

MENU

Documentation Technique

- 1 - Écran de connexion
- 2 - Menu Principal
- 3 - Recherche un adhérent
- 4 - Création d'un adhérent
- 5 - Menu Adhérent
- 6 - Modification Client
- 7 - Archivage
- 8 - Contrat
- 9 - Choix du contrat (affectation)
- 10 - Choix contrat spécifique
- 11 - Lecture Contrat
- 12 - Modification Contrat

1 - Ecran de connexion

1.1 - Description de la fonctionnalité :

L'écran de connexion et le premier écran qui sera visible après le lancement de l'application. L'utilisateur doit entrer l'identifiant et son mot de passe qui lui ont été fournis afin de se connecter.

1.2 - Organisation et description des fichiers :

Dans le dossier '*CUSTOMER-MGMT/USER/SIGN-IN*' se trouve les fichiers qui compose la fonctionnalité de connexion de l'utilisateur, nous y retrouvons les fichiers :

'siback.cbl' : C'est le programme principal pour la connexion qui gère toute la logique avec la validation de saisie pour l'identifiant et le mot de passe.

'sifront.cbl' : Ce programme affiche l'écran de connexion d'un utilisateur jusqu'à ce que la connexion soit établie et appelle le sous-programme 'siback'.

Le fichier '*screen-sign-in.cpy*' qui contient la définition de l'écran de saisie pour la connexion de l'utilisateur se trouve lui dans le dossier COPYBOOK.

Il contient la screen-section qui servira d'interface pour permettre à l'utilisateur d'entrer son identifiant et mot de passe avant de valider.

1.3 - L'interface utilisateur.

L'écran est structuré avec des lignes vides et des espaces pour la mise en page différents champs de saisie sont alignés et identifiés avec des étiquettes.

La screen-section comprend :

- Un écran de connexion : '*SCREEN-SIGN-IN*'.
- Des Zones de saisie : '*SI-USER-ID*' pour l'identifiant.
'SI-USER-PWD' pour le mot de passe.
- Une zone pour le message d'erreur : '*SI-ERROR-MESSAGE*'.
- Des champs : '*Se connecter*' pour valider la saisie.
'SI-CONNECT' champ de contrôle

1.4 - Logique d'authentification de l'utilisateur :

Ce programme gère la validation des identifiants et des mots de passe saisis par l'utilisateur.

Il se connecte à une base de données pour vérifier les informations et retourne un indicateur de succès ou d'échec.

Il gère les erreurs SQL en fonction des valeurs des code retour SQL.

C'est également grâce à une requête sql qu'il récupère les informations de l'utilisateur si l'identifiant et le mot de passe sont corrects.

DECLARE CURSOR CURSOR FOR : Déclare un curseur pour sélectionner l'identifiant et le mot de passe de l'utilisateur à partir de la table user_tab.

OPEN CURSOR : Ouvre le curseur.

FETCH CURSOR INTO : Récupère les résultats de la requête dans les variables SQL-USER-ID et SQL-USER-PWD.

IF SQLCODE EQUAL ZERO: Si la requête réussit, assigne 'TRUE' à LK-PWD-CHECK.

CLOSE CUSOR : Ferme le curseur

1.5 - Instructions de tests unitaires et mesures de sécurité :

- Test de Connexion Réussi :
 - a. Vérifier que le programme se connecte correctement à la base de données avec les informations d'identification correctes.
 - b. Exécuter le programme avec les informations d'identification valides.
 - c. La connexion réussit sans erreur (SQLCODE = 0), le programme continue d'exécuter les étapes suivantes.
- Test de Connexion Échouée :
 - a. Vérifie que le programme gère correctement une tentative de connexion échouée en raison d'identifiants incorrects.
 - b. Modifier les informations (*USER_NAME* ou *PASSWRD*) pour qu'elles soient incorrectes et exécuter le programme.
 - c. La connexion doit échouer avec un code retour spécifique (par exemple, -01).
 - d. Le programme affiche le message d'erreur approprié.
- Test de Connexion Échouée (Base de Données Indisponible).
 - a. Vérifie que le programme gère correctement une tentative de connexion échouée en raison d'une base de données indisponible.
 - b. La base de données doit être indisponible, arrêter le service de la base de données et exécuter le programme
 - c. La connexion doit échouer avec un SQLCODE spécifique (par exemple, -30).

Par mesure de sécurité et garantir que les opérations de sélection de menu sont sécurisées, toutes les entrées utilisateur sont valides pour éviter les saisies incorrectes et les attaques par injection.

Utilisation de *FUNCTION UPPER-CASE* pour s'assurer que les entrées correspondent aux valeurs attendues ('O' pour les options valides).

2 - Menu Principal

2.1 - Description de la fonctionnalité :

Le menu permet à l'utilisateur de naviguer dans les différentes pages de l'application afin d'effectuer les opérations souhaitées (Créer/Rechercher adhérent).

2.2 - Organisation et description des fichiers :

Dans le dossier '*CUSTOMER-MGMT/CUSTOMER/MANAGEMENT-CUSTOMER*' il y a le fichier '*manacust.cbl*'.

