## Section 1 - Project Description

### 1.1 Project
Url Shortener Service

### 1.2 Description
The service will take long URLs as input and provide shortened, unique aliases that redirect to the original URLs.

### 1.3 Revision History

| Date | Comment | Author |
|---|---|---|
| 08/04/2024 | First Release | Gihan |
| | | |
| | | |
| | | |
| | | |

## Contents

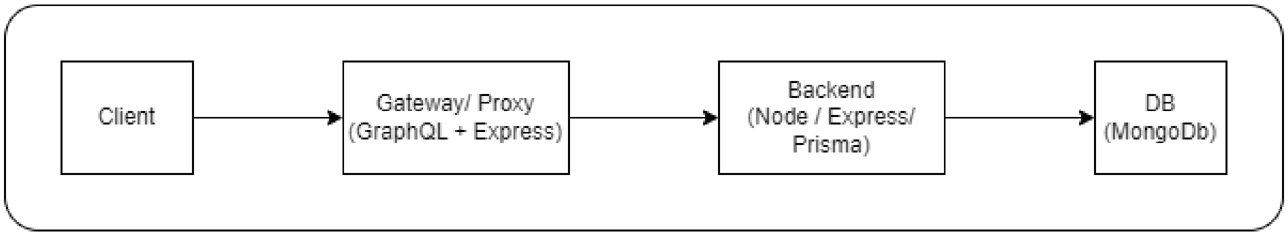
## Section 2 - Overview

### 2.1 Purpose

The service will take long URLs as input and provide shortened, unique aliases that redirect to the original URLs.

### 2.2 Scope

System designed with two modules
1. Backend System
2. Gateway Proxy System


Client
Gateway/ Proxy (GraphQL + Express)
Backend (Node / Express/ Prisma)
DB (MongoDb)

Backend system designed on top of Node.js runtime environment and used Express as the Framework

Gateway system designed on top of Node.js runtime environment and used GraphQl + Express as the Framework

*****Note: Required installation and setup details can be acquired from the README.md file in each project repo***

-Tech Stacks
-Node.Js
-TypeScript
-Express.js
-MongoDb
-Prisma ORM
-Graphql

-Features

```
- ***Package managament*** -- npm
- ***Testing*** -- Jest and Supertest
- ***Cross-Origin Resource-Sharing*** -- using cors
- ***Secured HTTP Headers*** -- helmet
- ***Logging*** -- winston
- ***Environment variables*** -- dotenv
- ***Compression*** -- gzip
- ***Git hooks*** -- husky
- ***Code quality*** -- Eslint
- ***Code style and formatting*** -- Prettier
- ***Containerization*** -- Docker
```

**- Project Structure in Backend Service**

~~~
Project Structure in Backend Service
```bash
__test__/
├ app.test.ts
├ url.routes.test.ts
└ url.service.test.ts
dist/
logs/
prisma/
src/
├ config/
│ └ config.ts
├ controllers/
│ └ shortener.controller.ts
├ middleware/
│ ├ logger.ts
│ └ validate.ts
├ routes/
│ ├ index.ts
│ └ url.router.ts
├ services/
│ └ url.service.ts
├ utils/
│ ├ compressFilter.util.ts
│ └ validationSchema.util.ts
├ app.ts
├ env.d.ts
└ index.ts
```
~~~

**- Project Structure in Gateway Service**

~~~
```bash
__test__/
┗ server.test.ts
dist/
logs/
src/
┣ config/
┃ ┗ config.ts
┣ Middleware/
┃ ┣ connection.handler.ts
┃ ┗ logger.ts
┣ Schema/
┃ ┣ Mutations/
┃ ┣ Queries/
┃ ┃ ┗ UrlQueries.ts
┃ ┣ TypeDefs/
┃ ┃ ┗ Urls.ts
┃ ┗ index.ts
┣ env.d.ts
┗ server.ts
```
~~~

## 2.3 Requirements

Design and implement a URL shortener service using Node.js and Typescript. You are free to choose any database for storing data, with a preference for scalable solutions. Additionally, consider incorporating gRPC or GraphQL for communication, Prisma for database interactions, and Nest.js for the application framework.

