# Tables

## UserDetails

|  |  |  |
| --- | --- | --- |
| UserId | Integer | PK |
| LoginType | Varchar(50) | NOT NULL (Facebook|Gmail|Custom) |
| UserName | Varchar(50) | NOT NULL |
| Password | Varchar(100) | NULL – this will be null for FB/Gmail login users |
| RoleId | Integer | FK – Roles |
| Status | Varchar(50) | NOT NULL – Default Active |
| DateCreated | Datetime | NOT NULL |
| EmailAddress | Varchar(250) | NULL |
| MobileNumber | Varchar(50) | NULL |

* Populate default Admin user, Password should be encrypted
* Default admin – super | Admin#234

## Roles

|  |  |  |
| --- | --- | --- |
| RoleId | Integer | PK |
| RoleName | Varchar(20) | NOT NULL |

**Possible roles:** Admin, Partner, Customer

## LoginHistory

|  |  |  |
| --- | --- | --- |
| ID | Long | PK |
| UserId | Integer | NOT NULL |
| IPAddress | Varchar(250) | NOT NULL |
| LoginDate | Datetime | NOT NULL |
| LoginStatus | Varchar(50) | NOT NULL |

**Note:** Each login attempt through login API should be logged in this LoginHistory table.

## PartnerDetails

|  |  |  |
| --- | --- | --- |
| ID | Integer | PK |
| UserId | Integer | FK NOT NULL |
| PartnerName | Varchar (500) | NOT NULL |
| PartnerLogoPath | Varchar (500) | NOT NULL – Some default image path |
| DateCreated | Datetime | NOT NULL |
| DateExpiry | Datetime | NULL |

## PartnerLocationDetails

|  |  |  |
| --- | --- | --- |
| ID | Integer | PK |
| PartnerId | Integer | FK NOT NULL |
| AddressLine1 | Varchar (500) | NOT NULL |
| AddressLine2 | Varchar (500) | NOT NULL |
| City | Varchar (100) | NOT NULL |
| State | Varchar (100) | NOT NULL |
| Country | Varchar (100) | NOT NULL |
| ZipCode | Varchar (100) | NOT NULL |
| Latitude | Float | NOT NULL |
| Longitude | Float | NOT NULL |
| MobileNumber | Varchar (20) | NOT NULL |
| DateCreated | Datetime | NOT NULL |
| DateActivated | Datetime | NULL |
| DateExpiry | Datetime | NULL |

## PartnerLocationWorkTime

|  |  |  |
| --- | --- | --- |
| ID | Integer | PK |
| PartnerId | Integer | FK NOT NULL |
| LocationId | Integer | FK NOT NULL |
| WorkDay | Varchar (500) | NOT NULL |
| WorkTime | Varchar (100) | NOT NULL |

## PartnerCharges

|  |  |  |
| --- | --- | --- |
| Id | Integer | PK |
| PartnerId | Integer | FK NOT NULL |
| Frequency | Varchar (100) | NOT NULL |
| Category | Varchar (100) | NOT NULL |
| Charges | Float | NOT NULL |
| IsDefault | Bool | NOT NULL DEFAULT TRUE |

**Frequency** = Monthly, Quarterly, Annually

**Category** = 1st Location, Subsequent Location, Offers, Featured Listing

When new partner is added, 6 records should get created in PartnerRates table with default values from the SysConfig table.

## PartnerRates

|  |  |  |
| --- | --- | --- |
| Id | Integer | PK |
| PartnerId | Integer | FK NOT NULL |
| FromRate | Varchar (100) | NOT NULL |
| ToRate | Varchar (100) | NOT NULL |
| Guaranteed | Float | NOT NULL |
| Indicative | Float | NOT NULL |
| DateCreated | Datetime | NOT NULL |
| LastModified | Datetime | NOT NULL |
| Status | Varchar (20) | NOT NULL DEFAULT ACTIVE |

## PartnerOffers

|  |  |  |
| --- | --- | --- |
| Id | Integer | PK |
| PartnerId | Integer | FK NOT NULL |
| OfferName | Varchar (50) | NOT NULL |
| OfferType | Varchar (5) | NOT NULL  Text | Image |
| OfferText | Varchar (200) | NOT NULL |
| OfferStartDate | Datetime | NOT NULL |
| OfferEndDate | Datetime | NOT NULL |
| Status | Varchar (20) | NOT NULL DEFAULT ACTIVE |
| DateCreated | Datetime | NOT NULL |
| LastModfied | Datetime | NOT NULL |
| DateActivation | Datetime | NOT NULL |
| DateExpiry | Datetime | NOT NULL |

