Table of Contents

[Introduction 2](#_Toc39021209)

[Purpose 2](#_Toc39021210)

[Overview 2](#_Toc39021211)

[System Architecture 2](#_Toc39021212)

[Architecture Design 2](#_Toc39021213)

[Database Design 2](#_Toc39021214)

[URL Mapping 3](#_Toc39021215)

[Functional Overview 3](#_Toc39021216)

[Running Instructions 3](#_Toc39021217)

[Test Results 3](#_Toc39021218)

# Introduction

## Purpose

The purpose of this document is to provide the complete overview of application.

## Overview

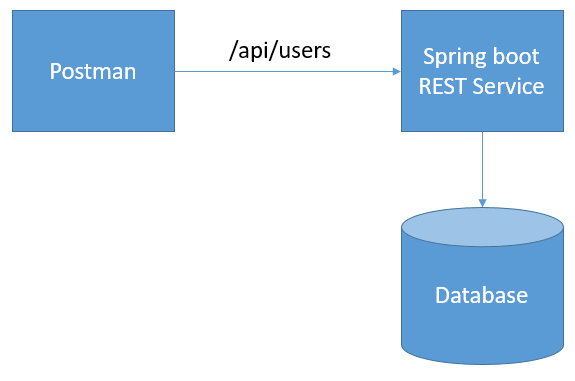
This RESTful Web Service application that will be used to manage two data types:

1. User: A user of the system
2. Phone: A phone belonging to a user. Each user can have 0 to many phones.

# System Architecture

## Architecture Design

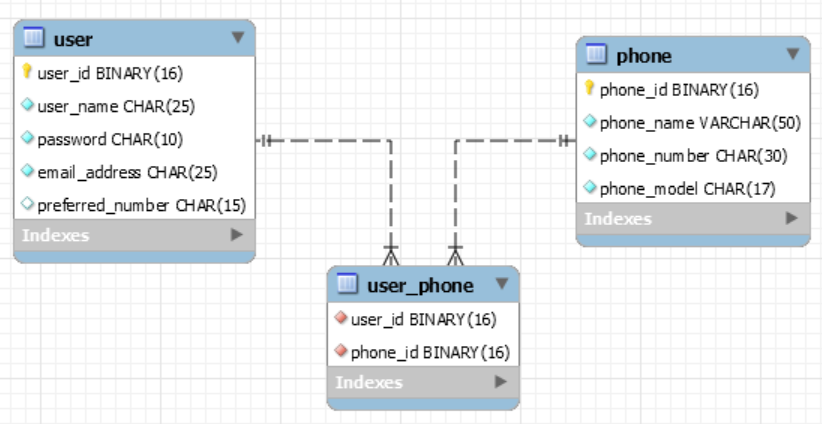
This is a basic architecture



## Database Design

There are three different tables –

1. User table – This table holds the user information
2. Phone table – This table holds the user’s phone details
3. User\_phone table – This table holds the mapping for user and phone.



## URL Mapping

|  |  |  |
| --- | --- | --- |
| **Methods** | **URL** | **Purpose** |
| GET | /api/users | Provide list of users in the system. |
| GET | /api/users/{userId}/phone | Provide list of phone details for specific user. |
| POST | /api/users | New user will be added in the system. |
| POST | /api/users/{userId}/phone | New phone details will be mapped to that specific user. |
| PUT | /api/users | Update user details in the system. |
| DELETE | /api/users/{userId} | Delete the specific user. |
| DELETE | /api/users/{userId}/phone/{phoneId} | Delete the specific phone detail for that user |

# Functional Overview

* 1. In this system user can hold their details like – user name, password, email address and preferred phone number.
  2. If the user wants to add additional phone details, they can add them as well.
  3. Additional phone details will consist of phone name, phone model, phone number.
  4. Phone model can of 4 types - IPHONE, ANDROID, DESK\_PHONE, SOFT\_PHONE
  5. Each user can add 0 – many phone numbers.
  6. User can delete their phone details and can update user details as well.
  7. In case the user is removed from the system all the user details and phone details will be deleted from the system.

# Running Instructions

In order to run the application on local system below are the steps –

1. Update the database details in *application.properties*.
2. Update the file path for log in *logback.xml.*
3. Run the maven clean command to clean the files and directories – *mvn clean*.
4. Run the maven install package command to generate the jar file – *mvn install package*.
5. Once the jar is generated run the given command to run the application –

*java -jar <JarFileName>.jar*

1. Start the postman and trigger a GET request on http://<hostname>:<port>/api/users

# Test Results

Attached are the Test Results