Ce fichier contient la logique du menu permettant à l'utilisateur de choisir entre les options rechercher et créer un adhérent.

Dans le dossier '*COPYBOOK*' nous retrouvons le fichier '*screen-management-customer.cpy*' qui contient la screen-section qui servira d'interface pour permettre à l'utilisateur d'entrer son choix.

2.3 - L'interface utilisateur :

L'écran est structuré avec des lignes vides et des espaces pour la mise en page différents champs de saisie sont alignés et identifiés avec des étiquettes.

Il y a 3 champs 1 pour la recherche *WS-SEARCH-CHOICE* un pour la création *WS-CREATE-CHOICE* et enfin le champ de retour au menu *LK-RETURN-CHOICE*.

2.4 - Logique de filtrage des options en fonction du profil utilisateur :

L'utilisateur aura le choix entre trois options :

- Le retour au menu, dans ce cas revient à l'écran précédent.
- La création dans ce cas, passe à l'écran de création d'adhérent.
- La recherche dans ce cas, passe à l'écran de recherche d'adhérent.

Le programme vérifie les entrées utilisat :ur jusqu'à ce qu'une option valide soit sélectionnée.

2.5 - Instructions de tests unitaires :

- a. Tests qui doivent fonctionner :

- L'utilisateur a bien accès au menu via le screen section qui s'affiche correctement avec toutes les options possibles affiché et sélectionnable
 - L'utilisateur peut sélectionner l'une des trois options proposé
 - Une fois la validation effectuer l'application passe à l'écran souhaitez
- b. Tests qui doivent échouer :
- L'utilisateur entre plusieurs choix à la fois

2.6 - Mesures de sécurité :

Par mesure de sécurité et garantir que les opérations de sélection de menu sont sécurisées, toutes les entrées utilisateur sont valides pour éviter les saisies incorrectes et les attaques par injection.

Utilisation de *FUNCTION UPPER-CASE* pour s'assurer que les entrées correspondent aux valeurs attendues ('O' pour les options valides).

3 - Recherche adhérent :

3.1 - Description de la fonctionnalité :

L'option recherche d'adhérent permet à l'utilisateur d'effectuer une recherche soit par numéro de sécurité sociale soit par nom prénom et date de naissance ou avec toutes ces informations à la fois.

3.2 - Organisation et description des fichiers :

La partie recherche adhérents est composée d'un programme principal searcust.cbl qui appelle plusieurs sous-programmes :

- 'scback.cbl' : Gère la logique de la recherche d'adhérent.
- 'scfront.cbl' : Gère la logique de la partie front de l'application

Il y a également le fichier :

- 'screen-frame.cpy' : Contient la screen-section qui servira d'interface pour naviguer dans la recherche des adhérents

C'est dans le fichier scback.cbl que s'effectue la recherche dans la table customer d'un ou plusieurs adhérents en fonction des saisies de l'utilisateur dans les différents champs de recherche.

3.3 - L'interface utilisateur :

Les champs de saisie pour le numéro de sécurité sociale, le prénom, le nom et la date de naissance sont définis pour permettre à l'utilisateur de saisir les informations de recherche.

L'utilisateur a le choix entre retourner au menu principal et l'option "rechercher". Après avoir rempli les différents champs, l'utilisateur doit saisir la lettre "O" pour valider sa sélection.

Si les champs requis ne sont pas correctement remplis ou sont incomplets, un message d'erreur sera affiché pour prévenir l'utilisateur. Le statut de la requête et les résultats sont affichés après la recherche.

3.4 - Logique de recherche et de traitement des résultats :

Quand l'utilisateur choisit l'option "Retour au menu", il sera redirigé vers la partie "Gestion d'adhérent". Si le résultat de la requête SQL est nul, l'utilisateur sera redirigé vers le début du programme avec un message adéquat.

Pour la recherche d'adhérent, selon le type de recherche (par numéro de sécurité sociale, parrnom, prénom et date de naissance, ou par numéro de sécurité sociale + nom + prénom + date de naissance), le programme doit se connecter à la base de données pour accéder aux informations de la base client. Si la connexion échoue, un message avertissant l'utilisateur sera affiché.

Une fois connecté à la base de données et que l'utilisateur a saisi les informations, si un résultat est trouvé, les requêtes SQL retournent l'adhérent ou les adhérents non archivés en fonction de la recherche effectuée par l'utilisateur :

- Effectue le fetch pour le curseur de *code_secu*.
- Effectue le fetch pour le curseur de *lastname, firstname* et *birth_date*.
- Effectue le fetch pour le curseur de *code_secu, lastname, firstname* et *birth_date*.

Le résultat de la requête SQL est ensuite stocké dans la table customer.

3.5 - Instructions de tests unitaires :

Tests qui doivent fonctionner :

- L'utilisateur a accès au menu via la screen section qui s'affiche correctement avec toutes les options possibles affichées et sélectionnables.
- L'utilisateur peut effectuer une recherche uniquement avec le numéro de sécurité sociale.
- L'utilisateur peut effectuer une recherche avec le nom, le prénom et la date de naissance.
- L'utilisateur peut effectuer une recherche avec le numéro de sécurité sociale, le prénom et la date de naissance.

