**Business Logic**

**Mobile Recharge**

**Data Stores:**

* **Recharge Request** data store => Before sending to Pay2All Create a Record for the transaction
* **Recharge Point Register** Data Store => Non Redeemable Transactions
* **Member Income** data store => Redeemable Transactions
* **Member Wallet** data store = Member current Balances

**Variables :**

**Reference Variables**

* #RECHARGE\_CASHBACK\_PERCENT#
* #ALLOW\_MOBILE\_RECAHRGE#
* #MOBILE\_WELCOME\_ADJUSTMENT\_PERCENT#
* #TEAM\_CASHBACK\_PERCENT#
* #PARENTS#

**Calculation Variables**

|  |  |
| --- | --- |
| #RECHARGE\_AMOUNT# | User entered the amount for recharge |
| #WELCOME\_BALANCE# | Remaining Welcome balance |
| #REDEEMABLE\_BALANCE# | Remaining Redeemable Balance |
| #NON\_REDEEMABLE\_BALANCE# | Remaining Redeemable Balance |
| #REQUESTID# | Unique ID to identify recharge transaction |
| #WELCOME\_DEDUCTION# | Actual deduction from Welcome balance |
| #REDEEMABLE\_DEDUCTION# | Actual deduction from Redeemable balance |
| #NON\_REDEEMABLE\_DEDUCTION# | Actual deduction from Non Redeemable balance |
| #REST\_DEDUCTION# | Deduction other than Welcome Amount |
| #CASHBACK\_AMT# | Cashback amount being given to the user |
| #EQUAL\_AMOUNT# | Amount to be credited to parents |
| #TEAM\_CASHBACK\_AMOUNT# | Calculated Team Cashback amount |
| #TEAM\_CASHBACK\_LEFTOVER# | After distributing Cashback amount to parents if a fraction is left then it is credited to Mansha Account in Recharge Point Register data store |

**Algorithm**

|  |  |
| --- | --- |
| Recharge Parameters are extracted | **#RECHARGE\_CASHBACK\_PERCENT#** -> Currently 2%  **#ALLOW\_MOBILE\_RECAHRGE#** -> Currently Allowed  **#MOBILE\_WELCOME\_ADJUSTMENT\_PERCENT#** -> Currently 1% (If chosen)  **#TEAM\_CASHBACK\_PERCENT#** -> Currently 25% |
| if **#ALLOW\_MOBILE\_RECAHRGE#** is true then abort the operation with message | |
| Extract all parents in **#PARENTS#** | |
|  | |
| Extract balances from **Member Wallet** | #WELCOME\_BALANCE#,  #REDEEMABLE\_BALANCE#,  #NON\_REDEEMABLEBALANCE# |
| **Calculate**  **#WELCOME\_DEDUCTION#** = #RECHARGE\_AMOUNT# X #MOBILE\_WELCOME\_ADJUSTMENT\_PERCENT# \* 0.01 | |
| If **#WELCOME\_BALANCE#** < **#WELCOME\_DEDUCTION#** then #WELCOME\_DEDUCTION#=0 | |
| **#REST\_DEDUCTION#** = #RECHARGE\_AMOUNT# - #WELCOME\_DEDUCTION# | |
| If **(#WELCOME\_BALANCE# + #REST\_DEDUCTION#)** > (#REDEEMABLE\_BALANCE# + #NON\_REDEEMABLEBALANCE#) abort the operation with message | |
| **Distribute Demand**  \* if **#REST\_DEDUCTION#** <= #NON\_REDEEMABLEBALANCE#  #NON\_REDEEMABLE\_DEDUCTION# = #REST\_DEDUCTION#  #REDEEMABLE\_DEDUCTION# = 0  else  #NON\_REDEEMABLE\_DEDUCTION# = #NON\_REDEEMABLEBALANCE#  #REDEEMABLE\_DEDUCTION# = #REST\_DEDUCTION# - #NON\_REDEEMABLE\_DEDUCTION# | |
| Add a record in **Recharge Request data store** and get #REQUESTID# | |
| Update Recharge Token from Pay2All Service into database and get Token | |
| Try to recharge from Pay2All Service if error abort with message | |
| Update parameters given from Pay2All to Recharge Request data store | |
| \* Update following calculated values in Member Wallet data store  **#WELCOME\_BALANCE#** = #WELCOME\_BALANCE# - #WELCOME\_DEDUCTION#  **#REDEEMABLE\_BALANCE#** = #REDEEMABLE\_BALANCE# - #REDEEMABLE\_DEDUCTION#,  **#NON\_REDEEMABLEBALANCE#** = #NON\_REDEEMABLEBALANCE# - #NON\_REDEEMABLE\_DEDUCTION# | |
| if **#NON\_REDEEMABLE\_DEDUCTION#** > 0 then update RechargePointRegister data store | |
| if **#REDEEMABLE\_DEDUCTION#** > 0 then update Member Income data store | |
| #**CASHBACK\_AMT#** = #RECHARGE\_AMOUNT# \* #RECHARGE\_CASHBACK\_PERCENT# \* 0.01 | |
| Update **#CASHBACK\_AMT#** in Recharge Point Register data store | |
| **#TEAM\_CASHBACK\_AMOUNT#** = #CASHBACK\_AMT# \* #TEAM\_CASHBACK\_PERCENT# \* 0.01 | |
| **#EQUAL\_AMOUNT#** = #TEAM\_CASHBACK\_AMOUNT# / Count of Parents | |
| **#TEAM\_CASHBACK\_LEFTOVER#** = #TEAM\_CASHBACK\_AMOUNT# - (#EQUAL\_AMOUNT# \* Count of Parents) | |
| Add amount to all the parent in Recharge Point Register data store | |
| **Success** | |
|  |  |

