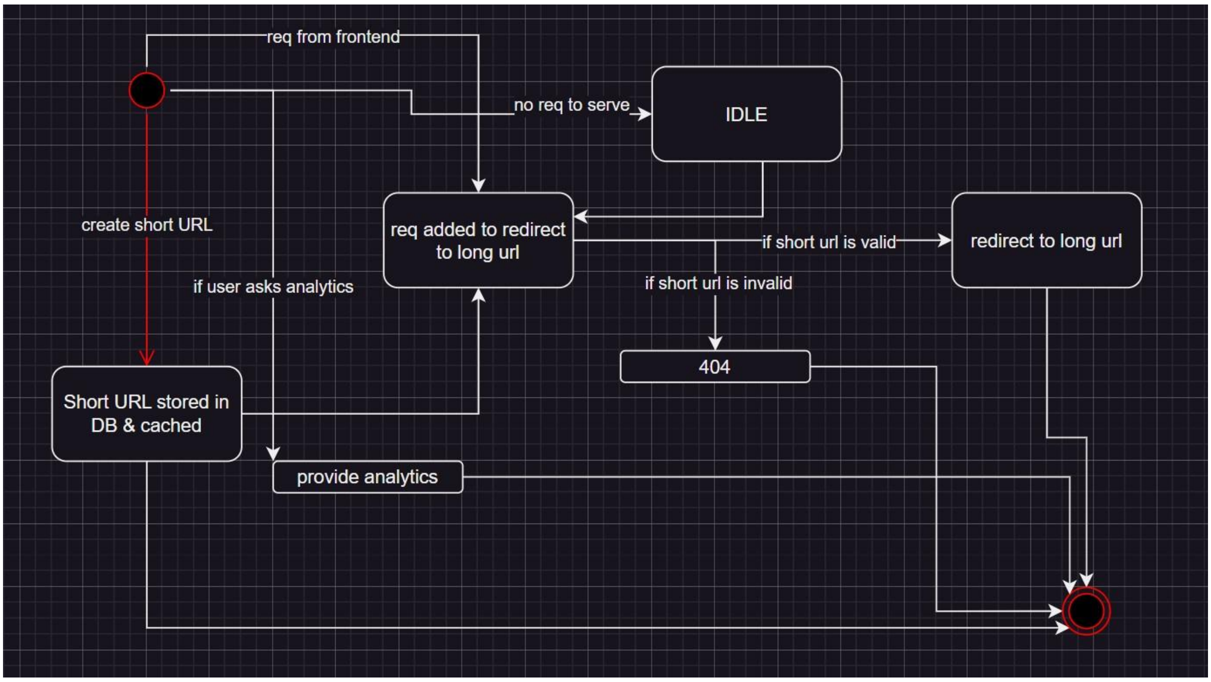
Class Diagram



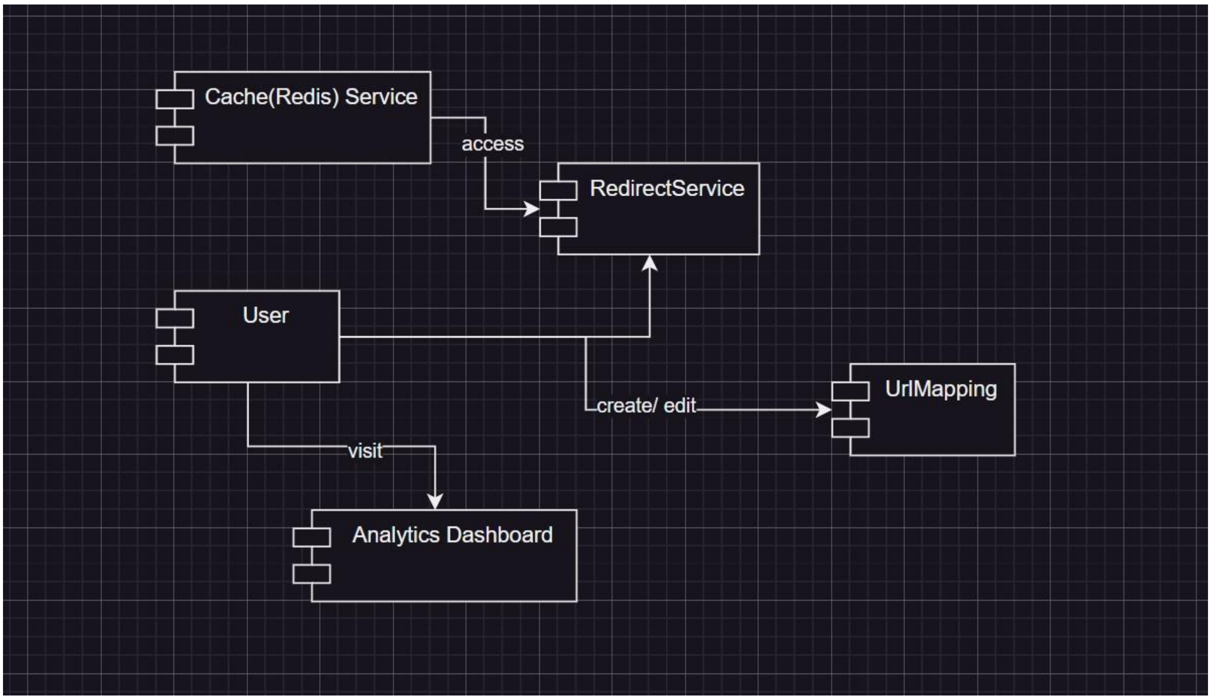
Sequence Diagram



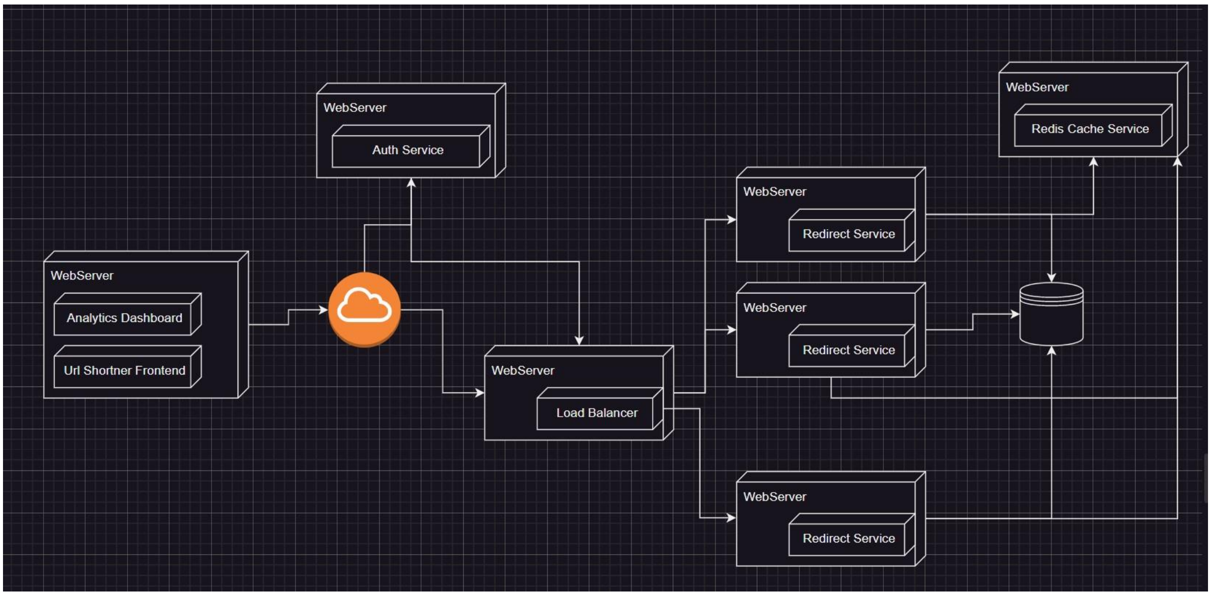
State Chart Diagram



Component Diagram

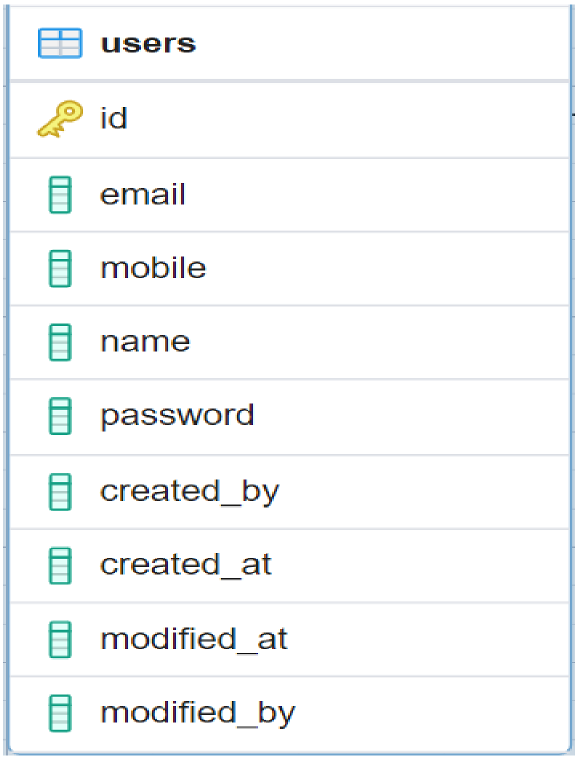


Deployment Diagram



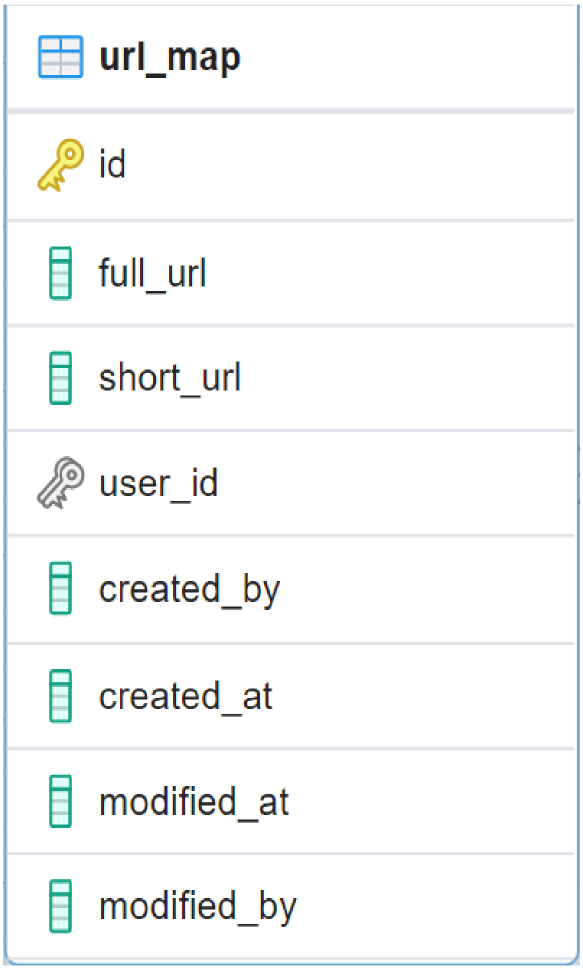