Tests qui doivent échouer :

Recherche avec le numéro de sécurité sociale :

- Effectuer une recherche avec un numéro de sécurité sociale invalide.
- Effectuer une recherche avec le champ vide.
- Recherche avec nom, prénom et date de naissance :
- Effectuer une recherche avec tous les champs vides.
- Effectuer une recherche avec un des champs laissé vide, avec les autres champs correctement renseignés
- Effectuer une recherche avec des chiffres ou caractères spéciaux dans le champ nom, avec les autres champs correctement renseignés.
- Effectuer une recherche avec des chiffres ou caractères spéciaux dans le champ prénom, avec les autres champs correctement renseignés.

- Effectuer une recherche avec une date de naissance erronée, avec les autres champs correctement renseignés.

Recherche avec le numéro de sécurité sociale, le nom, le prénom et la date de naissance :

- Effectuer une recherche avec tous les champs vides.
- Effectuer une recherche avec un des champs laissé vide, avec les autres champs correctement renseignés.
- Effectuer une recherche avec des chiffres ou caractères spéciaux dans le champ nom, avec les autres champs correctement renseignés.
- Effectuer une recherche avec des chiffres ou caractères spéciaux dans le champ prénom, avec les autres champs correctement renseignés.
- Effectuer une recherche avec une date de naissance erronée, avec les autres champs correctement renseignés.
- Effectuer une recherche avec un numéro de sécurité sociale invalide, avec les autres champs correctement renseignés.

3.6 - Mesures de sécurité :

Par mesure de sécurité et garantir que les opérations de sélection de menu sont sécurisées, toutes les entrées utilisateur sont valides pour éviter les saisies incorrectes et les attaques par injection.

Utilisation de *FUNCTION UPPER-CASE* pour s'assurer que les entrées correspondent aux valeurs attendues ('O' pour les options valides).

4 - Crédit Adhérent

4.1 - Description de la fonctionnalité :

La fonctionnalité "Création Adhérent" permet de saisir et d'enregistrer un nouvel adhérent dans le système. Elle est composée de plusieurs éléments :

1. *creacust* : Programme principal qui orchestre le processus de création.
2. *ccfront* : Sous-programme gérant l'interface utilisateur pour la saisie des données.
3. *ccback* : Sous-programme gérant la logique de traitement et d'enregistrement des données dans la base de données.
4. *screen-add-customer.cpy* : COPY contenant la définition de l'écran de saisie.

4.2 - Programme Principal : *creacust*

4.2.1 - Description

Le programme principal *creacust* coordonne les appels aux sous-programmes *ccfront* et *ccback*, assurant la collecte, la validation et l'enregistrement des données de l'adhérent.

4.2.1 - Fonctionnalités

- Initialisation et appel à *ccfront* :
- Initialise les variables nécessaires pour la saisie.

- Appelle le sous-programme *ccfront* pour afficher le formulaire de saisie et recueillir les données de l'adhérent.
- Validation et boucle de retour :
- Si l'utilisateur choisit de retourner au menu (*WS-MENU-RETURN* égal à 'O'), le programme revient au début.
- Sinon, il appelle *ccback* pour valider et enregistrer les données.
- Appel à *ccback* :
- Si les données saisies sont valides (*WS-VALIDATION-STATUS* égal à 'Y'), le sous-programme *ccback* est appelé pour insérer les données dans la base de données.

4.2 - Interface Utilisateur : *ccfront*

4.2.1 Description

Le sous-programme *ccfront* gère l'affichage de l'interface utilisateur et la validation initiale des données saisies par l'utilisateur.

4.2.2 Fonctionnalités :

Affichage du Formulaire :

- Utilise la COPY *screen-add-customer.cpy* pour afficher les champs à saisir.
- Chaque champ (nom, prénom, adresse, etc.) est défini avec ses contraintes de longueur et de type.

Validation des Champs :

- Vérifie que tous les champs obligatoires sont remplis.
- Contrôle que les données respectent les formats et les longueurs spécifiés (par exemple, numéro de téléphone numérique, date de naissance valide).

Messages d'Erreur :

- Affiche des messages d'erreur spécifiques en cas de saisie incorrecte, par exemple "Le genre est obligatoire" ou "Le numéro de téléphone doit être numérique".

Navigation :

- Permet de retourner au menu principal ou de continuer la saisie en fonction des choix de l'utilisateur.

4.3 - Logique de Traitement : *ccback*

4.3.1 - Description :

Le sous-programme *ccback* traite les données saisies et les enregistre dans la base de données.

Fonctionnalités

Connexion à la Base de Données :

Utilise des identifiants pour se connecter à la base de données spécifiée.

Insertion des Données :

Insère les informations de l'adhérent dans la table *CUSTOMER* si la validation est réussie.

Déplace les données des variables de liaison (*LK-ADHERENT-INPUT*) vers les variables de stockage de travail (*WS-ADHERENT*).

