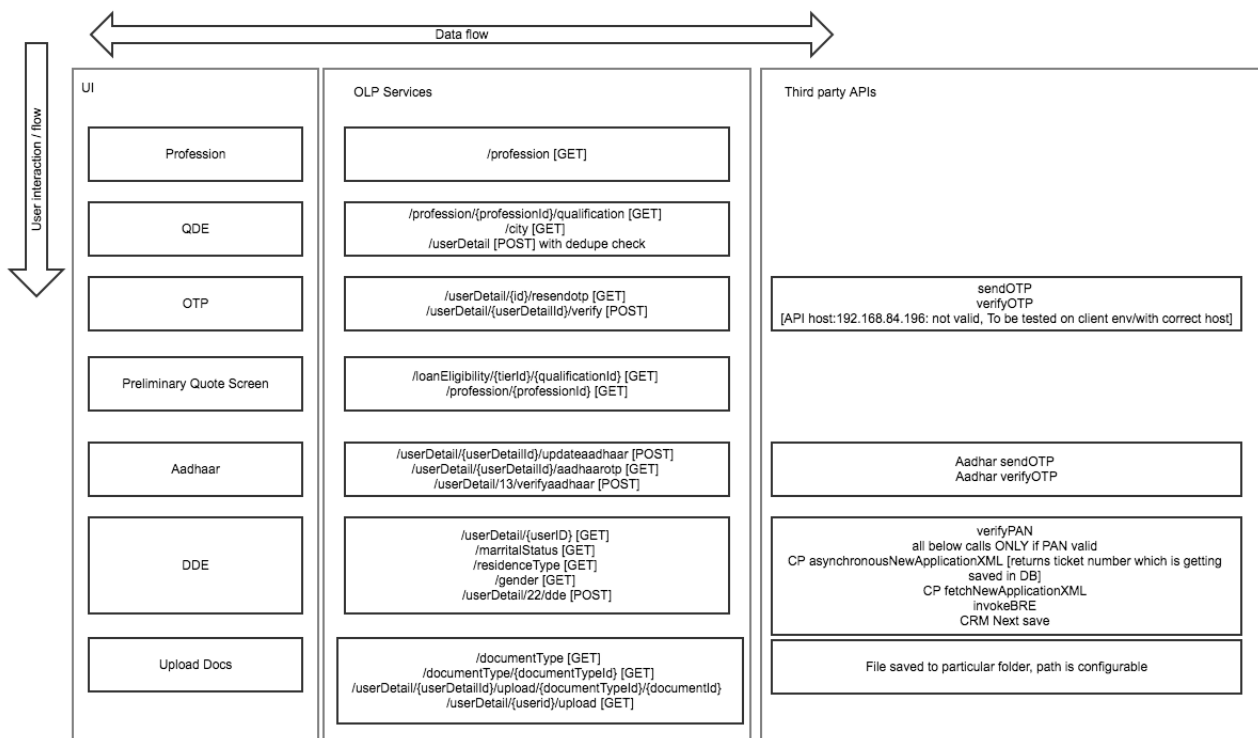


API Integration:



API Response decoding:

1. Below are the http codes used in API and their meaning.

406: Request object validation failed
 201: Object created
 200: Response OK
 404: Not Found
 500: Internal Server Error
 208: Duplicate Request

2. Response body:

- On all success status codes [2xx], response body is JSON response data
- On all failure status codes [4xx,5xx], response body is JSON data with error message
- For response returning list, which might require pagination [to be used in admin web], additional attribute "size" will be available in response JSON, and actual list will be in "value" attribute.

Landing Page:

To fetch the list of professions [profession id to be passed for QDE to fetch list of qualifications]

HTTP Method: GET

API: /profession

QDE:

Qualification dropdown:

HTTP Method: GET

API: /profession/1/qualification

Note: "1" in the above URL to be replaced by profession id. [Master data added for doctor's only]

City dropdown:

HTTP Method: GET

API: /city

Note: response has tier object for each city. Tier Id along with qualification id to be passed for getting loan eligibility on Preliminary quote screen.

Submit QDE form:

HTTP Method: POST

API: /userDetail

Response codes:

406: Request object not valid

208: Record already in DB [dedupe-check]

Note: this API 1. Saves data in MySQL , 2. Calls OTP API to get OTP. Response consists of "id" which is to be sent in OTP verification API

Sample request body:

```
{
  "name": "test doctor name",
  "active": "true",
  "email": "test@asdf.com",
  "mobile": "31234124",
  "city": {"id": 1},
  "qualification": {"id": 1},
  "profession": {"id": 1},
  "professionalDetail": {
    "workAddress": {
      "entityName": "test hospital name",
      "workingSince": "2015-01-24"
    }
  }
}
```

OTP verification screen:

OTP Verification:

HTTP Method: POST

API: /userDetail/{id}/verify

Note: this API calls OTP verification API. On Success Response calls of Fullerton OTP API, 1. updates status in DB, 2. Calls CP WS, updates response ticket number in DB on successful CP WS response.

Sample request:

```
{  
  "otp": "1234"  
}
```

Resend OTP:

HTTP Method: GET

API: /userDetail/{id}/resendotp

Preliminary Quote Screen:

HTTP Method: GET

API: /loanEligibility/{tierId}/{qualificationId}

To get loan tenure, use profession.loanDuration

HTTP Method: GET

API: /profession/{professionId}

Aadhaar Screen:

To save Aadhaar for user:

HTTP Method: POST

API: /userDetail/{userId}/updateaadhaar

Sample request:

```
{  
  "aadhaarNumber": "123441231234"  
}
```

To generate verification OTP for Aadhaar:

HTTP Method: GET

API: /userDetail/{userId}/aadhaarotp

To verify Aadhaar OTP

HTTP Method: POST

API: /userDetail/13/verifyaadhaar

Sample request:

```
{
  "otp": "1234"
  "consent": "<appropriate consent>"
}
```

Detailed Data Entry:

To get saved user detail for pre-populating already entered data.

HTTP Method: GET

API: /userDetail/{userID}

HTTP Method: GET

API: /marritalStatus

HTTP Method: GET

API: /residenceType

HTTP Method: GET

API: /gender

HTTP Method: POST

API: /userDetail/22/dde

Sample request:

```
{
  "id": {userDetailId},

  "professionalDetail": {
    "id": {professionalDetailId, if not passed, will create new professional detail},
    "annualIncome": 100000,
    "monthlyIncome": 11000,
    "otherIncome": 10000000,
    "panNumber": "asfasdfas",
    "registrationNumber": "asdf452",
    "workAddress": {
      "id": {workAddressId},
      "address1": "address line one work..",
      "address2": "line 222",
      "address3": null,
      "landline": "02012341234",
      "city": {"id": 1},
      "entityName": "test hospital 222222",
      "workingSince": 1422057600000
    }
  },
  "personalDetail": {
```

```
"dob":"2015-01-24",
"upild":"asdf1234",
"gender":{"id":1},
"maritalStatus":{"id":1},
"residenceAddress":{"
  "address1": "address line one home..",
  "address2": "asf",
  "address3": "assadfas",
  "city":{"id":1},
  "livingSince":"2015-01-24",
  "residenceType":{"id":1}
}
}
```

Document uploads:

Fetch all document types:

HTTP Method: GET

API: /documentType

Fetch all documents for a document type:

HTTP Method: GET

API: /documentType/{documentTypeId}

Upload a file:

HTTP Method: POST

API: /userDetail/{userId}/upload/{documentTypeId}/{documentId}

Get files uploaded for a user:

HTTP Method: GET

API: /userDetail/{userId}/upload