## SystemParam

|  |  |  |
| --- | --- | --- |
| Id | Integer | PK |
| PartnerId | Integer | FK NOT NULL |
| Key | Varchar (100) | NOT NULL |
| DisplayValue | Varchar (100) | NOT NULL |
| Value | Varchar (100) | NOT NULL |
| ParentId | Integer | NULL DEFAULT -1 |
| DateCreated | Datetime | NOT NULL |
| Status | Varchar (20) | NOT NULL DEFAULT ACTIVE |

Populate following values as a part of initial setup:

* Default charges (12 values – Monthly, 1st Location, Subsequent Location, Offers)
* Supported currencies (AED, INR, USD)
* All configurations for Admin & Partners shall be stored in SystemParam table.
* Images shall not be saved in SystemParam table instead its relative path should be saved in table.

## UserNotifications

|  |  |  |
| --- | --- | --- |
| Id | Integer | PK |
| UserId | Integer | FK NOT NULL |
| NotificationText | Varchar (200) | NOT NULL |
| NotificationType | Varchar (100) | NOT NULL  (Expiry – Membership, Location, Offer) |
| IsRead | Bool | NOT NULL DEFAULT FALSE |
| DateCreated | Datetime | NOT NULL |
| DateRead | Datetime | NULL |
| Status | Varchar (20) | NOT NULL DEFAULT ACTIVE |

## UserReminders

|  |  |  |
| --- | --- | --- |
| Id | Integer | PK |
| UserId | Integer | FK NOT NULL |
| RateCondition | Varchar (20) | NOT NULL  (Equals|Greater Than Or Equal|Less Than Or Equal) |
| Rate | Float | NOT NULL |
| Distance | Float | NOT NULL DEFAULT 10 |
| DateCreated | Datetime | NOT NULL |
| Status | Varchar (20) | NOT NULL DEFAULT ACTIVE |

# APIs

1. All Dates should be stored in GMT
2. Generate Auth Token, all subsequent APIs should have this token in header.

|  |  |  |
| --- | --- | --- |
| Serial # | API | Rules |
| 1 | Login |  |
| 2 | Get partners |  |
| 3 | Add new partner |  |
| 4 | Get Partner Locations |  |
| 5 | Add Partner Location |  |
| 5.1 | Bulk Insert Partner Locations | * This will internally call Add Partner Location logic. * API will accept excel file & loop through the records to insert into DB * For each record, call Google GEO code API to get Latitude & Longitude before inserting the records. |
| 6 | Get Partner Offers |  |
| 7 | Add Partner Offer |  |
| 8 | Get Partner Charges |  |
| 9 | Get Default Charges |  |
| 10 | Save Partner Charges |  |
| 11 | Get Global Configurations |  |
| 12 | Save Global Configurations |  |
| 13 | Get Partner Settings |  |
| 14 | Save Partner Settings |  |
| 15 | Get Partner Rates |  |
| 16 | Save Partner Rates |  |
| 17 | Forgot Password |  |
| 18 | Change Password |  |
| 19 | Search Service Providers |  |
| 20 | Set Reminders |  |
| 21 | Generate Notifications | Check if any of the following is expiring soon (30 days) OR already Expired –   * Membership * Locations * Offers   Insert Records in notifications table for all such instances. |
| 22 | Get Notifications | Send across all unread notifications based on userid |
| 23 | Send Reminder Alert | Send across email alerts to subscribed users |

## Login

|  |  |
| --- | --- |
| URL | /api/login |
| Method | Post |
| Body | { “username” : “testuser”, “password” : “” } |
| Response | { “result”: “success”, “brand” : { “name”: “name of brand”, “logo” : “relative logo path”, “token”: “some token” }} |
| Error Message | * Invalid credentials, Your account is not active * Unable to login at this moment |

## Get Partners

|  |  |
| --- | --- |
| URL | /api/getPartners |
| Method | GET |
| Header | AuthToken |
| Body | { “status” : “all|active|inactive”, “name” : “starts with%”, “page”: “page #”, “pagesize” : “50”} |
| Response | [{ “name”: “UAE Exchange”, “locations”: “1023”, “status”: “active”, “dateCreated” : “5 Jan 2019”}] |
| Error Message | * Invalid token * No records found |

## Add Partner

|  |  |
| --- | --- |
| URL | /api/addPartner |
| Method | POST |
| Header | AuthToken |
| Body | { “name” : “partner name”, “username” : “new partner’s user name”, “email address”: “email address of partner” }  **Note**: Validate that user name or email already exists. If yes then throw error. |
| Response | {“result”: “success”, “partnerId” : “123”} |
| Error Message | * Invalid token * Unable to add new partner * Username/email address already exists |