Messages d'Erreur :

- Gère les erreurs et affiche des messages appropriés en cas d'échec de la validation ou de l'insertion.

La COPY *screen-add-customer.cpy* contient la définition des champs de l'écran de saisie pour l'adhérent. Fonctionnalités Champs de Saisie :

- Définit les différents champs de saisie pour les informations de l'adhérent (nom, prénom, adresse, etc.).
- Chaque champ est associé à une position spécifique sur l'écran (ligne et colonne).

Messages d'Erreur :

- Zone dédiée pour afficher les messages d'erreur en cas de validation échouée

4.4 - Instructions de tests unitaires mesures de sécurité :

- Vérification de la Saisie :

Tester chaque champ pour s'assurer qu'ils acceptent les données Correctement et afficher les messages d'erreur appropriés.

- Insertion en Base de Données :

Vérifier que les données sont insérées correctement dans la base de données.

Par mesure de sécurité toutes les entrées utilisateur sont à valider pour éviter les attaques par injection SQL

- Gestion des Erreurs :

Utiliser des messages d'erreur génériques pour ne pas révéler d'informations sensibles.

- Connexion Sécurisée :

Protéger les identifiants de connexion à la base de données et utiliser des connexions sécurisées

5 - Menu adhérent :

5.1 - Description de la fonctionnalité :

Le menu adhérent permet à l'utilisateur une fois qu'il est sur le profil de l'adhérent d'accéder aux options pour :

- Modifier l'adhérent
- (Archiver l'adhérent)
- Accéder aux contrats
- (Accéder aux remboursements)
- (Accéder aux rapports)
- Valider son choix ou revenir au menu principal
-

5.2 - Organisation et description des fichiers :

La fonctionnalité menu adhérent est composée de plusieurs fichiers :

- 'mcback.cbl' : Programme principal qui contient toute la logique du menu .

Ce programme recherche dans la table 'customer' un adhérent en fonction de l'*UUID* reçu par le programme appelant et le stock dans le *CUSTOMER* de la *LINKAGE SECTION* de l'utilisateur dans les différents champs de recherche.

- 'mcfront.cbl' :Sous-programme gérant l'affichage de l'écran de menu de l'adhérent.

Il y a également le fichier :

- 'screen-menu-customer.cpy' : COPY contenant la définition de l'écran de saisie et de sélection pour l'utilisateur.

5.3 - L'interface utilisateur.

Le programme 'mcfront.cbl' gère la logique d'affichage du menu tandis que le copy 'screen-menu-customer.cpy' contient la définition de l'écran .

Les champs de modification ,d'archivage, de contrats et de remboursements sont affichés et sélectionnables par l'utilisateur .

Un message d'erreur est affiché en cas de mauvaise saisie et enfin l'utilisateur peut valider son choix ou retourner aux menu principale

5.3 - Logique de traitement

Ce programme recherche dans la table 'customer' un adhérent en fonction de l'*UUID* reçu par le programme appelant et le stock dans le *CUSTOMER* de la *LINKAGE SECTION*.de l'utilisateur dans les différents champs de recherche

Les données sont transférées de *LK-UUID-RECEVED* vers *WS-UUID-RECEVED* via des requêtes sql qui retourne un adhérent en fonction du UUID reçu par le programme appelant .

Un *FETCH* est ensuite effectué pour le *CURSOR CRSUUID* ,
enfin le résultat de la requête SQL est stocké dans le *CUSTOMER* de la *LINKAGE SECTION*.

5.4 - Instructions de tests unitaires et Mesures de sécurités

- L'utilisateur a accès au menu et à toutes les options qui s'affiche
- L'utilisateur peut sélectionner l'option choisie et valider
- L'utilisateur ne peut pas choisir plusieurs options à la fois

Par mesure de sécurité et garantir que les opérations de sélection de menu sont sécurisées, toutes les entrées utilisateur sont valides pour éviter les saisies incorrectes et les attaques par injection.

Utilisation de *FUNCTION UPPER-CASE* pour s'assurer que les entrées correspondent aux valeurs attendues ('O' pour les options valides).

6 - Modification adhérent :

6.1 - Description de la fonctionnalité :

cette fonctionnalité comme son nom l'indique sert à modifier les informations d'un adhérent

6.2 - Organisation et description des fichiers :

La modification de l'adhérent est composé de plusieurs fichiers dans le dossier '*CUSTOMER-MGMT/CUSTOMER/UPDATE-CUSTOMER*' nous retrouvons les fichiers :

'*ucback.cbl*' :

Ce programme gère toute la logique pour la partie modification de client , et appelle
'*ucfront.cbl*' :

Ce programme affiche le formulaire de modification d'un adhérent et appelle le sous-programme
'*ucback*' qui effectue la modification dans la DataBase

'*screen-update-customer.cpy*' :

Dans le dossier *COPYBOOK* ce fichier contient la définition de l'écran de saisie et de sélection pour l'utilisateur.

6.3 - L'interface utilisateur :

Le sous programme '*ucfront.cbl*' gère la logique d'affichage du menu tandis que le copy '*screen-update-customer.cpy*' contient la définition de l'écran .

