Moneythoring

Cahier des charges

HEIG-VD

Projet de semestre

# Descriptif

Que veut-on proposer ?

Description précise du projet.

# Fonctionnalités

## Fonctionnalités de base

### Création compte utilisateur

Si l’utilisateur ne possède pas de compte à l’ouverture de l’application, il peut sélectionner l’option “Créer un compte”. Celle-ci permet la création d’un compte utilisateur à l’aide d’une adresse email, d’un nom d’utilisateur et d’un mot de passe, qu’il est nécessaire de faire répéter deux fois à l’utilisateur pour confirmer ses informations. Aucune confirmation d’email ne sera demandée. Le compte est immédiatement créé et utilisable.

### Connexion sécurisée

Lors de l’ouverture de l’application, une connexion est exigée. Celle-ci doit être mise en place à l’aide d’une adresse email et d’un mot de passe. L’application ne nécessite pas de double authentification, puisqu’aucune information de compte bancaire ne sera spécifiée, si ce n’est le solde. En revanche le mot de passe doit être hashé dans la base de données.

### Compte bancaire

Ajout d’un compte avec un nom, un type (courant, épargne), un nom de banque, un montant actuel, un pourcentage d’intérêt, à utiliser ou non par défaut, ...

Suppression en virant ou non l’argent sur un autre compte. L’argent n’ayant pas été viré est “perdu” et sort du budget.

Modification du nom, du pourcentage d’intérêt. (La modification du montant se faisant automatiquement selon les entrées et sorties saisies par l’utilisateur)

### Catégorie

Un utilisateur a la possibilité de créer d’autres catégories que celles proposées par défaut. Il pourra lui définir un nom et une couleur.

Les catégories peuvent être modifiées et supprimées.

### Devise

Chaque utilisateur doit à la création d’un compte, spécifier sa devise principale. Il peut choisir la devise qu’il veut utiliser pour les revenus, les dépenses, les factures et les dettes. Une conversion automatique se fait en fonction de la devise du budget choisi, avec le taux de change du jour (si devises différentes).

### Transactions

Ajout d’un revenu ou d’une dépense avec un montant, une catégorie, un compte affecté (par défaut le compte courant), une devise utilisée, le taux de change de la journée (??)

Suppression d’un revenu dans le cas d’une erreur ou autre (remboursement, échange, etc.)

Modification du montant, du compte affecté, de la devise, de la catégorie, etc.

Récurrence pour les revenus et dépenses du genre salaire et factures. Cela peut être mensuel / annuel.

### Virement compte à compte

Il sera possible d’enregistrer des virements entre les différents comptes d’un utilisateur. Un virement sera pris en compte comme une transaction. Il y aura donc une dépense pour le compte à débiter et un revenu pour le compte à créditer.

### Dettes

Possibilité de calculer un éventuel intérêt lié à la dette, ou majorer le montant d’une sommation en cas de retard.

Dettes simples (ne concerne que le compte et est traitée entièrement manuellement) ou synchronisées (concerne un autre utilisateur qui devra confirmer les informations)

Simple / Sync : Comme revenu / dépense, avec possibilité de mettre une personne concernée et une description. Comporte, comme les factures, une deadline à respecter (notification à l’approche de cette dernière)

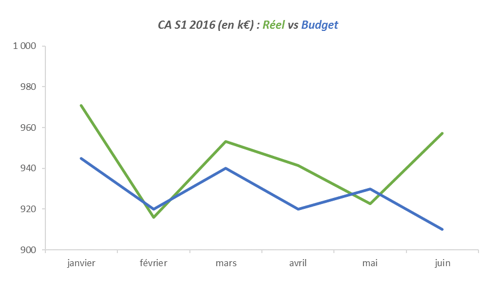
Sync : Concerne un compte existant (ou non, auquel cas spécifier un email) qui doit valider que la dette existe bien (qu’elle soit entrante ou sortante) entre les deux entités.

Une fois la dette acquittée, les deux partis doivent confirmer avant que la dette soit “classée”. Bien sûr, dans le cas où le deuxième parti ne répond pas (ou qu’il n’a pas de compte existant), mettre un délai pour invalider la chose. Une fois le délai écoulé, la dette est de toute façon classée.

### Vue globale

L’utilisateur peut en tout temps suivre l’évolution de son budget, grâce à une vue globale (depuis le début du budget jusqu’à maintenant), annuelle ou mensuelle. Cette vue permet de comparer de manière graphique les différentes dépenses et rentrées d’argent. En plus d’être filtrée chronologiquement, elle peut également l’être selon le mode d’affichage: graphique en courbe, diagramme circulaire, diagramme en barres.

