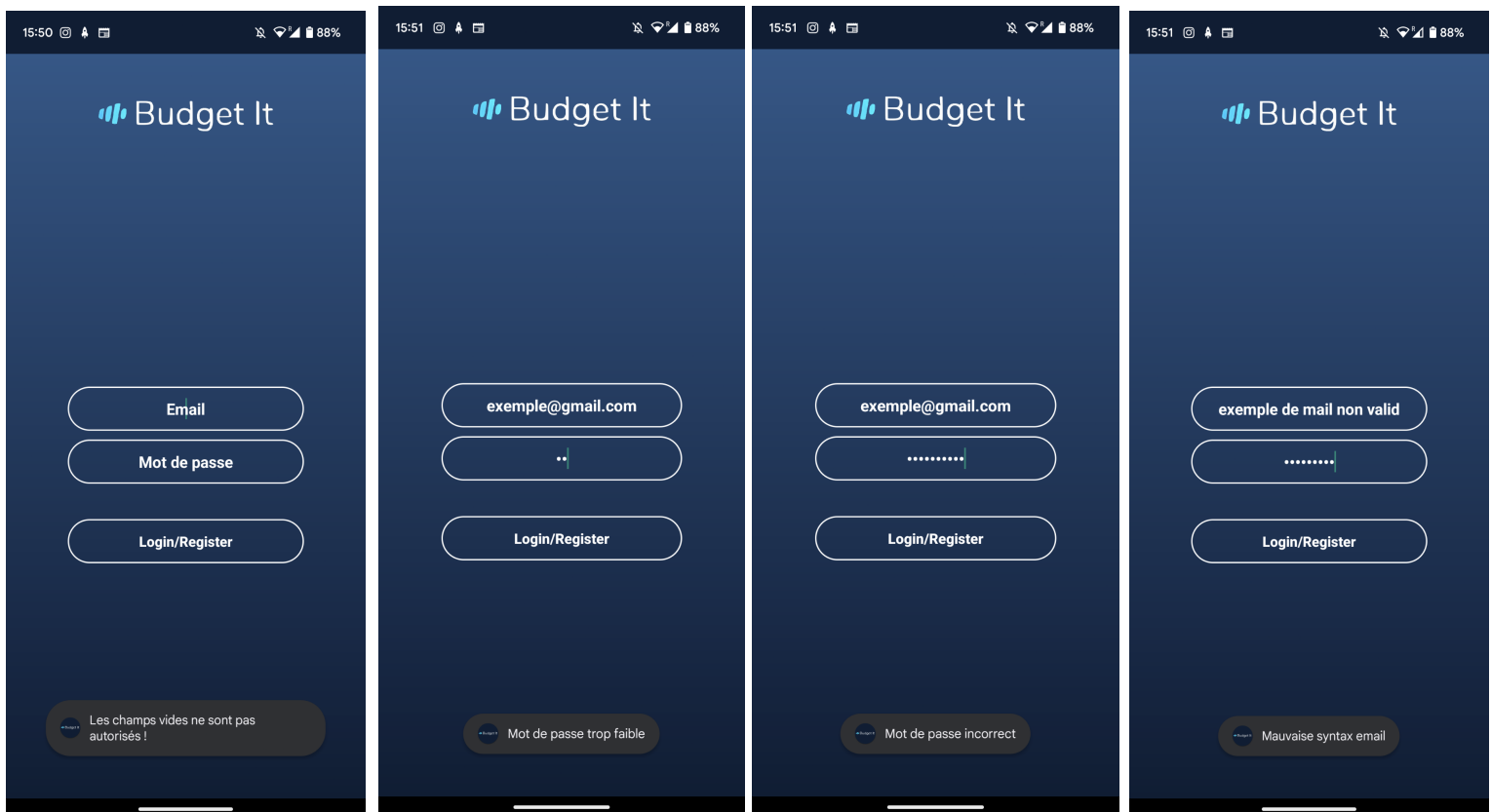


## I. Activités de notre application et leur interrelations

### A. Main activity / Page de Login

Cette classe gère la page de connexion de l'application, il est important que les utilisateurs puissent utiliser l'application de partout, c'est pourquoi on a utilisé Firebase Authentication pour se connecter à l'application. Elle inclut une vérification des champs de connexion pour empêcher les champs vides et utilise les erreurs renvoyées par Firebase pour gérer les problèmes d'authentification (par exemple, une adresse e-mail invalide ou un mot de passe trop faible). Elle gère également la création de nouveaux utilisateurs et la connexion des utilisateurs existants. La méthode `onStart()` est également utilisée pour vérifier si l'utilisateur est déjà connecté et, le cas échéant, le rediriger vers la page principale de l'application sans avoir à se reconnecter à chaque réouverture de l'application.

