

Mobile Financial Services

System Design

chapaa

Version 0.1

November 2017

Prepared by:

Rodrigue Kitindi Rumisho Shikonyi

**Document Approval**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **NAME** | **DATE** | **SIGNATURE** |
| Checked by |  |  |  |
|  |  |  |
| Approved by |  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **DATE** | **VERSION** | **DESCRIPTION** | **AUTHOR** |
| 11-Nov-2017 | 0.1 | Initial draft | Rumisho Shikonyi  Rodrigue Kitindi |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 System Overview 5](#_Toc498197484)

[2 Frontend 6](#_Toc498197485)

[2.1 Mobile Application for Customers 6](#_Toc498197486)

[2.1.1 Login 6](#_Toc498197487)

[2.1.2 Register 7](#_Toc498197488)

[2.1.3 Activate 9](#_Toc498197489)

[2.1.4 Home 10](#_Toc498197490)

[2.2 Mobile Application for Agents 12](#_Toc498197491)

[2.3 Mobile Application for Merchants 12](#_Toc498197492)

[2.4 Web Application 12](#_Toc498197493)

[3 Integration 13](#_Toc498197494)

[4 Backend 14](#_Toc498197495)

[4.1 Database 14](#_Toc498197496)

[4.1.1 General Setup 14](#_Toc498197497)

[4.1.2 User Setup 15](#_Toc498197498)

[4.1.3 Account Setup 16](#_Toc498197499)

[4.1.4 Payment Setup 18](#_Toc498197500)

[4.1.5 Product Setup 19](#_Toc498197501)

[4.1.6 Reward Setup 20](#_Toc498197502)

[4.1.7 Referral Setup 21](#_Toc498197503)

[4.1.8 Brokerage Setup 22](#_Toc498197504)

[5 Application Programming Interface (API) 24](#_Toc498197505)

[5.1 Convention 24](#_Toc498197506)

[5.1.1 Specification 24](#_Toc498197507)

[5.1.2 Authentication 24](#_Toc498197508)

[5.1.3 Header 25](#_Toc498197509)

[5.1.4 Status Code 25](#_Toc498197510)

[5.2 User API 25](#_Toc498197511)

[5.2.1 Email Validation 25](#_Toc498197512)

[5.2.2 MSISDN Validation 27](#_Toc498197513)

[5.2.3 User Registration 29](#_Toc498197514)

[5.2.4 Member Creation 32](#_Toc498197515)

# System Overview

# Frontend

## Mobile Application for Customers

### Login

|  |  |
| --- | --- |
|  | This screen is the gateway to chapaa mobile application, enabling access to the platform. |

#### Input

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S/N | FIELD NAME | TYPE | SIZE | REQ. | DESCRIPTION |
| 1 | Phone number | Numeric | 12 | Y | Main mobile number of the customer |
| 2 | PIN | Numeric | 4 | Y | Secret Personal Identification Number |

#### Validation

|  |  |  |
| --- | --- | --- |
| S/N | ACTION | DDESCRIPTION |
| 1 | Phone number | Must be 9-digit long (excluding **country code**, **0**, **00**, or **+**) |
| 2 | PIN | Must be 4-digit long |

#### Action

|  |  |  |
| --- | --- | --- |
| S/N | ACTION | DDESCRIPTION |
| 1 | Login | Submit customer’s login request to the platform |
| 2 | Register | Redirects the customer to [Register](#_Register) |
| 3 | Lost a PIN | Redirects the customer to Retrieve PIN |

#### Process Flow

### Register

|  |  |
| --- | --- |
|  | This screen allows the registration of a new customer. |

#### Input

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S/N | FIELD NAME | TYPE | SIZE | REQ. | DESCRIPTION |
| 1 | Full name | Text | - | Y | customer’s full name |
| 2 | Phone number | Numeric | 12 | Y | Phone number |
| 3 | Address | Text | - | N | Customer’s address |
| 4 | Security question 1 | Text | - | N | Security question to be part of security questions asked during PIN/account recovery process |
| 5 | Security question 2 | Text | - | N | Security question to be part of security questions asked during PIN/account recovery process |
| 6 | PIN | Numeric | 4 | Y | 4-digit secret code |

#### Validation

|  |  |  |
| --- | --- | --- |
| S/N | ACTION | DDESCRIPTION |
| 1 | Full name | Must not be empty |
| 2 | Phone number | Must be 9-digit long (excluding **country code**, **0**, **00**, or **+**). It must be unique |
| 3 | PIN | Must be 4-digit long |

#### Action

|  |  |  |
| --- | --- | --- |
| S/N | ACTION | DDESCRIPTION |
| 1 | Register | Submit customer’s registration request |
| 2 | Already registered | Redirects the customer to [Login](#_Login) |

#### Process Flow

### Activate

This screen activates a customer account. A customer cannot have access to the system until the account has been activated.