**Graphique en courbe**

Le graphique en courbe permet la comparaison entre la courbe des dépenses et la courbe des rentrées d’argent selon la chronologie sélectionnée.    

**Diagramme circulaire**

Le diagramme circulaire affiche les pourcentages de dépenses par catégorie. Une information supplémentaire est toutefois ajoutée à côté de ce graphique, pour signaler le pourcentage de dépenses et rentrées d’argent globales pour la chronologie sélectionnée.



### Budget

Un budget se caractérise en deux types : les budgets mensuels (en fonction des catégories) et les budgets ponctuels (en fonction de deux dates données début/fin). Ces budgets permettent de voir l’évolution des dépenses dans un domaine précis, tout en permettant à l’utilisateur d’être notifié quand il s’approche de la limite fixée. Les budgets seront représentés de manière graphique sur une ligne, avec le budget qu’il lui reste (vert par exemple) et les dépenses déjà effectuées (rouge). L’utilisateur aura également la possibilité de créer des budgets partagés, où les dépenses de chaque utilisateur sont donc communes et le budget évolue en fonction des participants (par exemple un budget de voyage ou de commissions pour la famille).

Ajout d’un budget avec un type, un nom, un montant. Lors de l’ajout du montant, l’utilisateur peut choisir la devise qu’il souhaite mettre.

Si le budget est récurrent (mensuel) ou ponctuel, il faut sélectionner une ou plusieurs catégories, mais s’il est ponctuel, une durée est demandée (date de début et de fin).

Il est possible de supprimer n’importe quel budget. Dans le cas d’un budget partagé, il peut l’être uniquement par le créateur, ce qui affecte tous les membres y participant.

L’utilisateur peut modifier le montant, la devise et la durée du budget et les catégories d’un budget.

### Budgets partagés

Si le budget est partagé, l’utilisateur peut inviter d’autres utilisateurs (via leur nom de compte) à rejoindre un budget. Les personnes concernées recevront une notification pour accepter l’invitation.

Pour les budget partagés, l’utilisateur peut également décider de sortir du budget (sans le supprimer). Dans ce cas-là les dépenses effectuées par l’utilisateur restent enregistrées dans le budget.

Lors de l’ajout d’une dépense, l’utilisateur peut indiquer qu’il souhaite se faire rembourser par certains utilisateurs. Ceux-ci recevraient donc une dette envers celui qui a payé pour les autres.

## Fonctionnalités optionnelles

### Vue globale

**Diagramme en barres**

Le diagramme en barres permet la comparaison entre les dépenses et rentrées d’argent par mois et par année sous forme de deux barres de couleurs différentes.

****

**Chaque vue est exportable au format PDF.**

### Liste de souhaits

L’utilisateur peut créer une liste de souhaits, composée de différents articles qu’il aimerait prochainement acheter. Ces articles sont identifiés par un nom, éventuellement un lien internet et un prix.

Cette liste est mise à jour en fonction du budget actuel. Si l’épargne de l’utilisateur est suffisante, alors le produit est catégorisé comme étant achetable. Si au contraire l’utilisateur n’a pas suffisamment de fonds disponibles, une estimation de temps d’attente avant achat possible est calculée sur la base de l’évolution du budget.

### Prévisions d’achat

Liste de souhaits “prioritaire”, qui correspondrait à “il faudrait (absolument) que j’achète l’article X avant la date Y”, X étant un objet (p.ex un cadeau, un outil, une fourniture, …).

La dépense serait effective une fois que l’utilisateur aurait confirmé avoir effectué cet achat.

L’application envoie un mail la veille de la date limite, pour rappeler à l’utilisateur de faire son achat, ou pour confirmer la dépense s’il l’a déjà effectuée.

### Dettes

Possibilité de scanner les dettes et les documents en lien avec la dette et les lier à celle-ci. Fichier à titre informatif, on ne le lit pas ou autre.

### Simulation

Une simulation est possible en se basant sur les revenus et les dépenses récurrentes (salaire, abonnements, factures, loyer, capitalisation du compte, ...)

Peut-être également prendre en compte les dépenses qui réapparaissent plus ou moins régulièrement sans pour autant être récurrent (notamment les courses qui ne sont pas forcément à un rythme régulier) ?

Un utilisateur peut modifier ses dépenses et revenus en mode simulation, s’il souhaite voir comment évolue son budget sans pour autant sauvegarder les modifications réalisées. L’activation de ce mode est repérée par des indications visuelles, permettant à l’utilisateur de ne pas se tromper de mode.

# Mockups

# Architecture

Quelle mise en œuvre / architecture ?

Quelle DB ?

# Implémentation

Quel langage / interface graphique ?

Java + JavaFX