|  |  |
| --- | --- |
| **Ticket ID** | TT4619 |
| **Ticket description** | Generate promo code against each category through CS App |
| **Created by** | Manali Bhadirage |
| **Created on** | 29/08/2022 |
| **Priority** | High |
| **Version** | 1.0 |

**Version**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Version no** | **Version Date** | **User name** | **User department** |
| 1 | 1.0 | 29/08/2022 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Approvals**

|  |  |  |
| --- | --- | --- |
| **Field** | **Name of the User** | **Approved date by the user** |
| **Actual User Name** |  |  |
| **Actual User Department** |  |  |
| **Organization Name** | CSJewellers |  |
| **Assigned BA** | Manali Bhadirage |  |
| **Assigned Developer** | Nikhil Bhosale |  |
| **Assigned Tester** | Priyanka Dupargude |  |

**Estimation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Department name** | **Estimated Time (In hr)** | **Scheduled Date (Starting date )** | **Estimated date** | **Actual delivery date** |
| **BA** |  |  |  |  |
| **Development** |  |  |  |  |
| **Testing** |  |  |  |  |
|  |  |  |  |  |

**Introduction**

Users offer discount on bills to customers. This discount should be enabled through promo code. On My CS app, we need to create option where user will click on Category (Types) and get the generated promo code. Further this promo code will be used by users for billing purpose.

Also, we need to create reports in back-end about promo code and its details.

**Business requirement**

Enter details of the ticket requirement.

**Abbreviations & terms**

Enter all the abbreviations used in the document.

**Existing system**

There is no such provision in existing application.

**Proposed system**

Step 1. We need to create Category (type) Master in back-end system (Connect Us).

Category defines the type of discount that users wish to offer to customers.

Category Master will have following fields:

* Add Data (Button)
* Action
* Category Name
* Assigned to Role
* Remark
* Is Active

|  |  |
| --- | --- |
| **FIELDS** | **DESCRIPTION** |
| Add Data | User can add new category by clicking on ‘Add Data’ button. Once clicked, following input fields should be displayed:   * Category Name * Assigned to Role * Remark * Is Active * Submit (button)   Category Name field consists of input box where user will enter the name of the category. It should accept both capital and small characters. It should accept space and maximum size of 50 characters.  (For example: Festive Discount, Celebration Offer, Special Customer Offer, etc.  Please make a not that the names of the categories will be defined by the end user later.)  Assigned to Role field should be mapped with Connect Us-> Master -> Role master. This field specifies which role should have provision to which category.  Example: Sales manager role can be mapped to all the categories. Sales person can be mapped to have provision for Festive Discount and Celebration Offer category only.  Remark field consists of input box which enables user to enter additional comments if any. It should accept both capital and small characters, space and all special characters. Maximum size of 100 characters should be accepted. Remark column should not be mandatory.  Is Active field consists of two icons namely ‘Yes’ and ‘No’.  If user clicks ‘Yes’, then this particular category will be displayed as active and this category will be displayed to user on App for code generation.  If user selects ‘No’, then this category won’t be displayed to user on App and subsequently will not be able to generate code under this category.  Once user fills the above details and clicks on submit, on successful addition, pop-up message should be displayed above as “Category added Successfully”. All the successfully added data should be displayed in grid (table on main page of this master). |
| Action | This field consists of View data, Edit data and Delete data actions.  View data should display the details of added category in vertical format.  Edit data should enable user to edit category name, from date, to date, remark and is active field and must have submit button at bottom. If user edits any of the input field and clicks on submit, then the edited data of category should be displayed on grid.  Delete data once clicked should delete the category and its details from the grid. |
| Assigned to Role | The list of roles to which the particular category is assigned should be displayed here. |
| Remark | Any remark if added by user while adding category, it should be displayed here. |
| Is Active | Is Active column should display either Yes or No status of category. |

Step 2. We need to set limit for promo code for each role.

Example: Sales manager may have limit to generate 50 codes each month under his login. Or, Sales person may have limit to generate 30 codes under his login for the month.

