
DATA MANAGEMENT

DOCUMENTATION TECHNIQUE

Sommaire

Ecran de connexion	3
Description de la fonctionnalité :	3
Organisation et description des fichiers :	3
L'interface utilisateur	3
Logique d'authentification de l'utilisateur.....	3
Instructions de tests unitaires et mesures de sécurité	4
Menu des fonctionnalités annexes	6
Description de la fonctionnalité :	6
Organisation et description des fichiers :	6
L'interface utilisateur.....	6
Logique de fonctionnement.....	6
Instructions de tests unitaires et mesures de sécurité	7
Statistiques de Boboniort.....	8
Description de la fonctionnalité :	8
Organisation et description des fichiers :	8
L'interface utilisateur	8
Logique de fonctionnement.....	8
Instructions de tests unitaires et mesures de sécurité	10
Charger un fichier client csv	11
Description de la fonctionnalité :	11
Organisation et description des fichiers :	11
L'interface utilisateur	11
Logique de fonctionnement.....	12
Instructions de tests unitaires et mesures de sécurité	12

Ecran de connexion

Description de la fonctionnalité :

L'écran de connexion est le premier écran accessible au lancement de l'application .

L'utilisateur doit entrer l'identifiant et son mot de passe qui lui ont été fournis afin de se connecter.

Organisation et description des fichiers :

Dans le dossier 'DATA-MGMT/USER/SIGN-IN' se trouvent les fichiers qui composent la fonctionnalité de connexion de l'utilisateur:

'*siback.cob*' : Ce programme contrôle l'identifiant et le mot de passe de l'utilisateur.

'sifront.cbl' : Ce programme affiche l'écran de connexion et appelle le sous programme 'siback'. Tant que l'identifiant et le mot de passe ne sont pas validés par siback, l'utilisateur reste sur cet écran.

Dans le dossier 'DATA-MGMT/Copybook' se trouve le fichier copy contenant la description de l'écran de connexion: le fichier '*screen-sign-in-data.cpy*' .

L'écran contient une zone de saisie de l'identifiant de l'utilisateur, une zone de saisie du mot de passe et un bouton Connexion.

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 :

- [illegible]

Logique d'authentification de l'utilisateur.

Cette fonctionnalité gère la validation de l'identifiant et du mot de passe saisis par l'utilisateur dans l'écran.

sifront affiche l'écran, lorsque l'utilisateur demande la connexion il appelle le programme siback. Selon la réponse de ce dernier, il appelle l'écran suivant ou affiche le message d'erreur de connexion: "Connexion echouee, verifier vos identifiants"

siback 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 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 CURSOR : Ferme le curseur

Instructions de tests unitaires et mesures de sécurité

- Test de Connexion Réussie :
 - 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 pour 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).

Menu des fonctionnalités annexes

Description de la fonctionnalité :

Le menu permet à l'utilisateur d'aller visualiser les statistiques de Boboniort ou de charger un fichier client de format csv.

Organisation et description des fichiers :

Dans le dossier 'DATA-MGMT/USER/MENU-USER' se trouve le programme affichant le menu:

'menudata.cbl' affiche l'écran menu et permet l'accès aux autres écrans.

Dans le dossier 'DATA-MGMT/Copybook' se trouve le fichier copy contenant la description de l'écran du menu: le fichier '*screen-menu-data.cpy*'.

L'écran contient deux champs de navigation et un bouton Quitter.

L'interface utilisateur.

L'écran contient 2 libellés de navigation: "Statistiques de Boboniort" et "Charger un fichier client csv". Chaque libellé est suivi d'un champ, respectivement WS-STAT et WS-FILE.

Le bouton "Quitter" ferme l'application, lorsque l'utilisateur saisit "o" dans le champs WS-EXIT.

Le champ SC-MESSAGE permet d'afficher un message d'erreur pour le cas où l'utilisateur ne sélectionnerait pas correctement sa navigation 'Erreur de saisie, veuillez sélectionner votre choix avec "o"'.

Logique de fonctionnement.

L'utilisateur aura le choix entre trois options :

- Quitter, dans ce cas il sort de l'application.
- Statistique de Boboniort, l'utilisateur passe à l'écran d'affichage des statistiques.
- Charger un fichier client csv, passe à l'écran de chargement du fichier individus.csv.