**Add Money**

**Execution Steps**

* **Get TxtID**
* **Recharge Point Register** Data Store => Non Redeemable Transactions

**Data Stores:**

* **Payment Gateway** => Store references to Razorpay
* **Member Deposits** => Added amounts are stored here
* **Member Wallet** data store = Member current Balances

**Variables:**

**Reference Variables**

* #MEMBERSHIP\_FEE#
* #TAX\_PERCENT#
* #CASHBACK\_REWARD#
* **#API\_KEY#**
* **#API\_SECRET#**

**Calculation Variables**

|  |  |
| --- | --- |
| #TIKETNO# | Unique string given to Razorpay for Identity |
| #ORDERID# | Unique Order ID given by Razorpay for Identity |

**Algorithm Get TxtID**

|  |  |
| --- | --- |
| Parameters are extracted | **#MEMBERSHIP\_FEE**# => Currently **1000**  **#TAX\_PERCENT#** => Currently **18%**  **#CASHBACK\_REWARD#** => Currently **30 points** |
| #TIKETNO# = I am Generate a Unique String to send to Razorpay to process a new order id  #ORDERID# = Razorpay returns Order ID for my reference | |
| Records are updated in Payment Gateway Data Store | |
| **Note: OrderID is referred to as TxnID throughout the code** | |
| **Success: TxnID is returned** | |

**Algorithm Update Payment**

|  |
| --- |
| **Valications:**   * Amount in multiple of 100 * Max Limit 5000 * Multiply amount with 10000 (this is how Razorpay accepts) |
| Send payment request to Razorpay with   * TxmID * Amount (Amount \* 10000) |
| If success the update Payment Gateway Store and follow below updates |
| Update Recharge Point Register for non redeemable amount addition |
| **Success** |

**Member Signup**

**Execution Steps**

* **Create Temp Member**
* **Confirm Membership**

**Data Stores:**

* **Recharge Request** data store => Before sending to Pay2All Create a Record for the transaction
* **Recharge Point Register** Data Store => Non Redeemable Transactions
* **Member Income** data store => Redeemable Transactions
* **Member Wallet** data store = Member current Balances

**Variables :**

**Reference Variables**

* #RECHARGE\_CASHBACK\_PERCENT#

**Calculation Variables**

|  |  |
| --- | --- |
| #MEMBERSHIP\_FEE# | Basic Membership Amount |
| #TAX\_PERCENT# | GST applicable |
| #CASHBACK\_REWARD# | Points added to Non Redeemable Balance |
| #LEVEL1\_LEADERSHIP\_INCOME# | Cut % from child child’s level income |
| #LEVEL2\_LEADERSHIP\_INCOME# | Cut % from grand child child’s level income |
| #ALLOW\_COMPANY\_REFERAL\_CODE# | Allow Mansha as Referrer |
| #ALLOW\_NEW\_MEMBERS# | Allow New members to be added |
|  |  |
|  |  |