For code limit, we need to create Promo Code Limit Master. It should consist of following fields:

* Add Data
* Action
* Role
* Monthly Promo Code Limit
* Is active

|  |  |
| --- | --- |
| **FIELDS** | **DESCRIPTION** |
| Add Data | Through Add data button, user should be able to set promo limit for each role.  It should consists of following fields:   * Role * Monthly Promo Code Limit * Is Active * Submit button   In role field, user should be able to select role from list of roles from role master. List of Role should be mapped here from master -> role master.  User should be able to manually enter the digits in Monthly Promo Code Limit input box. It shouldn’t accept characters / special characters / decimals / space.  Is Active field consists of two icons namely ‘Yes’ and ‘No’.  If user clicks ‘Yes’, then this particular category will be displayed as active and this category will be displayed to user on App for code generation.  If user selects ‘No’, then this category won’t be displayed to user on App and subsequently will not be able to generate code under this category.  Once user fills the above details and clicks on submit, on successful addition, pop-up message should be displayed above as “Promo Code limit set successfully”. All the successfully added data should be displayed in grid (table on main page of this master). |
| Action | This field consists of View data, Edit data and Delete data actions.  View data should display the details of role and set promo code limit in vertical format.  Edit data should enable user to edit role, monthly promo code limit, is active field and must have submit button at bottom. If user edits any of the input field and clicks on submit, then the edited data should be displayed on grid.  Delete data once clicked should delete the added data and its details from the grid. |
| Role | Role selected by user while adding data and setting monthly promo code limit should be displayed here. |
| Monthly Promo Code Limit | Monthly promo code limit set by user against each user should be displayed here. |
| Is Active | Is Active column should display either Yes or No status of role in grid. |

Step 3. Once users signs in the MY CS application, on the home page we need to create one menu namely “Generate Code”. Once user taps on Generate menu, the list of categories assigned to that user should be displayed. Each category should have “Generate Code” button. Once user clicks on generate code button against any of the category, following details should be displayed on new page:

* Code Number
* Date & Time
* Remaining Promo Code limit

|  |  |
| --- | --- |
| **FIELDS** | **DESCRIPTION** |
| Promo Code Number | Once user clicks on generate code button from any of the category name, each time 6 digit unique promo code should be generated and displayed here.  Promo Code Number should consists of capital characters, small characters and digits.  Each role will have monthly limit for promo code generation.  When user has used all his code generation limit for the current month, and still if user clicks on “Generate code” button then it should display message as “Your code limit is exhausted for this month.” |
| Category | Name of the category which user has selected to generate the code should be displayed here. |
| Date & Time | Date and time on which the code is generated should be displayed.  Date should be in DD/MM/YYYY format.  Time should be in HH:MM:SS format. |
| Remaining Promo Code limit | Each role will have monthly limit for promo code generation.  Each time when user generates code, the count of his limit should be deducted from monthly code limit and remaining count of promo code should be displayed here.  Example: Suppose sales manager has monthly limit of 50 code generation.  When for first time, if he generates code under ‘festive discount category’, then his remaining code limit should be displayed as 49 codes.  Also, as code limit will be set for each month against each role then each month this code limit should be auto-renewed. |

Step 4. Moreover, at back-end user should have provision to extract report for the details of codes. Hence, create menu in connect-us namely “Promo Code Report”. It should consists of following input fields:

* From Date - Mandatory fields
* To Date - Mandatory fields
* Role - Not mandatory field
* Filter button
* Reset button
* Export button

|  |  |
| --- | --- |
| **FIELDS** | **DESCRIPTION** |
| Filter | The Filter allows user to filter & get desired data. |
| Reset | The Reset allows user to reset all the input fields. |
| Export | The Export action allows user to extract the report in CSV format. |

User should be able to extract report by any of the input fields. From date, Role, User name are not mandatory.

However, please ensure that if user selects “From date” then in this case selection of “to date” should be mandatory.

Once user enters input fields, following columns should be displayed in grid stating details of each promo code:

* Sr. No.
* Date
* Time
* Promo Code Number
* Category
* Role
* User - Username of the person who generated the code. This user name should be displayed as per the login of each user account.
* Total - Subtotal of promo codes should be displayed at the end of the desired report.

**References of the users**

|  |  |  |  |
| --- | --- | --- | --- |
| **User** | **Name** | **Mail** | **Contact number** |
| **Actual user** |  |  |  |
| **Ticket created by (if any)** | Harshali Rananaware | Harshali.rananaware@techneai.com | 2964 |
| **Assigned business analyst** | Manali Bhadirage | Manali.bhadirage@techneai.com | 2964 |
| **Assigned developer** | Nikhil Bhosale | Nikhil.bhosale@techneai.com | 2965 |
| **Assigned tester** | Priyanka Dupargude | Priyanka.dupargude@techneai.com | 2964 |