#### Input

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S/N | FIELD NAME | TYPE | SIZE | REQ. | DESCRIPTION |
| 1 | Activation code | Numeric | 6 | Y | customer’s full name |

#### Validation

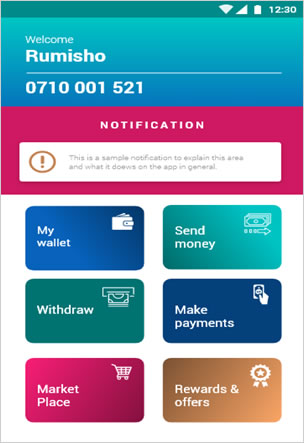
|  |  |  |
| --- | --- | --- |
| S/N | ACTION | DDESCRIPTION |
| 1 | Activation code | Must be 9-digit long |

#### Action

|  |  |  |
| --- | --- | --- |
| S/N | ACTION | DDESCRIPTION |
| 1 | Activate | Submit customer’s activation request |
| 2 | Already registered | Redirects the customer to [Login](#_Login) |

#### Process Flow

### Home



This is the welcome screen.

#### Input

None

#### Validation

None

#### Action

|  |  |  |
| --- | --- | --- |
| S/N | ACTION | DDESCRIPTION |
| 1 | My wallet | Opens My wallet screen |
| 2 | Send money | Opens Send money screen |
| 3 | Withdraw | Opens Withdraw screen |
| 4 | Make payments | Opens Payment screen |
| 5 | Marketplace | Opens Marketplace screen |
| 6 | Rewards & offers | Opens Rewards & offers screen |

# Integration

# Backend

## Database

### General Setup

#### Exchange Rate

CREATE TABLE mfs\_exchange\_rate

{

id bigserial NOT NULL,

}

### User Setup

#### Locale

CREATE TABLE chapaa.locale

{

id serial NOT NULL,

code character varying(2) NOT NULL,

lcode character varying(7) NOT NULL,

name character varying(100) NOT NULL,

description character varying(255),

CONSTRAINT locale\_pkey PRIMARY KEY(id),

CONSTRAINT locale\_code\_unique UNIQUE(code),

CONSTRAINT locale\_lcode\_unique UNIQUE(lcode)

}

#### Frequency

#### Title

#### Contact Type

#### Organisation Type

#### Business Industry

#### Currency

#### Address Type

#### Country

#### Address

#### Member

#### Member Contact

#### Member Address

#### User

#### Group

#### Group Member

# Application Programming Interface (API)

## Convention

### Specification

All APIs will adopt the [JavaScript Object Notation (JSON) API Specification](http://jsonapi.org/format/)., one that is already registered with the [Internet Assigned Numbers Authority (IANA)](https://www.iana.org/assignments/media-types/application/vnd.api+json).

### Authentication

### Header

The head must contain the media type as well as the access token supplied after successful authentication.

"Content-Type: application/vnd.api+json"

"Authorization: Bearer <Access-Token>"

### Status Code

A response will return a valid HTTP Status Code, as documented in the [Hypertext Transfer Protocol (HTTP) Status Code Registry](http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml).

Errors will be returned in the following format:

{

"errors": [

{

"status": "422",

"source": {

"pointer": "/data/attributes/first-name"

},

"title": "Invalid Attribute",

"detail": "First name must contain at least three characters."

}

]

}

## User API

### Email Validation

API used to validate an email address that is provided during user registration or change of user information such as email address.

#### URI

/users/emailValidation

#### Request

{

"email": "kitindir@gmail.com"

}

#### Business Logic

##### Process Flow

##### Validation

^([a-zA-Z0-9\_\-\.]+)@([a-zA-Z0-9\_\-\.]+)\.([a-zA-Z]{2,5})$

#### Child Processes

None

#### Database

##### Relations

* [User](#_User)

##### Queries

SELECT

TRIM(email) as email

FROM

chapaa.mfs\_user

WHERE

email = @email

#### Response

{

"valid": "1"

}

### MSISDN Validation

API used to validate a MSISDN (or phone number) that is provided during user registration or change of user information such as main telephone number.

