


# Open Number Management Design Document

Document Author:	Krzysztof Andryjowicz
Document Owner:	Krzysztof Andryjowicz
Document Version:	0.1
Document Status:	Draft in progress
Created on:	2018-02-09
Last Modification:	2018-02-09
File name:	<i>Open_Number_Management_Design_Document_v0.1.doc</i>

## Document History

Date	Author	Summary of changes	Version
2018-02-09	Krzysztof Andryjowicz	Document created	0.1

## Attachments

Name	Attachment
MySQL script	 Open_Number_Management_DB_mysql_create.sql

## Glossary

Term	Definition
ONM	Open Number Management
DB	Database
API	Application Program Interface
REST	An architecture style for designing networked applications
GUI	Graphical User Interface
MSISDN	A number uniquely identifying a subscription in a GSM or a UMTS mobile network
IMSI	International Mobile Subscriber Identity
SIM	A subscriber identity module, is a smart card that stores data for GSM cellular telephone subscribers, e.g. user identity, location and phone number, network authorization data, personal security keys, contact lists and stored text messages

## Table of contents

1. Developer's note .....	4
2. Document purpose .....	5
3. Project Phases approach .....	5
4. Functionality .....	6
5. Architecture .....	8
6. Database .....	9
7. REST APIs.....	10
8. Unit Tests approach.....	11
9. Open Issues .....	12

DRAFT

## 1. DEVELOPER'S NOTE

I build this application to meet the following personal goals:

1. To make practice of building web application with Java Spring framework
2. To connect my 10+ years IT consultant working experience in Telecom Industry, very small part of Business needs of this market area, and my need to build Business Support application from the scratch
3. To practice using of modern Java Development tools, mostly Spring framework
4. Have a piece of code to which I have intellectual rights ;-)
5. For fun ;-)

DRAFT

## 2. DOCUMENT PURPOSE

The purpose of this document is to describe high level of business needs which are covered and technical design details of Open Number Management Java based web application.

## 3. PROJECT PHASES APPROACH

I plan to deliver this project in 3 phases. Please find list of phases and scope of each of them:

1. Phase I /fire phase/
  - a. Very limited number of functional requirements /exposes only via APIs/
    - i. User Management, including roles and permissions
    - ii. Resource Types Management
    - iii. Resource Lifecycle Management
    - iv. Resource Management
  - b. REST APIs to expose 100% coverage of functionality listed in 1a
2. Phase II /UI phase/
  - a. Graphical User Interface to consume REST APIs delivered in 'fire phase'
3. Phase III /Business phase/
  - a. Introduce new functionality...

Unfortunately, I'm not able to fix on any delivery date, and I know this is a high risk issue which can cause I will never finish this project but I have to accept this risk ;- ) and I will do my best to manage this ;- )

Please note that I build this project being in line with Open Source License. Any collaboration are welcome and strongly desired ☺

## 4. FUNCTIONALITY

Open Number Management is design to manage of Number resources, e.g. SIM Card, MSISDN, IMSI. We can distinguish the following functional areas:

1. User management
  - a. Add User
  - b. Modify User:
    - i. Change User Details, including: personal details, assigned role
    - ii. Lock/unlock user
    - iii. Change password
  - c. Get User
2. Access management
  - a. Role management
    - i. Add role
    - ii. Delete role
    - iii. Modify role
    - iv. Get role
  - b. Permission management
    - i. Add permission
    - ii. Delete permission
    - iii. Modify permission
    - iv. Get permission
  - c. Permission to resource type management
    - i. Add resource type to permission
    - ii. Delete resource type from permission
    - iii. Get resource types by permission
  - d. Role to permission management
    - i. Add permission to role
    - ii. Delete permission from role
    - iii. Get permissions by role
3. Resource type management
  - a. Add resource type
  - b. Delete resource type
  - c. Modify resource type
  - d. Get resource type
4. Resource statuses management
  - a. Add resource status
  - b. Delete resource status
  - c. Modify resource status
  - d. Get resource status
5. Resource lifecycle management
  - a. Add resource status transition
  - b. Remove resource status transition
  - c. Get possible resource status transitions by source resource status
6. Resource management
  - a. Resource instance management
    - i. Add resource
    - ii. Modify resource, including: change status, related resource
    - iii. Get resource by:
      1. Resource type name
      2. Resource name

- b. Resource instance history management
  - i. Add history entry
  - ii. Get history entries by resource name/id

DRAFT

## 5. ARCHITECTURE

DRAFT



## 6. DATABASE

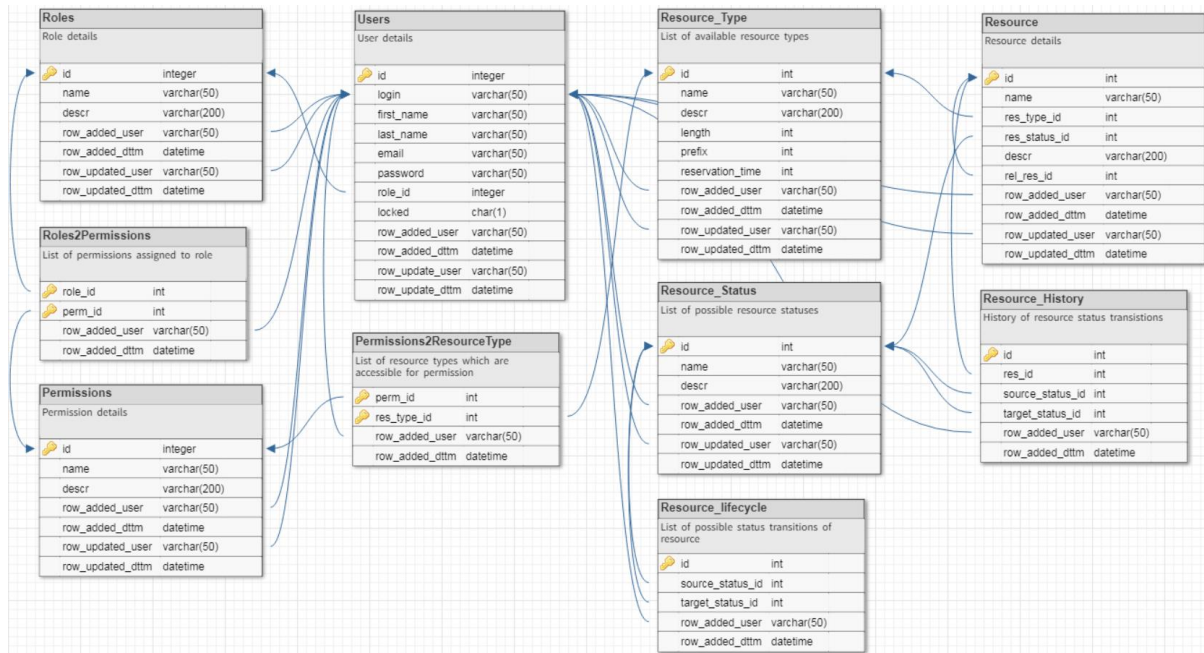


Figure 1. Relational Database Model

## 7. REST APIS

DRAFT

## 8. UNIT TESTS APPROACH

DRAFT

## 9. OPEN ISSUES

Ref	Reported by	Status	Description	Comment
1				

DRAFT