Design Documentation

1 Functionalities

1. Create a unique short URL with a path length of 7 characters for any given URL.
2. Redirect to the original URL using a given registered short URL.
3. Retrieve the registered URL information by sending either a generated short URL or an original URL.
4. User registration and authentication for session management.
5. User account is needed to create a new short URL and each user can generate a limited number of short URLs. The limit can be increased by request.
6. Able to generate private short URLs that only the registered user can use using an unexpired JWT token.
7. Generated short URLs have an expiration time that can be extended upon request.
8. Users can fetch their URLs and expiration times.

2 Non-functionalities

1. Scalability

   All microservices are horizontally scalable except for the token service, which does not need to be scaled. The short URL paths are generated using 64 characters and a length of 7, allowing about 4.4 trillion unique short URLs.

2. Low latency

   The major request that requires low latency will be the redirection service. Using Cassandra to fetch URLs allows low latency for this functionality. JWT token authenticates a user without using the User service, thereby lessening the latency in redirecting with a private URL.

3. Low cost

   Data for expired URLs or tokens are regularly cleaned up in all databases by setting recurring events or expiration times.

4. Security

   API Gateway is used to prevent external access to internal communication. A JSON Web Token is used for session management which has a default expiration time of 15 minutes. OAuth 2.0 is applied to enable direct access to the Short URL Service functions that require user information.

5. Strong consistency

   User data is modified using the row lock functionality of MySQL with a timeout. Once a URL is registered, it is not modified unless expired.

6. Fault tolerance

   By scaling the microservices in a Cloud using Kubernetes, the program will be fault-tolerant, highly available, and have low latency.

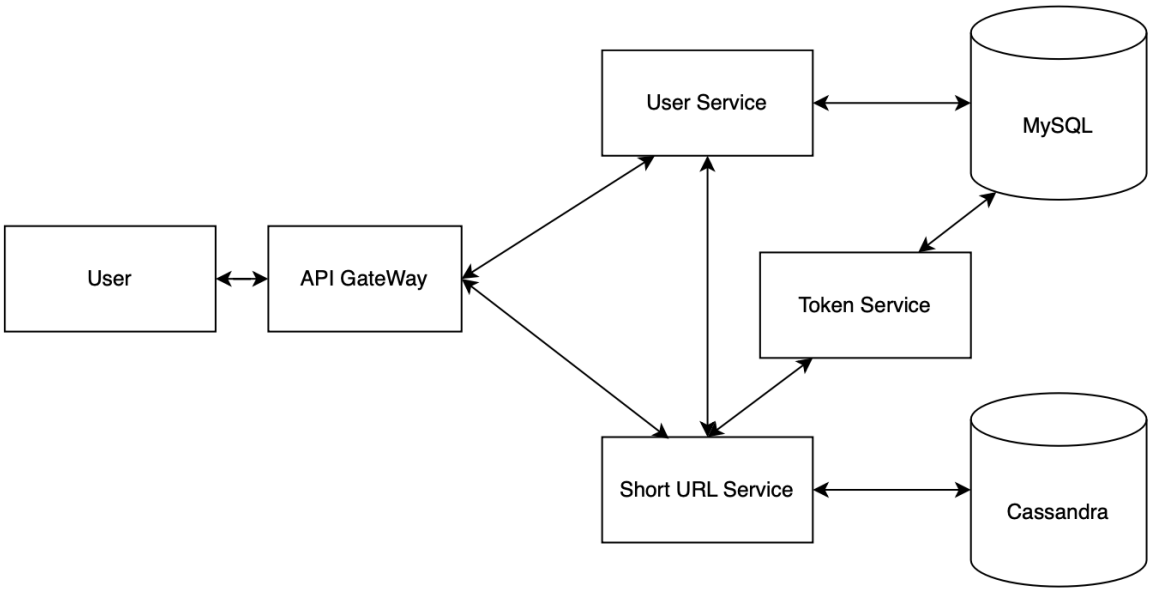
7. High availability

   Fault tolerance and scaling will allow high availability.

8. Durability

   Cassandra has data durability because it replicates data in multiple nodes. MySQL has durability by maintaining a transaction log file and can further enhance durability by setting master-slave replication.