### 2.3.1 Estimates

| # | Description | Hrs. Est. |
|---|---|---|
| 1 | Brief description of task / module with link | # est |
| | **TOTAL:** | # est tot |

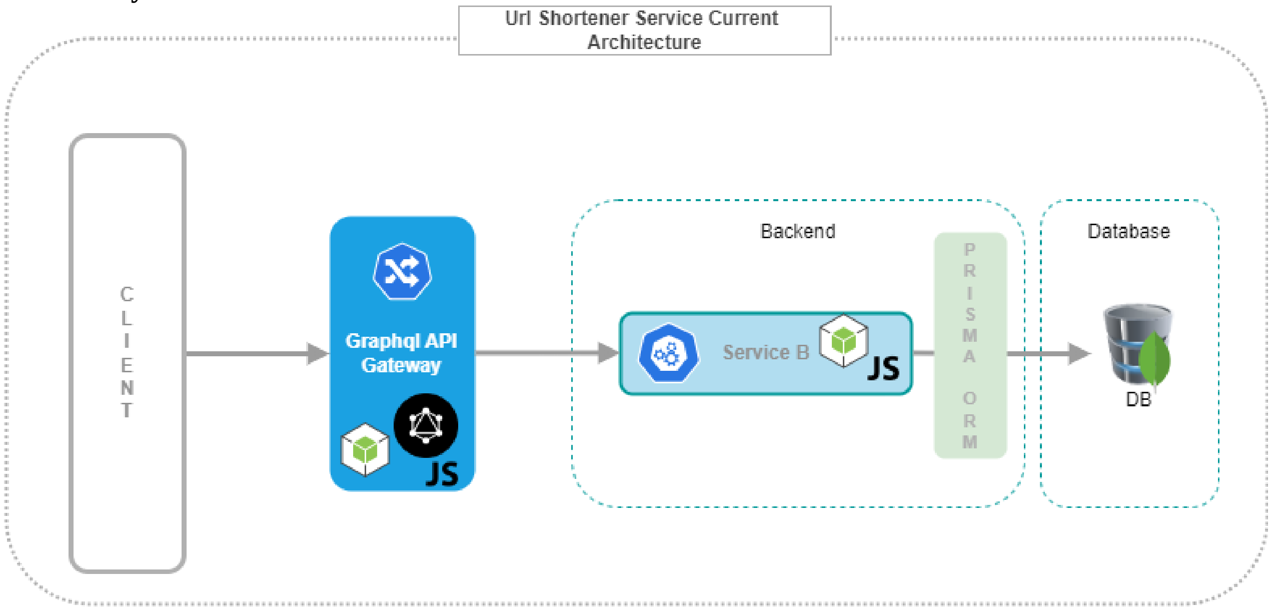
### 2.3.2 Traceability Matrix

Cross reference this document with your requirements document and link where you satisfy each requirement

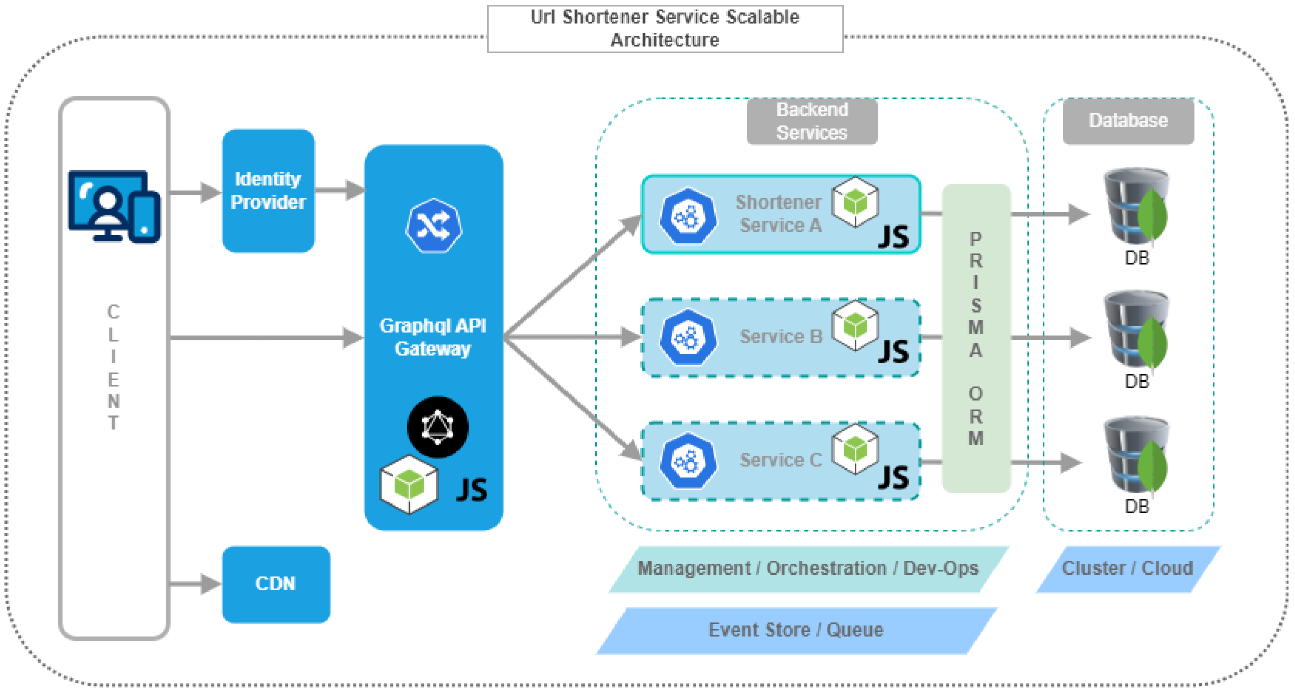
| SRS Requirement | SDD Module |
|---|---|
| Req 1 | |
| | |
| | |