Database Schema

- ❖ User table



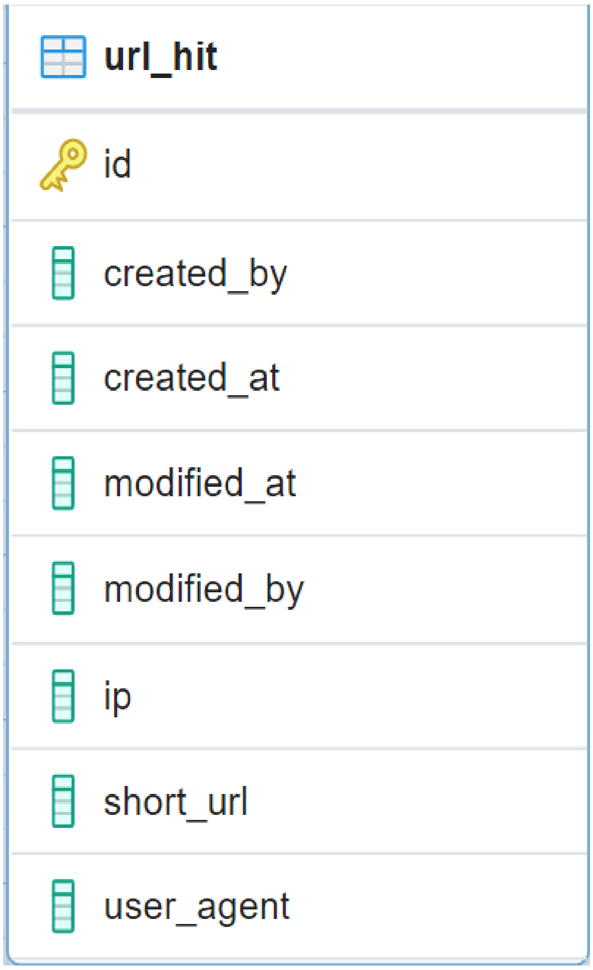
users
id
email
mobile
name
password
created_by
created_at
modified_at
modified_by

- ❖ Url Map table



url_map
id
full_url
short_url
user_id
created_by
created_at
modified_at
modified_by

- ❖ Url hit table



url_hit

id

created_by

created_at

modified_at

modified_by

ip

short_url

user_agent

Books & References

Stack Overflow	**https://stackoverflow.com/**
Spring	**https://spring.io/**
Vue Js	**https://vuejs.org/**
Postman	**https://www.postman.com/**
Kotlin	**https://kotlinlang.org/**