**HMIS REST API High Level Design**

Draft 0.1

Last Updated: January 25, 2015

**Abstract**

This document describes the scope, high level design, database mappings and API specifications for the HMIS.

Contents

[2 Version History 3](#_Toc410044315)

[3 References 3](#_Toc410044316)

[4 Abbreviations and Acronyms 3](#_Toc410044317)

[5 Assumptions, Issues and Risks 4](#_Toc410044318)

[5.1 Assumptions 4](#_Toc410044319)

[5.2 Issues 4](#_Toc410044320)

[5.3 Risks 4](#_Toc410044321)

[6 Introduction 5](#_Toc410044322)

[6.1 Scope 5](#_Toc410044323)

[6.2 Out of Scope 5](#_Toc410044324)

[7 Solution Design 5](#_Toc410044325)

[7.1.1 Outreach Mobile UI 5](#_Toc410044326)

[7.1.2 OAuth Provider 5](#_Toc410044327)

[7.1.3 Service 5](#_Toc410044328)

[7.1.4 Database 5](#_Toc410044329)

[7.2 Security Model 5](#_Toc410044330)

[7.3 Interface Specification and data mappings 6](#_Toc410044331)

[7.3.1 Generic Response Type 6](#_Toc410044332)

[7.3.2 Authentication and Authorization Service 6](#_Toc410044333)

[7.3.3 Client Service 6](#_Toc410044334)

[7.3.4 Gender Service 10](#_Toc410044335)

[7.3.5 Race Service 12](#_Toc410044336)

[7.3.6 Ethnicity Service 15](#_Toc410044337)

[8 Implementation Strategy 17](#_Toc410044338)

[8.1 Technology Selection 17](#_Toc410044339)

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| Draft 0.1 | 01-25-2015 | Ashaar Riaz | Initial Draft |

Table 1: Version History

# References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No. | Version | Source | Author | Document Name |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table 2: References

# Abbreviations and Acronyms

|  |  |
| --- | --- |
| Abbreviation/Acronym | Description |
| API | Application Programming Interface |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 3: Abbreviations and Acronyms

# Assumptions, Issues and Risks

## Assumptions

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | Assumption | Comments | Status |
|  |  |  | Closed |
|  |  |  |  |
|  |  |  |  |

Table 4: Assumptions

## Issues

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | Assumption | Comments | Status |
|  |  |  |  |
|  |  |  |  |

Table 5: Issues

## Risks

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | Assumption | Comments | Status |
|  |  |  |  |
|  |  |  |  |

Table 6: Risks

# Introduction

Outreach Mobile is a mobile web application primarily focused in

## Scope

* All use cases specified in Outreach\_SRS\_v1.1.pdfdocument.
* The layout design is based on the approved design mockup. The same layout will be used for phone and tablet.
* Outreach Mobile Web application Web Services design, implementation and deployment.
* Outreach UI design, Implementation and deployment.

## Out of Scope

* Layout for desktop is not in the current scope of work.
* Database Design

# Solution Design

### Outreach Mobile UI

Web container to publish HTML screens.

### OAuth Provider

OAuth 2.0 provider.

### Service

Outreach Mobile Web Service container

### Database

MySQL Cloud database.

## Security Model

## Interface Specification and data mappings

### Generic Response Type

All response to the APIs in the subsequent sections will have the following structure for the web service response.

{

"header": {

"code": 0,

"message": "",

"total\_count": 234

},

"body": [

Array of Requested Data Structure

]

}

The code field identifies whether the response is success or failure. If the value of code is 0 then it is a success else it is a failure. The following is a sample failure response. Only header is sent for failure scenarios.

{

"header": {

"code": 1001,

"message": "No records found",

}

}

|  |  |  |
| --- | --- | --- |
| Field | Description | Mandatory(M)/Optional(O) |
| header | A generic response header | M |
| code | Response Code  0 – Success  >0 – Failure | M |
| message | Description of failure/error. | O |

Table 7: Generic Response Header

### Authentication and Authorization Service

All web services are authenticated and authorized using the OAuth 2.0 implicit grant flow. Please refer section 7.2 for the OAuth 2.0 implicit grant flow.

### Client Service

The following sample XML/JSON represents a ClientDetailVO object.