L'utilisateur peut modifier le nombre de champs qu'il souhaite et valider pour l'enregistrement.

En cas de champ mal renseigné un message d'erreur sera affiché invitant l'utilisateur à modifier sa saisie.

6.4 - Logique de traitement :

Ce programme effectue les modifications dans la DataBase en fonction des informations saisies par l'utilisateur dans le formulaire de modification.

Dans un premier temps il initialise les données pour la requête SQL pour ensuite mettre à jour les données de l'adhérent en fonction des modifications saisies par l'utilisateur dans le formulaire de modification.

MOVE WS-CUSTOMER TO LK-CUSTOMER : Copie les informations de travail dans les variables de liaison.

6.5 - Instructions de tests unitaires et mesures de sécurité :

- Test de Modification Réussie :

Vérifie que le programme met correctement à jour les informations d'un adhérent avec des données valides.

Les informations d'identification (*USER_NAME,PSW*) sont correctes.

- a. Un adhérent avec l'*UUID* fourni existe dans la base de données.
- b. Fournir des informations valides pour l'adhérent à modifier et exécuter le programme

Les informations de l'adhérent doivent être mises à jour dans la base de données et Le programme se termine sans erreur (*SQLCODE = 0*).

- Test de Modification Échouée (Adhérent Inexistant):
Vérifie que le programme gère correctement une tentative de modification échouée en raison d'un adhérent inexistant.
- c. Les informations d'identification (*USER_NAME, PSWRD*) sont correctes.
- e. Fournir un *UUID* inexistant pour l'adhérent à modifier et exécuter le programme
- f. Le programme affiche un message d'erreur approprié.

Par mesure de sécurité et garantir que les opérations de sélection de menu sont sécurisées, toutes les entrées utilisateur sont valides pour éviter les saisies incorrectes et les attaques par injection.

Utilisation de *FUNCTION UPPER-CASE* pour s'assurer que les entrées correspondent aux valeurs attendues ('O' pour les options valides).

7 - Archivage adhérent :

7.1 - Description de la fonctionnalité :

Le programme permet à l'utilisateur d'archiver les informations d'un adhérent.

L'archivage implique le déplacement des informations de l'adhérent de la table principale vers une table d'archivage et la suppression de ses données de la table principale.

Le programme gère également l'interaction avec l'utilisateur pour confirmer l'archivage et affiche des messages appropriés en fonction des actions effectuées.

7.2 - Organisation et description des fichiers :

La fonctionnalité archivage de l'adhérent est composée de plusieurs fichiers:

- '*acfront.cbl*' : Programme principal qui contient toute la logique de la lecture du contrat .
Le programme permet de sélectionner des taux de remboursement spécifiques pour l'adhérent afin de personnaliser son contrat .

- ‘screen-archive-customer.cpy’ : *COPY contenant la définition de l’écran de saisie / sélection pour l’utilisateur.*

Le chemin du fichier ‘acfrontt.cbl’ est ‘CUSTOMER-MGMT/CUSTOMER/ARCHIVE-CUSTOMER/acfront.cbl’.

Le chemin du fichier ‘screen-specific-contrat.cpy’ est ‘CUSTOMER-MGMT/COPYBOOK/screen-archive-customer.cpy’.

7.3 - L’interface utilisateur.

Après avoir sélectionné l’option ‘Archivage’ l’utilisateur arrive sur la page de validation de son choix il peut confirmer l’archivage de l’adhérent grâce au message de confirmation et qui rappelle le nom et prénom de l’adhérent .

Un champ de validation est présent ainsi que l’option Retour au menu en bas à droite de l’écran.

7.4 - Logique de traitement :

Le programme est conçu pour archiver l’adhérent sélectionné afin qu’il ne soit plus visible dans le fichier client.

Le programme vérifie l’option choisie par l’utilisateur et une fois la validation faite insère des informations du client dans la table d’archivage et Suppression des informations du client de la table principale.

Pour l’affichage de l’écran de gestion d’erreur si l’utilisateur entre autre chose que la lettre ‘o’ dans le champ de validation ou de retour au menu.

Enfin un message de validation confirme l’archivage de l’adhérent.

7.5 - Instructions de tests unitaires et mesures de sécurité :

-Vérifier la sélection de l’option choisie par l’utilisateur:

Tester l’option ‘Archivage’ en entrant la lettre ‘o’.

Tester l’option ‘Retour au menu’ en entrant la lettre ‘o’.

Tester l’option ‘Archivage’ en entrant autre chose que la lettre ‘o’.

Tester l’option ‘Retour au menu’ en entrant autre chose que la lettre ‘o’.

-Vérifier l’archivage est réussi en recherchant l’adhérent sur la page ‘Recherche Adhérent’.

Par mesure de sécurité et garantir que les opérations de sélection de menu sont sécurisées, toutes les entrées utilisateur sont valides pour éviter les saisies incorrectes et les attaques par injection.

Utilisation de *FUNCTION UPPER-CASE* pour s’assurer que les entrées correspondent aux valeurs attendues (‘O’ pour les options valides).

