



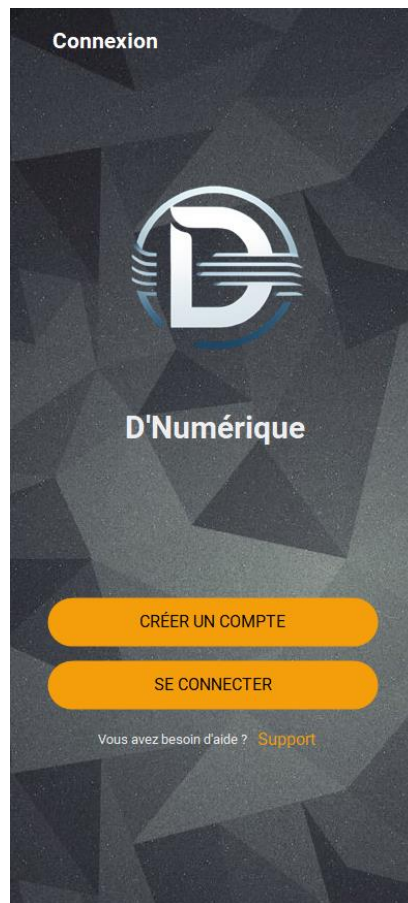
DOCUMENTATIONS

Utilisateurs / développeurs



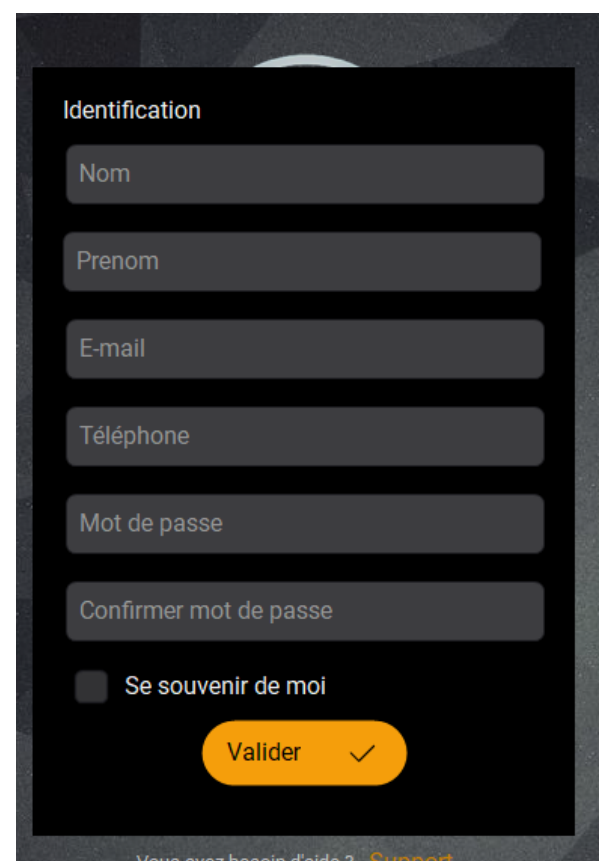
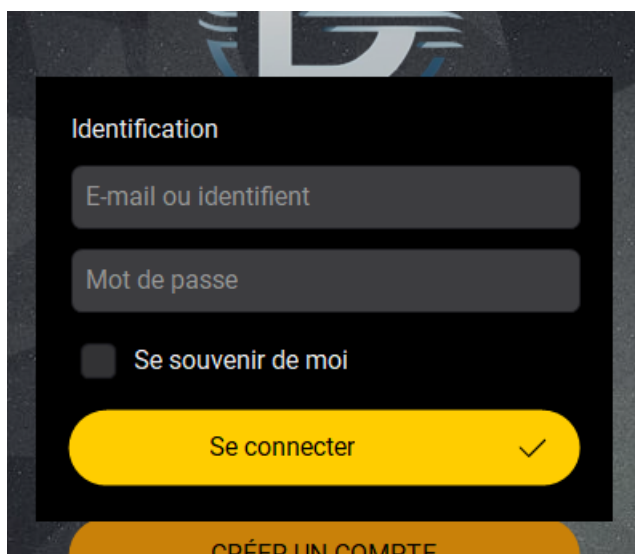
22 MARS 2024
LYCEE RENE CASSIN
67000 Strasbourg
4 Rue Schoch