## Get Partner Locations

|  |  |
| --- | --- |
| URL | /api/getPartnerLocations?partnerId=123 |
| Method | GET |
| Header | AuthToken |
| Body | { “status” : “all|active|inactive”, “zipcode” : [“zip code list”], “city”: “start with %”,”page #”: “1”, “pagesize” : “50”}  **Note:** By default return all the locations. |
| Response | [{ “address”: {“addressLine1”: “”, “addressLine2”: “”, “city”: “”, “state”: “”, “country”: “”,”zipcode”: “”,”latitude” :””, “longitude”:””}, “mobileNumber”: “1234567890”, “workTime”:[{“day”: “Monday”, “time”: “9 AM to 9 PM”}], “status“: “Active”, “dateActivated” : “5 Jan 2019”,“dateExpired” : “5 Feb 2019”}] |
| Error Message | * Invalid token * No locations found for given search criteria |

## Add Partner Location

|  |  |
| --- | --- |
| URL | /api/addPartnerLocation |
| Method | POST |
| Header | AuthToken |
| Body | [{ “partnered”: “123”, “address”: {“addressLine1”: “”, “addressLine2”: “”, “city”: “”, “state”: “”, “country”: “”,”zipcode”: “”,”latitude” :””, “longitude”:””}, “mobileNumber”: “1234567890”, “workTime”:[{“day”: “Monday”, “time”: “9 AM to 9 PM”}], “status“: “Inactive”, “dateActivated”: “”, “dateExpiry”: “” }] |
| Response | {“result”: “success”, “locationId” : “123”} |
| Error Message | * Invalid token * Unable to add location |

**Note:** Check Location expiry date, if the location expiry date is exceeding partner expiry date then update the location expiry date to partner’s date of expiry.

## Get Partner Offers

|  |  |
| --- | --- |
| URL | /api/getPartnerOffers?partnerId=123 |
| Method | GET |
| Header | AuthToken |
| Body | { “status” : “all|active|inactive”,”offerName: “starts with %”,”page #”: “1”, “pagesize” : “50”} |
| Response | [{“offerName”: “”, “startDate” : “”, “endDate”: “”, “offerText”: “”, “offerImagePath”:”relative path”}] |
| Error Message | * Invalid token * Unable to get offer list |

## Add Partner Offer

|  |  |
| --- | --- |
| URL | /api/addPartnerOffer |
| Method | POST |
| Header | AuthToken |
| Body | {“partnered”: “123”, “offerName”: “”, “startDate” : “”, “endDate”: “”, “offerText”: “”, “offerImage”:”base64 Image” }  **Note:** Store the image in physical location & save relative path in DB. |
| Response | {“result”: “success”, “offerId” : “123”} |
| Error Message | * Invalid token * Unable to add offer |

**Note**: When new offer is inserted, check the end date. If End date is exceeding the Partner’s Membership expiry date then update the endDate to partner expiry date.

## Get Partner Charges

|  |  |
| --- | --- |
| URL | /api/getPartnerCharges?partnerId=123 |
| Method | GET |
| Header | AuthToken |
| Body |  |
| Response | [{  "id": "",  "frequency": "Monthly|Quaterly|Annually",  "category": "1st Location, Subsequent Location, Offers, Featured Listing ",  "dateActivation": "",  "dateExpiry": "",  "charges": "",  "isDefault": "",  }] |
| Error Message | * Invalid token * Unable to get charges |

## Get Default Charges for all partners

|  |  |
| --- | --- |
| URL | /api/getDefaultCharges |
| Method | GET |
| Header | AuthToken |
| Body |  |
| Response | [{  "frequency": "Monthly|Quaterly|Annually",  "category": "1st Location, Subsequent Location, Offers, Featured Listing ",  "charges": ""}] |
| Error Message | * Invalid token * Unable to get charges |

## Save Partner Charges

|  |  |
| --- | --- |
| URL | /api/savePartnerCharges |
| Method | POST |
| Header | AuthToken |
| Body | [{“partnered: “123”, “chargeType”: “auto|custom”,  "frequency": "Monthly|Quaterly|Annually",  "category": "1st Location, Subsequent Location, Offers, Featured Listing ",  "charges": ""}] |
| Response | {“result”: “success”} |
| Error Message | * Invalid token * Unable to save partner charges |

## Get Global Configurations

|  |  |
| --- | --- |
| URL | /api/getGlobalConfig |
| Method | GET |
| Header | AuthToken |
| Body |  |
| Response | {"offers": "3",  "redirect": "true|false",  “charges” [{  "frequency": "Monthly|Quaterly|Annually",  "category": "1st Location, Subsequent Location, Offers, Featured Listing ",  "charges": ""}]  } |
| Error Message | * Invalid token * Unable to get configurations |