8 - Menu gestion de contrat :

8.1 - Description de la fonctionnalité :

Le menu de gestion de contrat proposé à l'utilisateur à l'utilisateur 3 options :

- Affectation : affecter un contrat à l'adhérent.
- Lecture : affiché le contrat auquel a souscrit l'adhérent.
- Modification : Modifier le contrat auquel a souscrit l'adhérent.

Le nom ,le prénom et le numéro de sécurité social de l'adhérent est également affichés pour l'utilisateur.

8.2 - Organisation et description des fichiers :

La fonctionnalité menu de gestion de contrat est composée de plusieurs fichiers :

- '*menucont.cbl*' : Programme principal qui contient toute la logique du menu-contrat .
Le programme permet de sélectionner l'une des trois options proposées.
- '*screen-menu-contrat.cpy*' : COPY contenant la définition de l'écran de saisie / sélection pour l'utilisateur.

8.3 - L'interface utilisateur :

Le programme '*menucont.cbl*' gère la logique d'affichage du menu tandis que le copy '*screen-menu-contrat.cpy*' contient la définition de l'écran .

L'utilisateur peut sélectionner l'option choisie en entrant la lettre 'o' et en appuyant sur la touche entrer , il peut également revenir au menu avec l'option qui est en bas à droite de l'écran .

Le nom,le prénom et le numéro de sécurité social de l'adhérent sont affichés en haut à gauche de l'écran.

Un message d'erreur est affiché en cas de mauvaise saisie.

8.4 - Logique de traitement :

Ce programme gère l'affichage et la gestion des options de menu pour les contrats. Il permet à l'utilisateur de sélectionner des options pour affecter, lire, modifier ou retourner au menu principal.

L'entrée utilisateur est vérifiée en bouclant avec le paragraphe *1000-SCREEN-LOOP-START*jusqu'à ce qu'une option valide soit sélectionnée.

Le paragraphe *1100-PREPARE-SCREEN-START* prépare les informations de l'adhérent qui vont être affichés

Le paragraphe *3000-WITCH-CHOICE-START* vérifie chaque choix de l'utilisateur et appelle le sous-programme approprié en fonction de l'option sélectionnée

Le paragraphe *9200-ERROR-MESSAGE-START* affichera un message d'erreur en cas d'erreur .

8.5 - Instructions de tests unitaires et mesures de sécurités :

Vérifier que la gestion des contrats via le menu fonctionne correctement et gérer les erreurs de saisie de manière appropriée.

Vérifie que l'option "Retour" fonctionne correctement

Selectionner l'option "Retour" en entrant 'O' et exécuter le programme.

Vérifie que l'option "Affectation" fonctionne correctement.

Selectionner l'option en entrant 'O' et exécuter le programme.

Vérifie que l'option "Lecture" fonctionne correctement.

Selectionner l'option en entrant 'O' et exécuter le programme.

Vérifie que le programme gère correctement les saisies incorrectes.

Entrer une valeur incorrecte pour n'importe quel choix (autre que 'O') et exécuter le programme.

Par mesure de sécurité et garantir que les opérations de sélection de menu sont sécurisées, toutes les entrées utilisateur sont valides pour éviter les saisies incorrectes et les attaques par injection.

Utilisation de *FUNCTION UPPER-CASE* pour s'assurer que les entrées correspondent aux valeurs attendues ('O' pour les options valides).

9 - Sélection du contrat classique :

9.1 - Description de la fonctionnalité :

La fonctionnalité gère la sélection du type de contrat par un adhérent. L'écran affiche les informations des trois contrats types disponibles et propose plusieurs actions possibles à l'utilisateur

Choisir un contrat : L'utilisateur peut sélectionner l'un des trois types de contrat disponibles.

Confirmer sans changement : L'utilisateur peut confirmer le contrat sélectionné sans apporter de modifications.

Choisir un contrat spécifique : L'utilisateur peut demander une personnalisation du contrat sélectionné.

Enfin, il est possible de revenir au menu principal des contrats.

9.2 - Organisation et description des fichiers :

La fonctionnalité sélection de contrat est composée de plusieurs fichiers:

- ‘*clascont.cbl*’ : Programme principal qui contient toute la logique de la sélection du contrat .
Le programme permet de sélectionner l’un des trois contrats proposés par la mutuelle , ou d’opter pour un contrat spécifique .

- ‘*screen-classique-contrat.cpy*’ : *COPY* contenant la définition de l’écran de saisie / sélection pour l’utilisateur.

Le chemin du fichier ‘*manacust.cbl*’ est ‘*CUSTOMER-MGMT/CONTRACT/MENU-CONTRACT*’.

Le chemin du fichier ‘*screen-classique-contrat.cpy*’ est ‘*CUSTOMER-MGMT/COPYBOOK/screen-menu-classique-contrat.cpy*’.

9.3 - L’interface utilisateur :

Le prénom, le nom et le numéro de sécurité sociale de l’adhérent sont affichés en haut à gauche de l’écran.

L’écran pour la sélection de contrat classique affiche les 3 types de contrat proposés par la mutuelle Allégé , Modéré et Excellence .