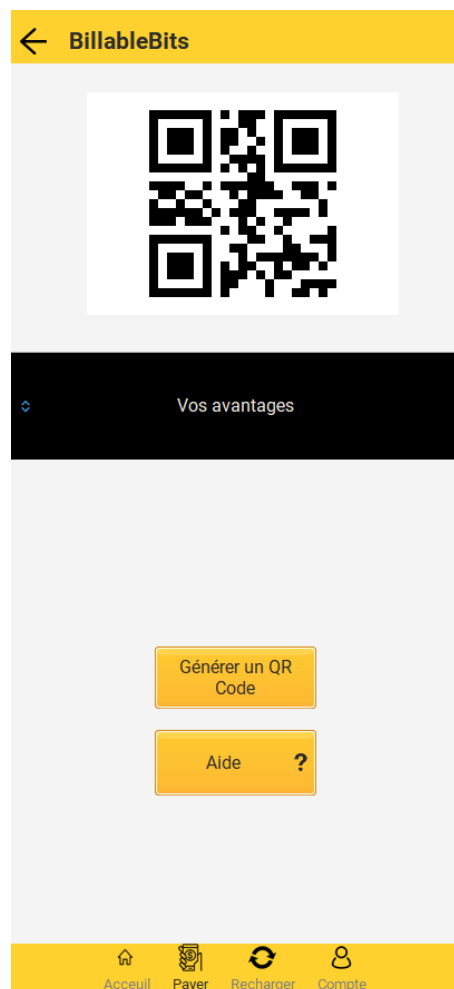


Ici nous avons la page de connexion qui permet de pouvoir créer un compte pour se connecter ou se connecter. Lors de l'inscription nous avons le pop-up ci-après où nous devons inscrire nos informations personnelles pour valider l'inscription.

Sur l'image en dessous, nous avons le volet de connexion pour pouvoir accéder à l'application grâce aux informations de l'inscription.



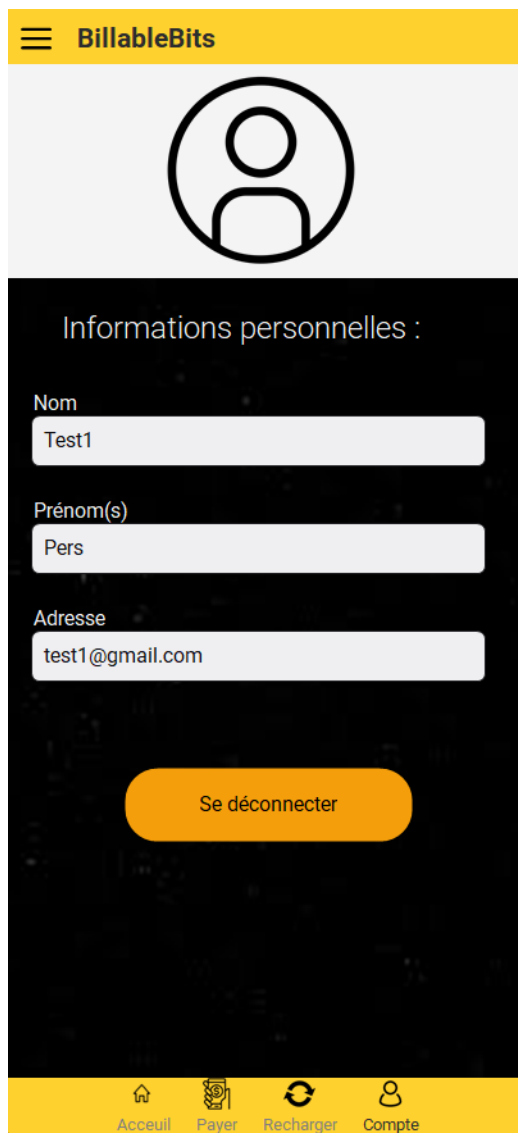
Ceci est la page d'accueil où le client pourra retrouver le crédit restant sur son compte. Ensuite il y a « Secondes » qui représente le temps passé et qui démarre lors d'un clic sur le bouton « Démarrer » et s'arrête lors d'un clic sur le bouton « Arrêter » et qui déduit directement le temps restant avec le temps passé. Et tout en bas, nous avons les différents onglets de l'application.



Là, nous pouvons retrouver l'onglet « Payer » qui permet aux utilisateurs de générer un QR Code que l'entreprise pour scanner pour identifier l'utilisateur.



