**High Level Design**

for

**My Social Connections App**



AUTHOR: **SANDEEP PAPUDESI**

EMAIL ID: **SANDEEP.PAPUDESI@YAHOO.COMC**

**TABLE OF CONTENTS**

[1. iNTRODUCTION 4](#_Toc521774643)

[1.1 In Scope 4](#_Toc521774644)

[1.2 Out of Scope 4](#_Toc521774645)

[1.3 Assumptions 4](#_Toc521774646)

[1.4 Dependencies 4](#_Toc521774647)

[2. Realized Design decissions 5](#_Toc521774648)

[3. Solution view 6](#_Toc521774649)

[3.1 Retrieve users with highest connections 7](#_Toc521774650)

[3.2 Retrieve users with lowest connections 7](#_Toc521774651)

[3.3 Retrieve total number of connections for Lucas (id 4) 7](#_Toc521774652)

[3.4 Total number of connections between two users 8](#_Toc521774653)

[3.5 Find Path between two users 8](#_Toc521774654)

[4. Visual UI 9](#_Toc521774655)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date of Revision** | **Description of Change** | **Author** |
| 0.1 | 08/10/2018 | Initial Draft | Sandeep Papudesi |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# iNTRODUCTION

The purpose of this document is to present a high-level design to address user stories provided by CapitalOne.

1. Which user has highest connections
2. Which user has lowest connections
3. How many total connections Lucas(id=4) has?
4. How many connections are there between two users
5. Who can introduce Lucas to Adam(id=62)
6. Build a web-based application to support above capabilities (Optional)

## In Scope

NA

## Out of Scope

* UI Validations
* E2E Exception Handling

## Assumptions

Solution will address all direct connections for user stories mentioned in Sec 1.

## Dependencies

NA

# Realized Design decissions

Following table provides the set of design decisions that have been realized as part of implementation

|  |  |  |
| --- | --- | --- |
| S. No | Requirement | Realized Design Decision |
| 1 | Technology Stack | **Decision:**  **Rational:** |
| 2 | Angular JS vs Angular 6 |  |
| 3 | Designed in generic | The application is designed to address more generic as compared to some user stories |
| 4 | Read the files each time to process | Appropriate Caching solution will be applied as part of the requirement |
| 5 | All User stories are exposed as REST services |  |

# Solution view

Below sequence diagram provides detail sequence of events to retrieve users with highest number of connections. The flow for rest of the services is going to be similar and will not be covered as part of this document.

As part of the design, application reads Persons.txt and Relationships.txt and associates the relationship in Person object and store the information in HashMap with key as “id” and values as “Person” object which internally contains Id, Name and List of connections.

All user stories are exposed as REST services

Web page URL : http://localhost:8080/



**Sample REST services output**

**"Output"**: [    
       {    
          **"id"**:"1",  
         **"name”**: “Jakson",  
          **"connections"**:"[ 3, 18, 30, 38, 49, 52, 57, 63, 70, 89]"  
 },  
       {    
          **"id"**:"4",  
         **"name”**: “Lucas",  
          **"connections"**:"[ 7, 27, 37, 38, 47, 51, 61, 79, 81, 95]"  
 }…

      ]

## Retrieve users with highest connections

## 

**URL: GET**: http://{server}:{port}/socialconnect/max

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Parameters** | **Data Type** | **Mandatory Attribute** | **Description** |
| NA | NA | NA | NA |

## Retrieve users with lowest connections

**URL:**  [**GET**: http://{server}:{port}/socialconnect/min](%20GET:%20http://%7bserver%7d:%7bport%7d/socialconnect/min)

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Parameters** | **Data Type** | **Mandatory Attribute** | **Description** |
| NA | NA | NA | NA |

## Retrieve total number of connections for Lucas (id 4)

**URL:**  [**GET**: http://{server}:{port}/socialconnect/{id}](%20GET:%20http://%7bserver%7d:%7bport%7d/socialconnect/%7bid%7d)

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Parameters** | **Data Type** | **Mandatory Attribute** | **Description** |
| Id | Integer | Yes | User Id to retrieve total connections |

## Total number of connections between two users

**URL:**  [**GET**: http://{server}:{port}/socialconnect/{fromId}/{toId](%20GET:%20http://%7bserver%7d:%7bport%7d/socialconnect/%7bfromId%7d/%7btoId)}

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Parameters** | **Data Type** | **Mandatory Attribute** | **Description** |
| fromId | Integer | Yes | From Id to retrieve common connections |
| toId | Integer | Yes | To Id to retrieve common connections |

## Find Path between two users

**URL:**  [**GET**: http://{server}:{port}/socialconnect/findpath/{fromId}/{toId}](%20GET:%20http://%7bserver%7d:%7bport%7d/socialconnect/findpath/%7bfromId%7d/%7btoId%7d)

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Parameters** | **Data Type** | **Mandatory Attribute** | **Description** |
| fromId | Integer | Yes | From Id to retrieve common connections |
| toId | Integer | Yes | To Id to retrieve common connections |

# Visual UI

Belowscreenshots depicts look and feel of all user stories