Pour les 3 types de contrats il ya les détails des taux de remboursements Soins courants , Hospitalisation, Optique et Dentaire.

L’utilisateur peut également opter pour un contrat spécifique dans ce cas la il entre son choix et sera redirigé vers la page de contrat spécifique.

Enfin, l’utilisateur peut revenir au menu principal en sélectionnant l’option.

9.4 - Logique de traitement :

Le programme permet à l’utilisateur d’assigner un contrat à un adhérent une fois que celui-ci a fait son choix ,en effet l’adhérent doit d’abord choisir le type de contrat sur lequel baser la personnalisation.

Cet écran affiche les informations des 3 contrats types disponibles et les actions possibles:

- choisir un contrat,
- confirmer sans changement,
- demander une personnalisation
- revenir au menu contrat.

Pour afficher les contrats, le programme se connecte à la base de données, prépare et ouvre le curseur SQL pour récupérer les informations des contrats classiques dans le paragraphe *1400-PREPARE-CRS-CLASSIC-SCREEN-START*.

Le paragraphe *1450-PREPARE-CRS-COUT-SCREEN-START* est pour la préparation et l’ouverture du curseur SQL pour récupérer les coûts des contrats.

9.5 - Instructions de tests unitaires et mesures de sécurité :

Vérifier que la sélection du contrat fonctionne correctement et gérer les erreurs de saisie de manière appropriée.

Test de Sélection :

- Vérifie que les options pour les contrats '*'ALLEGE'*' ,'*'MODERE'* et '*'EXCELLENCE'* fonctionnent correctement :
Sélectionner les options tour à tour en entrant 'O' et exécuter le programme pour chaque type de contrat.
- Vérifier que l'option "Retour" fonctionne correctement :
Sélectionner l'option "Retour" en entrant 'O' et exécuter le programme.
- Vérifie que l'option "Contrat spécifique" fonctionne correctement :
Sélectionner l'option "Contrat spécifique" en entrant 'O' et exécuter le programme.

Par mesure de sécurité et garantir que les opérations de sélection de menu sont sécurisées, toutes les entrées utilisateur sont valides pour éviter les saisies incorrectes et les attaques par injection.

Utilisation de *FUNCTION UPPER-CASE* pour s'assurer que les entrées correspondent aux valeurs attendues ('O' pour les options valides).

10 - Selection de contrat spécifique :

10.1 - Description de la fonctionnalité :

La fonctionnalité gère la sélection du type de contrat spécifique pour un adhérent qui en a fait la demande . L'écran affiche les informations des 3 contrats types disponibles et permet de faire un mixte des prestations.

L'utilisateur peut sélectionner le taux de remboursement souhaité par l'adhérent pour différents types de prestations comme les honoraires du médecin, les règlement paramédicaux , les hospitalisations, les verres pour lunettes simples ou progressifs, les couronnes dentaires et le détartrage.

10.2 - Organisation et description des fichiers :

La fonctionnalité sélection de contrat spécifique est composée de plusieurs fichiers:

- '*'speccont.cb'*' : Programme principal qui contient toute la logique de la sélection du contrat spécifique .
Le programme permet de sélectionner des taux de remboursement spécifiques pour l'adhérent afin de personnaliser son contrat .

- ‘screen-specific-contrat.cpy’ : COPY contenant la définition de l’écran de saisie / sélection pour l’utilisateur.

Le chemin du fichier ‘manacust.cbl’ est: ‘CUSTOMER-MGMT/CONTRACT/SPECIFIC-CONTRACT/speccont.cbl’.

Le chemin du fichier ‘screen-specific-contrat.cpy’ est:

‘CUSTOMER-MGMT/COPYBOOK/screen-specific-contract.copy’.

10.3 - L’interface utilisateur :

Le prénom, le nom et le numéro de sécurité sociale de l’adhérent sont affichés en haut à gauche de l’écran.

L’écran pour la sélection de contrat spécifique affiche également le type de prestations :

- Honoraires du médecin
- Les règlement paramédicaux
- Les hospitalisations
- Les verres simples
- Les verres progressifs,
- Les couronnes dentaires
- Le détartrage.

Pour chacune de ces prestations 3 choix sont disponibles pour le taux de remboursement .

Il y a un champ ‘Valider mes choix’ qui permet d'afficher le montant qui sera payé par l'adhérent et sera affiché dans le champ ‘Montant du contrat’ .

Enfin le champ de ‘Validation du contrat’ pour valider les choix et le champ ‘Retour au menu’.

10.4 - Logique de traitement :

Le programme permet à l’utilisateur d’assigner un contrat spécifique à un adhérent en tenant compte des spécificités.

Il affiche l’écran de personnalisation et attend les choix de l’utilisateur qui peut sélectionner des options de prestations et choisir de valider les sélections ou de revenir au menu précédent.

Les choix de l’utilisateur sont validés pour s’assurer qu'il n'y a pas de doublons et que chaque prestation à une sélection.

Le coût de chaque prestation choisie est additionné pour obtenir le coût total qui sera ajusté en fonction de l’âge de l’utilisateur et du nombre d’enfants à charge.