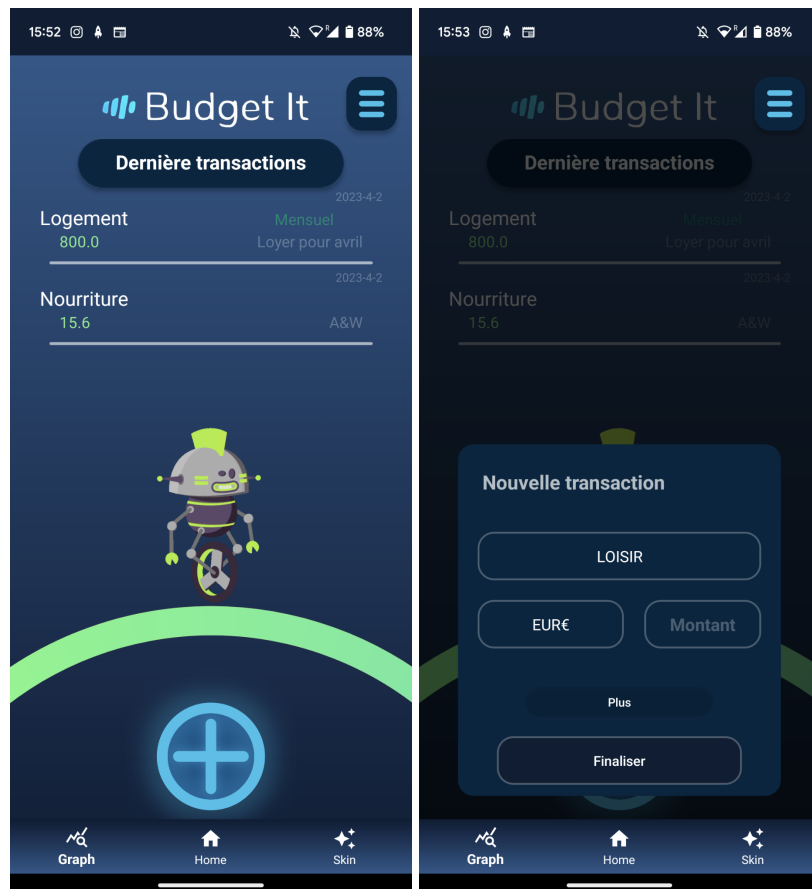


## B. Page accueil

La page d'accueil est l'activité la plus importante de notre projet. Elle comporte une liste des transactions les plus récentes, une barre de progression représentant le solde actuel de l'utilisateur en prenant en compte son objectif mensuel, un personnage représentant l'utilisateur, mais surtout le plus important : **un bouton plus**.

Ce bouton plus permet à l'utilisateur d'ajouter une transaction, qu'elle soit positive ou négative. Par défaut, on peut modifier le motif et le montant. Cependant, en appuyant sur "Plus" on peut modifier les informations suivantes de la transaction : sa description, sa date et si cette transaction sera effectuée mensuellement.

En plus des éléments mentionnés ci-dessus, la page d'accueil de notre application comporte également une barre de navigation en bas. Cette barre de navigation permet à l'utilisateur de basculer entre différentes pages, y compris la page des top dépense et la page des skins.



## C. Top dépense

La page top dépense de notre application vous permet de visualiser facilement vos dépenses les plus importantes par catégorie. Grâce à cette fonctionnalité, vous pouvez avoir une vue d'ensemble de vos habitudes de dépenses et identifier les domaines où vous pourriez réduire vos dépenses.

Le graphique sur la page top dépense vous donne une représentation visuelle de vos dépenses les plus importantes par catégorie. Les catégories sont représentées par des segments de couleurs différentes, et la taille de chaque segment correspond au montant total que vous avez dépensé dans cette catégorie. Vous pouvez passer votre souris sur chaque segment pour voir le pourcentage exact dépensé dans chaque catégorie.

Sous le graphique, il y a une liste des catégories. Chaque catégorie est un bouton qui, lorsqu'on clique dessus, affiche l'historique détaillé des transactions qui ont contribué à ce montant. Cette liste permet de visualiser les dépenses de l'utilisateur par catégorie. Chaque catégorie est accompagnée d'un bouton de retour. En cliquant sur ce bouton, l'utilisateur est ramené à la liste des catégories et peut ainsi choisir une autre catégorie pour voir ses transactions détaillées. Ce bouton est très utile pour naviguer facilement entre les différentes catégories sans avoir à revenir à la page précédente à chaque fois.



## **D. Skin**

Vous cherchez à donner un nouveau look à votre personnage dans votre application? Rien de plus simple avec notre sélection de skins exclusifs qui transformeront votre héros dans le style que vous aimeriez!

La page de skins est une fonctionnalité de notre application qui permet à l'utilisateur de personnaliser l'apparence de son personnage en achetant des skins avec les points qu'il a accumulés. Cette page présente une variété de skins avec des styles différents, des couleurs et des accessoires pour donner une apparence unique à votre personnage.

Cette page est également accompagnée d'un bouton de retour. En cliquant sur ce bouton, l'utilisateur est ramené à la page d'accueil, où il peut voir son personnage avec le skin. Cela permet une meilleure fluidité dans l'application.