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

Instructions de tests unitaires et mesures de sécurité

Tests qui doivent fonctionner:

- L'utilisateur a bien accès au menu via la screen section qui s'affiche correctement avec toute les options possibles affichées et sélectionnables
- L'utilisateur peut sélectionner l'une des trois options proposées
- Une fois la validation effectuée, l'application passe à l'écran souhaité

Par mesure de sécurité et pour garantir que les opérations de sélection du 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).

Statistiques de Boboniort

Description de la fonctionnalité :

Cet écran affiche les statistiques effectuées depuis les données de la base.

Organisation et description des fichiers :

Dans le dossier 'DATA-MGMT/STAT' se trouve le programme affichant les statistiques de Boboniort:

'stfront.cob' calcule les statistiques et les affiche dans l'écran.

Dans le dossier 'DATA-MGMT/Copybook' se trouve le fichier copy contenant la description de l'écran d'affichage des statistiques: le fichier '*screen-stat-data.cpy*'.

L'écran contient plusieurs tableaux de données et un bouton Retour.

L'interface utilisateur .

L'écran contient un bouton "Retour" qui permet de revenir au menu et 4 tableaux de données:

- La répartition en nombre des clients par région
- Le taux d'activité en pourcentage (nombre de client actif divisé par le nombre de client par catégorie genre, marital)
- La répartition des clients en nombre et pourcentage par âge et genre.
- Les montants remboursés par catégorie, et le pourcentage par tranche d'âge)

Logique de fonctionnement.

L'utilisateur ne peut que revenir au menu des fonctionnalités annexes en choisissant le bouton "Retour".

L'écran affiche des statistiques calculées par le programme.

Pour ce faire, un accès à la base de données est exécuté et la récupération des données se fait via des curseurs.

La répartition régionale:

Nom du curseur: CRSREGION.

Le curseur fait une jointure entre les tables CUSTOMER et BOBO_REGION.

Il compte le nombre de clients (table CUSTOMER) dont le code département appartient à la région x (BOBO_REGION). Seuls les clients actifs et inactifs (CUSTOMER_ACTIVE) sont extraits. Les informations

sont triées en DESCENDING sur le numéro (x) de région. Ce numéro correspond aussi à la position de la région dans le tableau (équivalent au tri alphabétique sur le libellé de région).

L'application s'appuie donc sur ce numéro pour charger le tableau.

Le taux d'activité:

Nom du curseur: CRSACTIVITY.

Le curseur extrait des données de la table CUSTOMER.

Il compte le nombre de clients par genre (CUSTOMER_GENDER), activité (CUSTOMER_ACTIVE) et statut marital (CUSTOMER_COUPLE). Seuls les clients actifs et inactifs (CUSTOMER_ACTIVE) sont extraits.

L'application calcule le pourcentage par genre et statut marital de la manière suivante:

Nombre de clients actifs/somme des clients du genre et du statut marital traité.

L'application charge le tableau en fonction du genre et du statut marital de la donnée.

La répartition par âge et genre:

Nom du curseur: CRSAGE.

Le curseur extrait des données de la table CUSTOMER.

Il compte le nombre de clients par genre (CUSTOMER_GENDER), par tranche d'âge en s'appuyant sur la date de naissance du client (CUSTOMER_BIRTH_DATE). Seuls les clients actifs et inactifs (CUSTOMER_ACTIVE) sont extraits.

Utilisation d'un EVALUATE sur la valeur de AGE.

Pour les moins de 30 ans, l'application additionne les données réceptionnées avec une valeur AGE de 10 ans (tranche d'âge de 10 à 19 ans pour les plus jeunes clients) et de 20 ans (tranche de 20 à 29 ans).

Pour les plus de 70 ans, l'application additionne les données réceptionnées avec une valeur AGE de 70 ans (tranche d'âge de 70 à 79 ans) et autre (tranches supérieures à 70 à 79 ans).

L'application charge la donnée par âge et genre dans le tableau.

L'application calcule les totaux par genre, le total de clients, le total par tranche d'âge.

Elle calcule les pourcentages suivants comme suit:

pourcentage par genre: Nombre total de clients par genre / nombre total de clients.

pourcentage par tranche d'âge: Nombre total de clients par tranche d'âge/ nombre total de clients.

L'application charge dans le tableau les totaux et les pourcentages par genre, ainsi que les pourcentages par âge.