Si l’utilisateur valide le contrat, les informations du contrat sont insérées dans la table ‘CUSTOMER_REIMBURSEMENT’ de la base de données puis génère un numéro unique pour le contrat et enregistre toutes les prestations sélectionnées ainsi que leur coût.

Les erreurs de saisie sont gérées en affichant des messages appropriés et en demandant à l'utilisateur de corriger les erreurs.

Une fois que la validation du contrat est réussie, le programme retourne au menu principal des contrats

10.5 - Instructions de tests unitaires et mesures de sécurité :

- S'assurer que la connexion à la base de données est établie sans erreur.
- Vérifier que les informations des contrats sont correctes.
- S'assurer que les coûts des prestations sont correctement chargés et affichés.
- S'assurer que le programme détecte et signale correctement les sélections manquantes.
- Vérifier que le coût total du contrat est correctement calculé en fonction des sélections de l'utilisateur.
- S'assurer que le coût est ajusté correctement en fonction de l'âge de l'utilisateur.
- Vérifier que le numéro unique du contrat est généré correctement.
- S'assurer que les informations du contrat sont correctement insérées dans la base de données.

Par mesure de sécurité et garantir que les opérations de sélection de menu sont sécurisées, toutes les entrées utilisateur sont valides pour éviter les saisies incorrectes et les attaques par injection. Utilisation de *FUNCTION UPPER-CASE* pour s'assurer que les entrées correspondent aux valeurs attendues ('O' pour les options valides).

11 - Lecture de contrat :

11.1 - Description de la fonctionnalité :

Ce programme récupère dans la base de données le contrat qui est affecté à l'adhérent et affiche les informations du contrat avec une *SCREEN SECTION*.

Il gère également les erreurs dans le cas où l'adhérent n'a pas de contrat d'affecté ou au contraire si l'adhérent a plus d'un contrat d'affecté.

L'utilisateur peut consulter le numéro, le type et le montant du contrat ainsi que les taux de remboursement par catégories de soins.

11.2 - Organisation et description des fichiers :

La fonctionnalité lecture de contrat est composée de plusieurs fichiers:

- 'readcont.cbl' : Programme principal qui contient toute la logique de la lecture du contrat . Le programme permet de sélectionner des taux de remboursement spécifiques pour l'adhérent afin de personnaliser son contrat .
- 'screen-read-contrat.cpy' : *COPY* contenant la définition de l'écran de saisie / sélection pour l'utilisateur.

Le chemin du fichier '*manacust.cbl*' est '*CUSTOMER-MGMT/CONTRACT/READ-CONTRACT/readcont.cbl*'.

Le chemin du fichier '*screen-specific-contrat.cpy*' est '*CUSTOMER-MGMT/COPYBOOK/screen-read-contrat.cpy*'.

11.3 - L'interface utilisateur :

Le prénom, le nom et le numéro de sécurité sociale de l'adhérent sont affichés en haut à gauche de l'écran.

L'écran pour la lecture de contrat affiche également les informations suivantes:

- le numéro de contrat
- le type de contrat
- le montant du contrat
- le taux de remboursement pour les soins courant
- le taux de remboursement pour l'hospitalisation
- le taux de remboursement pour l'optique
- le taux de remboursement pour le dentaire
- le statut de l'adhérent (en couple ou non)
- le nombre d'enfant

11.4 - Logique de traitement :

Le programme est conçu pour récupérer les informations du contrat affecté à un adhérent depuis une base de données et les afficher à l'écran. Il gère également les erreurs liées à l'absence de contrat ou à la présence de multiples contrats pour un adhérent.

Pour la sélection du contrat Une requête SQL est exécutée pour sélectionner les informations du contrat basé sur l'UUID de l'adhérent et un curseur est déclaré pour récupérer les informations de remboursement.

Pour la lecture et le stockage des données, les résultats de la requête SQL sont stockés dans les variables de travail et le type de contrat est déterminé en fonction des trois premières lettres du numéro de contrat.

Pour l'affichage de l'écran de gestion d'erreur si l'adhérent n'a pas de contrat ou a plus d'un contrat affecté, un message d'erreur est affiché et l'utilisateur est redirigé vers le menu sinon, les informations du contrat sont affichées à l'écran.

L'utilisateur peut également choisir de retourner au menu.

11.5 - Instructions de tests unitaires et mesures de sécurité :

- Vérifier que les informations de l'adhérent sont correctement formatées pour l'affichage.

- Vérifier que les informations du contrat sont correctement récupérées depuis la base de données.
- Simuler un adhérent avec un contrat affecté et vérifier que les informations sont récupérées.
- Simuler un adhérent sans contrat et vérifier que le message d'erreur approprié est affiché.

Par mesure de sécurité et garantir que les opérations de sélection de menu sont sécurisées, toutes les entrées utilisateur sont valides pour éviter les saisies incorrectes et les attaques par injection. Utilisation de *FUNCTION UPPER-CASE* pour s'assurer que les entrées correspondent aux valeurs attendues ('O' pour les options valides).