#### URI

/users/msisdnValidation

#### Request

{

"msisdn": "0713123672"

}

#### Business Logic

##### Process Flow

##### Validation

* Check whether the MSISDN is numeric only

#### Child Processes

None

#### Database

##### Relations

* [Config](#_Config)
* [User](#_User)

##### Queries

* Get country code of the user’s country from the database

SELECT

value

FROM

chapaa.mfs\_config

WHERE

key = ‘app\_country\_code’

* Get required length for a MSISDN in the current country

SELECT

value

FROM

chapaa.mfs\_config

WHERE

key = ‘app\_msisdn\_length’

* Check whether there is a record for that MSISDN

SELECT

msisdn

FROM

chapaa.mfs\_user

WHERE

msisdn = @msisdn

#### Response

{

"valid": "1"

}

### User Registration

API used to register a new user in the platform. This API will:

* Create a new user
* Create a new Person (member)
* Create an account

#### URI

/users

#### Request

{

"firstname": "Rodrigue",

"lastname": "Kitindi",

"business\_name": "",

"is\_business": "false",

"email": "kitindir@gmail.com",

"msisdn": "0713123672",

"pin": "",

"language": "en",

"title": "1",

"business\_industry": "1",

"organisation\_type": "1",

"address": "MRC Building, Sinza Mapambano, Dar es salaam",

"country": "TZ"

}

#### Business Logic

##### Process Flow

##### Validation

* Check whether the MSISDN is numeric only

#### Child Processes

* [Email Validation](#_Email_Validation)
* [MSISDN Validation](#_MSISDN_Validation)
* [Member Creation](#_Member_Creation)
* Address Creation
* Account Creation

#### Database

##### Relations

* [Config](#_Config)
* [Locale](#_Locale)
* [User](#_User)

##### Queries

* Get the ID of user’s selected locale

SELECT

id

FROM

chapaa. mfs\_locale

WHERE

code = @locale

* Get application locale in case the user has not provided any locale, then use this to get its ID

SELECT

value

FROM

chapaa.mfs\_config

WHERE

key = ‘app\_locale’

* Get application default PIN if the user has not provided any PIN information

SELECT

value

FROM

chapaa.mfs\_config

WHERE

key = ‘account\_default\_pin’

* Insert user record in the database

INSERT INTO chapaa.mfs\_user ( member\_id, locale\_id, username, password, pin, email, msisdn )

VALUES ( {member\_id}, {locale\_id}, @msisdn, @pin, @pin, @email, @msisdn )

#### Response

{

"id": "1",

"username": "255713123672",

"email": "kitindir@gmail.com",

"msisdn": "255713123672",

"date\_created": "2017-10-27 18:08:12",

"member": {

"is\_business": "false",

"first\_name": "Rodrigue",

"last\_name": "Kitindi",

"display\_name": "Rodrigue Kitindi",

"business\_name": "",

"title": "Mr",

"business\_industry": "",

"organisation\_type": ""

}

}

### Member Creation

API used to create a new member (individual or organisation).

#### URI

/members

#### Request

{

"firstname": "Rodrigue",

"lastname": "Kitindi",

"business\_name": "",

"is\_business": "false",

"title": "1",

"business\_industry": "1",

"organisation\_type": "1",

"telephone": "255713123672",

"extension": "",

"mobile": "255713123672",

"email": "kitindir@gmail.com",

"tax\_identification": "",

"date": ""

}

#### Business Logic

##### Process Flow

##### Validation

* Business name cannot be left blank is member type is business
* First name and Last name cannot be left blank is member type is not business

#### Child Processes

None

#### Database

##### Relations

* [Member](#_Member)

##### Queries

* Insert Member record in the database

INSERT INTO chapaa.mfs\_member (title\_id, first\_name, last\_name, organisation\_name, organisation\_type\_id, business\_industry\_id, telephone, extension\_number, mobile, email, is\_organisation, tax\_identification, birth\_date, date\_established)

VALUES ( {member\_id}, {locale\_id}, @msisdn, @pin, @pin, @email, @msisdn )

* Get application locale in case the user has not provided any locale, then use this to get its ID

SELECT

value

FROM

chapaa.mfs\_config

WHERE

key = ‘app\_locale’