Les remboursements par catégorie:

Ce tableau est chargé en dur dans l'écran car la fonctionnalité remboursement n'est pas encore développée.

Ce tableau affiche les montants de remboursement totaux par catégorie et le pourcentage de remboursement pour chaque catégorie par tranche d'âge.

Instructions de tests unitaires et mesures de sécurité

Tests qui doivent fonctionner:

- L'utilisateur a bien accès à l'écran des statistiques via la screen section qui s'affiche correctement.
- Une fois le retour sélectionné, l'application passe au menu des fonctionnalités annexes.

Par mesure de sécurité et garantir que les opérations de sélection du 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).

Charger un fichier client csv

Description de la fonctionnalité :

Cette fonctionnalité permet de charger le fichier individus.csv fourni par le client.

Elle ne doit être activée qu'une seule fois afin de ne pas générer de doublons dans la table CUSTOMER.

L'application charge le fichier dans une table de travail (BOBO_CUSTOMER) de même structure que le fichier et contenant les champs nécessaires dans la table CUSTOMER mais manquants dans le fichier (TOWN et CODE_SECU).

L'écran affiche le chemin et le nom du fichier à charger, un bouton Valider et un bouton Retour.

Organisation et description des fichiers :

Dans le dossier 'DATA-MGMT/CHARGE-FILES' se trouvent les programmes qui composent cette fonctionnalité:

'IND-UPDATE.sql' ce script SQL reformate certaines informations dans la table BOBO_CUSTOMER, et les charge les données manquantes.

Puis, il charge la table CUSTOMER depuis la table BOBO_CUSTOMER.

Ce fichier doit être copié dans le répertoire d'exécution de l'application pour être utilisé.

'individus.csv' ce fichier est le fichier client csv à charger.

Ce fichier doit être copié dans le répertoire d'exécution de l'application pour être utilisé.

'cfback.cob' : Ce programme charge le fichier dans la table BOBO_CUSTOMER et exécute le script IND-UPDATE.sql.

'cfront.cbl' : Ce programme affiche l'écran de confirmation du chargement. Lorsque l'utilisateur choisit le bouton Valider, il appelle le programme 'cfback'. Si l'utilisateur choisit le bouton Retour, l'application revient au menu des fonctionnalités annexes.

Dans le dossier 'DATA-MGMT/Copybook' se trouve le fichier copy contenant la description de l'écran de chargement d'un fichier client csv: le fichier 'screen-charge-file.cpy' .

L'écran contient une zone d'affichage du chemin complet du fichier à charger, un bouton Valider et un bouton Retour.

L'interface utilisateur .

L'écran contient un bouton "Retour" qui permet de revenir au menu des fonctionnalités annexes, un bouton "Valider" qui permet de charger le fichier, et une zone d'affichage du nom complet du fichier à charger.

Logique de fonctionnement.

L'utilisateur, en choisissant "Valider" confirme la volonté de charger le fichier individus.csv.

Lorsqu'il choisit le bouton "Retour" il est redirigé vers le menu des fonctionnalités annexes.

L'écran affiche le chemin complet du fichier à charger.

Le programme lit le fichier individus.csv (s'il est dans le même répertoire que l'exécutable), et écrit dans la table de travail BOBO_CUSTOMER les enregistrements lus.

Puis il exécute le script IND-UPDATE.sql via la ligne de commande suivante:

```
CALL 'SYSTEM' USING 'psql -h localhost -d boboniortdb -U cobol -f IND-UPDATE.sql'.
```

Ce script charge le champ TOWN et le champ CODE_SECU de la table BOBO_CUSTOMER.

Il reformate les champs BIRTHDATE et PHONE de la table BOBO_CUSTOMER pour l'adapter au format utilisé dans la table CUSTOMER.

Il charge les données de la table BOBO_CUSTOMER dans la table CUSTOMER.

Instructions de tests unitaires et mesures de sécurité

Tests qui doivent fonctionner:

- L'utilisateur a bien accès à l'écran via la screen section qui s'affiche correctement.
- Une fois le retour sélectionné, l'application passe au menu des fonctionnalités annexes.
- Une fois le bouton "Valider" sélectionné, l'application charge le fichier dans la base.

Par mesure de sécurité et garantir que les opérations de sélection du 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).