**Algorithm (Create Temp Member)**

|  |  |
| --- | --- |
| Input Validation | First Name is Required  Last Name is Required  Mobile No is Required  Referal Code is Required  Email is Required |
| Signup parameters are extracted | #MEMBERSHIP\_FEE# [Currently 1000]  #TAX\_PERCENT# [Currently 18%]  #CASHBACK\_REWARD# [Currently 30]  #LEVEL1\_LEADERSHIP\_INCOME# [Currently 5%]  #LEVEL2\_LEADERSHIP\_INCOME# [Currently 3]  #ALLOW\_COMPANY\_REFERAL\_CODE# [Currently Yes]  #ALLOW\_NEW\_MEMBERS# [Currently Yes] |
| If **#ALLOW\_COMPANY\_REFERAL\_CODE#** is false then restrict referrer code of Mansha | |
| if **#ALLOW\_NEW\_MEMBERS#** = false then restrict new members | |
| If Referal Code is invalid then return with error message | |
| If Mobile No or email already exists then return with error message | |
| If Mobile no is present in Temporary Store then update else create new Member | |
| Fields to be updated in **Temp User** Data Store   * #FIRST\_NAME# * #LAST\_NAME# * #MOBILE\_NO# * #REFERAL\_CODE# * #EMAIL# * #PASSWORD# * #ADDRESS# * #PARENT\_ID# * #MEMBERSHIP\_FEE# * #TAX\_PERCENT# * #TAX\_AMOUNT# * #NET\_AMOUNT# * #EXPIRY\_AT# * #IP# | |
| Create Razorpay Temp Order  Generate OTP, If found Valid then | |
| Create Payment Records then  Accept Payment through Razorpay | |
| **When Success proceed to next algorithm** | |

**Algorithm (Update and Verify Member after Payment)**

|  |  |
| --- | --- |
| Validate Input parameters | 'txn\_id'  'temp\_id'  'payment\_id'  'status' |
| Check if TempID is present in Temp Members data store else return with error message | |
| If Payment Gateway data store does not have record for TempID and TxnID then return with error message | |
| Update Payment Gateway data store | |
| Extract Parameters | #MEMBERSHIP\_FEE# [Currently 1000]  #TAX\_PERCENT# [Currently 18%]  #CASHBACK\_REWARD# [Currently 30]  #LEVEL1\_LEADERSHIP\_INCOME# [Currently 5%]  #LEVEL2\_LEADERSHIP\_INCOME# [Currently 3]  #ALLOW\_COMPANY\_REFERAL\_CODE# [Currently Yes]  #ALLOW\_NEW\_MEMBERS# [Currently Yes] |
| Create Member User  **Member User** data store | Update following fields in **Member User** data store   * #NAME# * #MOBILE\_NO# * #PASSWORD# * #API\_TOKEN# [Random String for the member] * #ACTIVE# [true] |
| Create Member  **Members** data store | Create a record in Members data store   * #TEMP\_ID# * #MEMBER\_ID# * #PARENT\_ID# * #GRAND\_PARENT\_ID# * #UNIQUE\_ID# * #FIRST\_NAME# * #LAST\_NAME# * #ADDRESS# * #EMAIL# * #REFERAL\_CODE# * #MOBILE\_NO# * #JOINING\_DATE# |
| Populate parents from Member Map data store | |
| Create Member Map for the New Member. It has a list of following attributes   * MEMBER\_ID * PARENT\_ID * CURRENT\_LEVEL | |
| Add Record in Member Deposit data store | Create a record in Member deposits data store   * #MEMBER\_ID# * #GATEWAY\_ID# => Payment Gateway Transaction ID * #AMOUNT# * #TAX\_PERCENT# * #TAX\_AMOUNT# * #NET\_AMOUNT# * #DEPOSIT\_TYPE# => 'MEMBERSHIP\_FEE' |
| Populate Level Master |  |
| Populate Club Master | |
| Update Member Wallet data store | #MEMBER\_ID#  #TOTAL\_MEMBERS# => 0  #WELCOME\_AMT# => #MEMBERSHIPFEE#+#MEMBERSHIP\_POINTS#  #REDEEMABLE\_AMT# => 0  #NON\_REDEEMABLE# => 0  #LEVEL\_INCOME# => 0  #LEADERSHIP\_INCOME# => 0  #CLUB\_INCOME# => 0  #TRANSFERIN\_AMOUNT# => 0  #TRANSFEROUT\_AMOUNT# => 0 |
| Update Level Income data store |  |
|  |  |