## **II. Les services de votre application, et les tâches de fond s'il y a lieu.**

### **A. Transaction**

La classe "Transaction" contient des données pour suivre les transactions financières, comme le motif de paiement, le montant, la description, si la transaction est mensuelle ou non, et la date de la transaction. Les motifs de paiement sont définis dans une énumération. La classe a des fonctions pour ajouter une transaction, récupérer toutes les transactions, récupérer les transactions pour le mois en cours, supprimer une transaction, et calculer le montant total pour un motif de paiement donné. La classe utilise également une classe pour charger et sauvegarder les données de transaction à partir des préférences partagées. Enfin, la classe a une fonction pour effectuer des transactions mensuelles et mettre à jour le solde en conséquence.

### **B. Solde**

La classe Solde permet de gérer le solde d'un utilisateur dans l'application. Elle utilise Firebase Realtime Database pour stocker et récupérer les données du solde. La classe a des fonctions pour obtenir, modifier et définir le solde, ainsi que pour sauvegarder et charger le solde à partir de Firebase. La classe utilise également la bibliothèque Gson pour convertir les données du solde en format JSON pour les stocker dans Firebase.

## **C. Character**

La classe Character permet de “gamifier” l’application, elle initialise et actualise le personnage qu’on incarne. Sa vie, qui est définie par l’argent qu’on possède et qu’on doit maximiser pour gagner des niveaux. Chaque dépense fait perdre de la vie et chaque salaire ou virement vers notre compte redonnent de la vie. Elle permet aussi de définir l’apparence du personnage grâce aux skins. C’est aussi ici qu’on anime le personnage quand on clique dessus.

### **III. Les ressources de votre application et les points essentiels du fichier manifest**

En modifiant le fichier manifest de notre application, nous avons pu personnaliser l’écran de démarrage pour qu’il reflète l’identité visuelle de notre marque. Cette personnalisation permettra de créer une expérience utilisateur plus cohérente et mémorable.

On a aussi pu contrôler la rotation automatique ou non de l’écran, pour adapter et avoir le meilleur rendu et utilisation client.

### **IV. Les interfaces de votre application et les éléments d’expérience utilisateur pris en considération**

Pour permettre à l'utilisateur de saisir rapidement ses dépenses et ses revenus, l'interface utilisateur de l'application mobile doit être intuitive et facile à utiliser. Elle devrait également inclure des graphiques et des tableaux pour afficher les données financières de manière claire et concise.

Il est essentiel que l'application comporte des éléments d'expérience utilisateur tels qu'une navigation facile, des boutons clairs et une conception globale attrayante pour l'utilisateur. Les couleurs utilisées doivent être choisies avec soin pour créer un aspect visuel agréable et cohérent.

En veillant à ce que l'interface utilisateur de l'application soit conçue de manière efficace, les utilisateurs seront en mesure de tirer le meilleur parti de leur expérience financière.

Pour se différencier des autres application de gestion de ses finances, on a donné un aspect plus jeu

## **V. L'utilisation des capteurs et/ou de la connectivité dans votre application**

Nous utilisons le capteur de rotation de l'écran pour adapter l'interface à l'écran. Ainsi que la connexion internet pour s'authentifier avec la base de données. Nous prévoyons aussi d'utiliser le gyroscope pour donner plus d'animation au personnage et augmenter l'interaction avec le héros et donc accentuer le côté jeu.

## **VI. L'utilisation des bases de données dans votre application**

L'application mobile de gestion de budget utilise Firebase Realtime Database pour stocker et récupérer les données de l'utilisateur, y compris ses préférences de budget et son solde. Les données de l'utilisateur sont sauvegardées sur Firebase, ce qui permet à l'utilisateur de les récupérer facilement en se connectant à son compte personnel. La classe Solde gère spécifiquement le solde de l'utilisateur en utilisant Firebase Realtime Database et la bibliothèque Gson pour stocker et récupérer les données en format JSON. La connexion à son compte personnel est donc un élément clé de l'expérience utilisateur de cette application de gestion de budget.

### **Dans le futur:**

L'utilisateur de cette application mobile pourra accéder à ses données personnelles et à ses préférences de budget en se connectant avec son compte. Ainsi, toutes les données de l'utilisateur seront stockées sur Firebase et récupérées par l'application lorsqu'il se connectera. Cela permettra également à l'utilisateur de sauvegarder ses données en cas de perte ou de changement de téléphone. La connexion à son compte personnel est donc un élément clé de l'expérience utilisateur de cette application de gestion de budget.

## Tableau de répartition :

	Ewan	Tristan	Lorenzo	Léo	Salmana
Page d'authentification	0%	50%	15%	35%	0%
Page d'accueil	80%	15%	5%	0%	0%
Top dépenses	0%	100%	0%	0%	0%
Page de skin	0%	75%	25%	0%	0%
Class Transaction	0%	0%	0%	100%	0%
Class Solde	0%	0%	0%	100%	0%
Character et Skins	0%	0%	100%	0%	0%
Splash Screen	0%	100%	0%	0%	0%
Total	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>0%</b>