Voici l'onglet « Recharger », où le client peut recharger le crédit sur son compte en choisissant le temps qu'il veut et en payant. Mais pour l'instant la méthode de paiement ne fonctionne pas.



BillableBits

Informations personnelles :

Nom
Test1

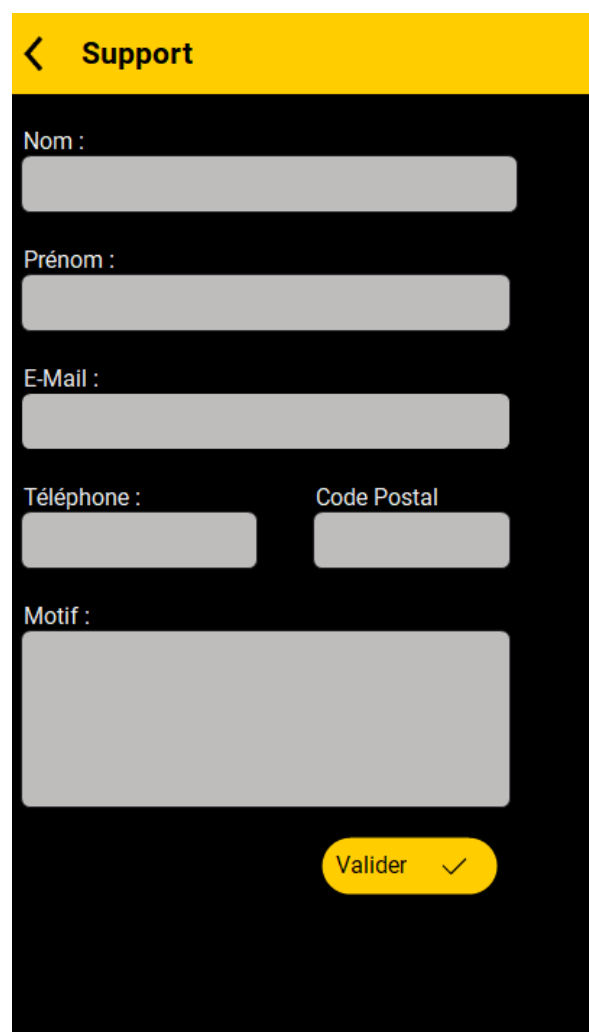
Prénom(s)
Pers

Adresse
test1@gmail.com

Se déconnecter

Accueil Payer Recharger Compte

Voici l'onglet « Compte », où le client peut retrouver ses informations personnelles mais aussi le bouton « Se déconnecter » pour se déconnecter de l'application.



Support

Nom :

Prénom :

E-Mail :

Téléphone : Code Postal

Motif :

Valider ✓

Et enfin, nous avons la page support que les utilisateurs pourront accéder avec les boutons « Aides » si jamais des problèmes surviennent et qu'ils pourront nous le communiquer.

```

P Initialisation de EpreuveE5-AppMobile / Si Erreur : par programme
// Déclaration des variables globales

// Variable enregistrant l'identifiant de l'utilisateur
GLOBAL
UserID est un entier sur 8 octets

// Variable enregistrant le temps choisi par le client
GLOBAL
ChoixClient est un entier

GLOBAL
DernierTempsRestant est un entier

```

Là nous pouvons retrouver les variables globales qui sont utilisées sur toutes les fenêtres de l'application et qui ne doivent pas être supprimées car cela pourrait provoquer des erreurs sur le fonctionnement de l'application.

```

Initialisation de BTN_Valider
// Version 1
// Description
// Bouton en validation

Clic sur BTN_Valider / Si Erreur : par programme
EcranVersFichier()
HAjoute(User)
Info("Ajout de " + SAI_Nom + " réussi")
SAI_Nom = ""
SAI_Prenom = ""
SAI_Email = ""
SAI_Telephone = ""
SAI_Mot_de_passe1 = ""
SAI_Mot_de_passe = ""