## Section 3 - System Architecture

### Current System Architecture



### Scalable System Architecture



## Section 4 - Data Dictionary

Used MongoDb as the database

(template of a database collection description)

**Table**
```
{
  "_id": {
    "$oid": "660fe38d26da569cc7331bd3"
  },
  "longUrl": "aaaaaaaa",
  "shortenedUrl": "bbbb"
}
```

| Field | Notes | Type |
|---|---|---|
| Id | Unique Identifier from urls | ObjectId |
| longUrl | String | String |
| shortenedUrl | The Value output from somewhere | String |

## Section 5 - Software Domain Design

### 5.1 Software Application Domain Chart

Describe / chart each major software application domain and the relationships between objects (UML, etc)

### 5.2 Software Application Domain

A Comprehensive high level description of each domain (package/object wherever it is better to start) within the scope of this module (or within the greater scope of the project if applicable)

#### 5.2.1 Domain X

A high level description of the family of components within this domain and their relationship. Include database domain, stored procedures, triggers, packages, objects, functions, etc.

##### *5.2.1.1 Component Y of Domain X*

Define Component Y, describe data flow/control at component level

5.2.1.1.1 Task Z of Component Y1 of Domain X
Define Task Z, describe data flow/control at task level

## Section 6 – Data Design

Describe the data contained in databases and other shared structures between domains or within the scope of the overall project architecture

### 6.1 Persistent/Static Data

Used MongoDb as the database

**Create Database Static Resords**

Once DB connection is created run below command to add test records in DB – in Backend service

```
npx prisma db seed
```

#### 6.1.1 Dataset
Describe persisted object/dataset and its relationships to other entities/datasets

#### 6.1.2 Static Data
Describe static data

#### 6.1.3 Persisted data
Describe persisted data

### 6.2 Transient/Dynamic Data
Describe any transient data, include any necessary subsections

### 6.3 External Interface Data
Any external interfaces' data goes here (this is for the data, section 8 is for the interface itself)

### 6.4 Transformation of Data
Describe any data transformation that goes on between design elements

## Section 7 - User Interface Design

### 7.1 User Interface Design Overview
Pictures, high level requirements, mockups, etc.

### 7.2 User Interface Navigation Flow
Diagram the flow from one screen to the next

### 7.3 Use Cases / User Function Description
Describe screen usage / function using use cases, or on a per function basis

## Section 8 - Other Interfaces
Identify any external interfaces used in the execution of this module, include technology and other pertinent data

### 8.1 Interface X
Describe interactions, protocols, message formats, failure conditions, handshaking, etc

## Section 9 - Extra Design Features / Outstanding Issues
Does not fit anywhere else above, but should be mentioned -- goes here

## Section 10 – References
Any documents which would be useful to understand this design document or which were used in drawing up this design.

## Section 11 – Glossary
Glossary of terms / acronyms