3 System Design



3.1 Token Service

It gives a token that allows about 4.4 thousand pre-defined short paths to Short URL servers. A seed is generated by adding a fixed big number to the previous seed and dividing it by 1 billion. A generated seed is sent to a User service by request. A URL service generates a number by adding 1 billion to the previous number. This number is converted

into base 64 where each place value is mapped to an alphanumeric character, a dash, or an underscore.

Token service should run in only one server.

3.2 User service and Short URL service

These services can horizontally scale up to handle more requests.

3.3 MySQL Database

3.3.1 Table ‘app_user‘

column name	data type	description
id	integer not null	Primary key
email	varchar(100) not null	A unique email address
password	varchar(255) not null	Password
firstname	varchar(50)	First name
lastname	varchar(50)	Last name
app_user_role	varchar(255)	Role: ADMIN, USER
available_short_url	integer	Remaining number of URLs the user can generate

3.3.2 Table ‘short_url_path‘

column name	data type	description
short_url_path	varchar(7) not null	Short URL path (Primary Key)
created_at	datetime(6) not null	Created date and time
expire_date	date not null	Expire date
is_active	bit	True if active
is_private	bit	True if private (Not updatable)
user_id	integer	Foreign key for the id of ‘app_user‘

3.3.3 Table ‘token‘

column name	data type	description
seed	integer not null	Seed of a token
created_at	datetime(6) not null	Created date and time
expire_date	date not null	Expire date

3.4 Cassandra Database

column name	data type	description
key	integer not null	The queried URL
query_name	datetime(6) not null	A secondary optional primary key
value	date not null	The retrieved URL
text	date not null	Username or URL description
TTL	(internal)	Time to live

4 User API

4.1 User services

- **Register a new user**

```
POST /api/v1/auth/register
```

Payloads: firstname, lastname, email, password
Return: a JWT token

- **Login (duration: 15 minutes)**

```
POST /api/v1/auth/authenticate
```

Payloads: firstname, lastname, email, password
Return: a JWT token

- **Generate a new short URL**

```
POST /api/v1/user/generate
```

Payloads: longUrl, isPrivate, description
Return: a JWT token

- **Increase the number of short URLs for current user**

```
POST /api/v1/user/refill
```

Payloads: number
Return: availableBefore, availableAfter

- **Extend URL Expiration time**

```
POST /api/v1/user/extend
```

Payloads: number, ShortUrl, longUrl
Returns: isExtended, prevExpireDate, newExpireDate, remainingNumber

Disactivate/reactivate URL

```
POST /api/v1/user/disactivate
POST /api/v1/user/activate
```

Payloads: shortUrl, isPrivate
No returns

- **Search for URLs registered by a user**

```
GET /api/v1/user/urls
```

Payloads: isActive
Return: list of (shortUrl, longUrl, description, isPrivate, isActive)

4.2 Short URL Services

URLs can be registered both 1) publicly or 2) privately by users. If URL is registered privately, it can be only accessed by the corresponding user.

- **Get a short URL of a given long URL**

```
POST /api/v1/shorturl/short
```

PayLoads: longUrl
Return: shortUrl

- **Get a long URL of a given short URL**

```
POST /api/v1/shorturl/long
```

PayLoads: shortUrl
Return: longUrl

- **Redirect to the long URL website**

```
GET /short path
```

Payloads: longUrl
Return: The redirected page will be opened.

(Redirect may fail if the long URL doesn't have prefixes such as "http://")

5 Local Run

- Run docker instances

  $ docker compose up –d

MySQL Database (port 3306), Cassandra Database (port 9042), Zipkin (port 9411)

- Run all services

```
$ mvn clean install
$ java -jar eureka-server/target/eureka-server-1.0-SNAPSHOT.jar
$ java -jar apigw/target/apigw-1.0-SNAPSHOT.jar
$ java -jar token/target/token-1.0-SNAPSHOT.jar
$ java -jar appuser/target/appuser-1.0-SNAPSHOT.jar
$ java -jar shorturl/target/shorturl-1.0-SNAPSHOT.jar
```

Eureka server (port 8761), API Gateway (port 80), User service (port 8081), Short URL service (port 8082), Token (port 8083)

The default hostname for URL generation is http://127.0.0.1, which is the local host. The hostname can be configured by modifying ‘hostname‘ in both appuser/src/resources/application.yml and shorturl/src/resources/application.yml.