<clientDetailVO>

<clientKey>75864</clientKey>

<dateOfBirth>1965-09-11</dateOfBirth>

<ethnicityVO>

<codeKey>105</codeKey>

<description>Hispanic</description>

<logDateTime>2014-12-18 20:26:02</logDateTime>

<recActiveGct>1</recActiveGct>

</ethnicityVO>

<genderVO>

<codeKey>1</codeKey>

<description>Male</description>

<logDateTime>2014-07-31 19:59:34</logDateTime>

<recActiveGct>1</recActiveGct>

</genderVO>

<logDateTime>2015-01-04 02:46:18</logDateTime>

<nameFirst>Joe</nameFirst>

<nameLast>Moss</nameLast>

<nameMiddle>A</nameMiddle>

<socSecNumber>37669885</socSecNumber>

</clientDetailVO>

{

clientDetailVO :

{

“clientKey”:75864,

“dateOfBirth”:” 1965-09-11”,

“logDateTime”:” 2015-01-04 02:46:18”,

“nameFirst”:”Joe”,

“nameLast”:”Moss”,

“nameMiddle”:”A”,

“socSecNumber”:”37669885”

ethnicityVO:

{

“codeKey”:105,

“description”:”Hispanic”,

“logDateTime”:” 2014-12-18 20:26:02”,

“recActiveGct”:1

}

genderVO:

{

“codeKey”:1,

“description”:”Male”,

“logDateTime”:” 2014-12-18 20:26:02”,

“recActiveGct”:1

}

}

}

<clientVO>

<clientKey>75864</clientKey>

<dateOfBirth>1965-09-11</dateOfBirth>

<ethnicityDescription>Hispanic</ethnicityDescription>

<genderDescription>Male</genderDescription>

<nameFirst>Joe</nameFirst>

<nameLast></nameLast>

<nameMiddle>A</nameMiddle>

<socSecNumber>37669885</socSecNumber>

</clientVO>

{

clientVO :

{

“clientKey”:75864,

“dateOfBirth”: 1965-09-11,

“ethnicityDescription”: “Hispanic”,

“genderDescription”:”Male”,

“nameFirst”:”Joe”,

“nameLast”:””,

“nameMiddle”:”A”,

“socSecNumber”:” 37669885”

}

}

#### Add New Client

|  |  |
| --- | --- |
|  |  |
| END POINT | /clients/addClient/<emailID>/<serviceAccountID> |
| HTTP METHOD | POST |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| REQUEST XML | CLIENTDETAILVO [Ref: 7.5.3.1] |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  <CLIENTDETAILVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Failed to add"  }  } |

#### Update Existing Client Information

|  |  |
| --- | --- |
|  |  |
| END POINT | /clients/updateClient/<ClientKey>/<emailID>/<serviceAccountID> |
| HTTP METHOD | PUT |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| REQUEST XML | CLIENTDETAILVO\_OBJECT [Ref: 7.5.3.1] |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  <CLIENTDETAILVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Failed to update"  }  } |

#### Lookup Client by ClientKey

|  |  |
| --- | --- |
|  |  |
| END POINT | /clients/client/<ClientKey>/<emailID>/<serviceAccountID> |
| HTTP METHOD | GET |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json,application/xml  Accept:application/json,application/xml |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  },  "body": [  {  <ClientVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  },  "body": [  {  <clientVO\_OBJECT>  }  ]  } |
|  |  |

|  |  |
| --- | --- |
|  |  |
| END POINT | /client/clientdetail/<ClientKey>/<emailID>/<serviceAccountID> |
| HTTP METHOD | GET |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| REQUEST JSON | CLIENT\_OBJECT [Ref: 7.5.3.1] |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  <CLIENTDETAILVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Failed to add"  }  } |

### Gender Service

The following sample JSON represents a Gender object (GENDERVO\_OBJECT).

<genderVO>

<codeKey>1</codeKey>

<description>Male</description>

<logDateTime>2014-07-31 19:59:34</logDateTime>

<recActiveGct>1</recActiveGct>

</genderVO>

{

“codeKey”:1,

“description”:”Male”,

“logDateTime”:” 2014-12-18 20:26:02”,

“recActiveGct”:1

}

#### Add New Gender

|  |  |
| --- | --- |
|  |  |
| END POINT | /gender/addGender/<emailID>/<serviceAccountID> |
| HTTP METHOD | POST |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| REQUEST XML/JSON |  |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  <GENDERVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Failed to add"  }  } |

#### Update Gender

|  |  |
| --- | --- |
|  |  |
| END POINT | /gender/updateGender/<emailID>/<serviceAccountID> |
| HTTP METHOD | PUT |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| REQUEST XML/JSON | GENDERVO\_OBJECT [Ref: 7.5.4.1] |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  <GENDERVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Failed to update"  }  } |

#### Lookup All Gender

|  |  |
| --- | --- |
|  |  |
| END POINT | /genders/gender/<emailID>/<serviceAccountID> |
| HTTP METHOD | GET |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  LIST of <GENDERVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Not Found"  }  } |

### Race Service

The following sample XMLJSON represents a Date of Engagement object (DOE\_OBJECT).