## Save Global Configurations

|  |  |
| --- | --- |
| URL | /api/saveGlobalConfig |
| Method | POST |
| Header | AuthToken |
| Body | {"offers": "3",  "redirect": "true|false",  “charges” [{  "frequency": "Monthly|Quaterly|Annually",  "category": "1st Location, Subsequent Location, Offers, Featured Listing ",  "charges": ""}]  } |
| Response | {“result”: “success”} |
| Error Message | * Invalid token * Unable to save configurations |

## Get Partner Settings

|  |  |
| --- | --- |
| URL | /api/getPartnerSettings?partnerId=123 |
| Method | GET |
| Header | AuthToken |
| Body |  |
| Response | {"brandName": "UAE Exchange",  "brandLogo": "relative image path",  "redirectLink": "https://google.com",  "emailAddress": "info@partner.com"} |
| Error Message | * Invalid token * Unable to get partner settings |

## Save Partner Settings

|  |  |
| --- | --- |
| URL | /api/savePartnerGeneralSettings |
| Method | POST |
| Header | AuthToken |
| Body | [{"key": "brandName",  "value": "UAE Exchange"},{  "key": "brandLogo","value": "relative image path"  },{"key": "redirectLink","value": "https://google.com"},{"key": "emailAddress",  "value": "info@partner.com"}] |
| Response | {“result”: “success”} |
| Error Message | * Invalid token * Unable to save settings |

## Get Partner Rates

|  |  |
| --- | --- |
| URL | /api/getPartnerRates?partnerId=123 |
| Method | GET |
| Header | AuthToken |
| Body |  |
| Response | [{  "id": ""  "from": "AED",  "to": "INR",  "guaranteed": "",  "indicative": ""  "status": ""  }] |
| Error Message | * Invalid token * API called from unauthorized source (other than domain name) |

## Save Partner Rates

|  |  |
| --- | --- |
| URL | /api/savePartnerRates |
| Method | POST |
| Header | AuthToken |
| Body | [{  "id": "If this is new record, ID will be empty",  “partnered”: “”,  "from": "AED",  "to": "INR",  "guaranteed": "",  "indicative": ""  "status": ""  },  "id": "If this is new record, ID will be empty",  “partnered”: “”,  "from": "AED",  "to": "INR",  "guaranteed": "",  "indicative": ""  "status": ""  ] |
| Response | {“result”: “success” } |
| Error Message | * Invalid token * API called from unauthorized source (other than domain name) |

## Forgot Password

|  |  |
| --- | --- |
| URL | /api/forgotPassword |
| Method | POST |
| Header |  |
| Body | { “username”: “”}  **Note:** This API should send email notification with change password link. |
| Response | {“result”: “success” } |
| Error Message | * API called from unauthorized source (other than domain name) |

## Change Password

|  |  |
| --- | --- |
| URL | /api/changePassword |
| Method | POST |
| Header |  |
| Body | { “userId“:”123”,”username”: “”, “oldPassword”:”123”, “newPassword”:”234” } |
| Response | {“result”: “success”} |
| Error Message | * API called from unauthorized source (other than domain name) |

## Search Service Provider

|  |  |
| --- | --- |
| URL | /api/search |
| Method | POST |
| Header |  |
| Body | {  “userId” : “”,  “latitude”: “”, “longitude”:””, “order”:”distance asc|rate desc|both”,”pageSize”:”20”,”page”:”1”,  “serviceType”: “remit”,  “serviceParams”: {“from”: “AED”, “to”: “INR”, “amount”: “1000”}  } |
| Response | [{  "brandName": "UAE Exchange",  "brandLogoPath": "",  "address":  {  "addressLine1": "",  "addressLine2": "",  "city": "",  "state": "",  "country": "",  "zipcode": "",  "latitude": "",  "longitude": ""  },  "workTime":[{"day": "Monday", "time": "9 AM to 9 PM"},{"day": "Tuesday", "time": "9 AM to 9 PM"}]  "distanceFromYourLocation": "",  "offer": {  "offerName": "",  "offerText": "",  "offerImagePath": "relative path"  },  "service": {  "amount": "1000",  "rate": "19.19",  "result": "19019"  }  }] |
| Error Message | * API called from unauthorized source (other than domain name) |

### Logic

1. You need to check offer if it is falling in today’s date range – Current date <= Date of expiry
2. You need to calculate the nearby locations based on GEO codes in input arguments. Also you need to calculate the distance while returning the response.
3. Sorting should be applied on two params – Nearest location / Highest rate
4. Offers should be attached to only top N results. N is configured and saved in SystemConfig table.
5. If UserID is missing in the request body then response should have offers as null.
6. If request body has valid userid then response should have offers based on search results.