```

Ensuite, nous avons le code qui se trouve dans le bouton « Valider » sur la fenêtre d'inscription qui permet d'enregistrer les informations inscrites par l'utilisateur dans la base de données.

```

Initialisation de BTN_Se_connecter
Clic sur BTN_Se_connecter / Si Erreur : par programme / Quand Exce
|
HlitRecherchePremier(User, Mail, SAI_Email)
SI PAS HTrouve(User) ALORS
    ToastAffiche("L'utilisateur existe pas", toastCourt, cvMilieu)
    RepriseSaisie(SAI_Email)
FIN
SI User.Password1 <> SAI_Mot_de_passe ALORS
    ToastAffiche("Le mot de passe est incorrect", toastCourt, cvMilieu)
    RepriseSaisie(SAI_Mot_de_passe)
FIN

// Récupération de l'ID de l'utilisateur
UserID = User.IDUser

OuvreFenêtreMobile(FEN_Index_Client)

```

Ci-après, nous pouvons voir le code qui se trouve dans le bouton « Se connecter » qui permet de récupérer les informations de l'utilisateur si celui-ci existe et l'authentifie à l'application. Mais le code permet aussi de récupérer l'identifiant de l'utilisateur et le stocke dans une variable globale.

```

Procédure locale DecompteTemps
// Résumé : <indiquez ici ce que fait la procédure>
// Syntaxe :
// DecompteTemps ()
//
// Paramètres :
//   Aucun
// Valeur de retour :
//   Aucune
//
// Exemple :
// <Indiquez ici un exemple d'utilisation>
//
PROCÉDURE DecompteTemps()
SAI_Heure = HeureSys()
SAI_Sec = SAI_Sec + 1

```

Là, nous avons une procédure, qui permet aux secondes de marcher au même rythme que les secondes de l'heure.

```

Clic sur BTN_Arrêt   Si Erreur : par programme
FinTimerSys(1)

SAI_Temps_Restant = SAI_Temps_Restant - SAI_Sec

// Enregistre les données de l'écran dans le fichier
EcranVersFichier()

// Crée un nouvel enregistrement
MonEnregistrement est un Enregistrement de TempsClient
MonEnregistrement.TempsRestant = SAI_Temps_Restant
MonEnregistrement.IDUser = UserID

// Ajoute l'enregistrement à la table
HAjoute("TempsClient", MonEnregistrement)

```

Ici, nous pouvons retrouver le code dans le bouton « Arrêter » qui déduit le temps sur le compte de l'utilisateur avec le temps passé et ensuite enregistre le temps restant du client dans la base donnée.

BillableBits

Secondes :

0

Temps restants :

3 000

Vos avantages

Démarrer

Arrêter

Aide ?

Acceuil

Payer

Recharger

Compte

```

Clic sur BTN_Valider / Si Erreur : par programme
ChoixClient = SAI_Temps
Info("Ajout de " + SAI_Temps + "secondes réussi")

Ajouter d'autres événements à BTN_Valider

```

Dans ce code, nous avons enregistré le choix du temps du client dans une variable globale pour pouvoir l'utiliser sur une autre fenêtre.

```

Clic sur BTN_GénérerQRCodeClic / Si Erreur : par programme

PROCÉDURE BTN_GénérerQRCodeClic()
MonQRCode est un CodeBarres
MonQRCode.TypeContenu      = cbTypeTexte
MonQRCode.TypeCodeBarres   = BC_QRCODE
MonQRCode.Contenu          = "http://www.pcsoft.fr"
SI CBA_ChampCB.Visible = Faux ALORS
    CBA_ChampCB.Visible = Vrai
SINON
    // Forcer une mise à jour du contrôle
FIN

```

Sur l'image au-dessus, nous avons le code qui permet de générer le QR Code.

```

Récupération de la propriété Valeur de FI_Compte_Client
// Assurez-vous que l'ID de l'utilisateur connecté est disponible
ID_Utilisateur est un entier sur 8 octets = UserID

// Lecture de l'enregistrement de l'utilisateur connecté
HlitRecherchePremier(User, IDUser, ID_Utilisateur)

// Vérification de la présence de l'enregistrement
SI HEnDehors(User) ALORS
    // Transfert des informations de l'utilisateur vers les champs de saisie
    SAI_Nom      = User.Nom
    SAI_Prenom   = User.Prenom
    SAI_Adresse  = User.Mail
SINON
    // Gestion de l'erreur si l'enregistrement n'est pas trouvé
    Erreur("L'utilisateur avec l'ID " + ID_Utilisateur + " n'a pas été trouvé.")
FIN

```

Et enfin, nous pouvons retrouver ici le code qui permet de récupérer les informations personnelles de l'utilisateur et de les afficher sur les champs qui les représentent.