{

"DateOfEngagementID": "4321s",

"ProjectEntryID": "w5641",

"DateOfEngagement": "2016-04-17"

}

#### Add New Race

|  |  |
| --- | --- |
|  |  |
| END POINT | /races/addRace/<emailID>/<serviceAccountID> |
| HTTP METHOD | POST |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| REQUEST XML/JSON | RACEVO\_OBJECT [Ref:7.5.5.1]  {  <RACEVO\_OBJECT>  } |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  <RACEVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Failed to add"  }  } |

#### Update Existing Race

|  |  |
| --- | --- |
|  |  |
| END POINT | /races/updateRace/{emailID}/{serviceAccountID} |
| HTTP METHOD | PUT |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| REQUEST XML/JSON | RACEVO\_OBJECT [Ref: 7.5.5.1]  {  <RACEVO\_OBJECT>  } |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  <RACEVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Failed to update"  }  } |

#### Lookup All the Races from the system

|  |  |
| --- | --- |
|  |  |
| END POINT | /races/race/<emailID>/<serviceAccountID> |
| HTTP METHOD | GET |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  List <RACEVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Not Found"  }  } |

### Ethnicity Service

The following sample JSON and XML represents an EthnicityVO object (ETHNICITYVO\_OBJECT).

<ethnicityVO>

<codeKey>0</codeKey>

<description>(select)</description>

<logDateTime>2015-01-10 09:30:25</logDateTime>

<recActiveGct>1</recActiveGct>

</ethnicityVO>

{

“codeKey”:0,

“description”:”(select)”

“logDateTime”:” 2015-01-10 09:30:25”

“recActiveGct”:1

}

#### Get all ethnicities from the system

|  |  |
| --- | --- |
|  |  |
| END POINT | /ethnicities/ethnicity/{emailID}/{serviceAccountID} |
| HTTP METHOD | GET |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| REQUEST XML/JSON | ETHNICITYVO\_OBJECT [Ref:7.5.6.1]  {  <ETHNICITYVO\_OBJECT>  } |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  List <ETHNICITYVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Failed to add"  }  } |

#### Update Existing Ethnicity

|  |  |
| --- | --- |
|  |  |
| END POINT | /ethnicities /updateEthnicity/<emailID>/<serviceAccountID> |
| HTTP METHOD | PUT |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| REQUEST XML/JSON | ETHNICITYVO\_OBJECT [Ref: 7.5.6.1]  {  <ETHNICITYVO\_OBJECT>  } |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  <ETHNICITYVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Failed to update"  }  } |

#### Add Ethnicity

|  |  |
| --- | --- |
|  |  |
| END POINT | /ethnicities /addEthnicity/<emailID>/<serviceAccountID> |
| HTTP METHOD | GET |
| TRANSPORT | SSL |
| AUTHENTICATION | HTTP Header  Authorization: |
| HTTP HEADERS | Content-Type:application/json, application/xml  Accept:application/json, application/xml |
| SUCCESS RESPONSE | HTTP 200 OK  {  "header": {  "code": 0  },  "body": [  {  <ETHNICITYVO\_OBJECT>  }  ]  } |
| FAILURE RESPONSE | HTTP 200 OK  {  "header": {  "code": 1001,  "message": "Not Found"  }  } |

# Implementation Strategy

## Technology Selection

All tools and technologies that we use are either Open Source or available for free.

| Business Functionality | Technology | Reason |
| --- | --- | --- |
| Solution Platform |  |  |
| Web Services | Jersey REST API |  |
| OAuth Provider |  |  |
| Database Access |  |  |
| Database | MySQL Cloud Database |  |
| JEE Web Server | Tomcat 8.0.1 |  |
| Java Version | JDK 7.0 |  |
| JEE Version | J2EE 6 |  |

Table 8: Recommendation Service Technology Selection

**<<END OF DOCUMENT>>**