



## INF8405 – Informatique mobile

### Travail pratique 1 - Application de suivi des dépenses

#### 1. Informations générales

Session	Hiver 2025
Chargé de laboratoire	Groupe 1 - Bilal Itani (bilal.itani@polymtl.ca)
Date de début	22 janvier 2025
Date de remise	1 mars 2025
Laboratoire	En ligne
Pondération	15%
Directives particulières	<ol style="list-style-type: none"><li>1. Tout projet rendu sera pénalisé de <b>10 points</b> s'il est soumis par une équipe dont la taille est différente de (4) étudiants, sans l'approbation préalable du chargé de laboratoire.</li><li>2. Rapport à rendre en format <b>PDF</b>.</li><li>3. Le code source + le rapport sont à rendre dans une archive <b>ZIP</b>, avec le fichier .APK de l'application</li><li>5. Soumission du rapport et des travaux réalisés par Moodle uniquement (<a href="http://moodle.polymtl.ca">http://moodle.polymtl.ca</a>).</li><li>6. Tout retard dans la soumission du travail réalisé mènera à une pénalité de <b>4 points par jour de retard</b></li></ol>
Auteur(s) originaux	Bilal Itani
Révision par	-

#### 2. Connaissances préalables

- Langage Java ou Kotlin (programmation orientée objet, développement d'interfaces graphiques)
- Langage XML
- Utilisation de l'API Android

### 3. Environnement et outils nécessaires

- Environnement de développement intégré (IDE) : Android Studio
- Telephone/Tablette Android

### 4. Objectifs du laboratoire

L'objectif principal de ce laboratoire est d'initier l'étudiant au développement de services pour des terminaux mobiles. De manière spécifique, au terme de ce laboratoire, il s'agira pour l'étudiant de :

- Développer une application pour terminaux mobiles;
- Se familiariser avec les caractéristiques des applications mobiles;
- Se familiariser avec un système d'exploitation pour terminaux mobiles.

Ce travail pratique consiste, par la même occasion, à évaluer deux des 12 qualités de l'ingénieur définies par le BCAPG (Bureau canadien d'agrément des programmes de génie). Le Bureau d'agrément a pour mandat d'attester que les futurs ingénieurs ont atteint ces 12 qualités à un niveau acceptable. Les deux qualités en question sont:

**Qualité 4 (Conception)** : capacité de concevoir des solutions à des problèmes d'ingénierie complexes et évolutifs et de concevoir des systèmes, des composants ou des processus qui répondent aux besoins spécifiés, tout en tenant compte des risques pour la santé et la sécurité publiques, des aspects législatifs et réglementaires, ainsi que des incidences économiques, environnementales, culturelle et sociales.

**Qualité 7 (Communication)** : habileté à communiquer efficacement des concepts d'ingénierie complexes, au sein de la profession et au public en général, notamment lire, rédiger, parler et écouter, comprendre et rédiger de façon efficace des rapports et de la documentation pour la conception, ainsi qu'à énoncer des directives claires et y donner suite.

## 5. Présentation de l'application

L'application de suivi des dépenses est conçue pour aider les utilisateurs à gérer efficacement leur budget personnel. Elle permet l'enregistrement des transactions financières, la catégorisation des dépenses, et offre des outils d'analyse pour une meilleure compréhension des habitudes de consommation. Les fonctionnalités principales comprennent :

- la répartition des dépenses et des revenus par catégories, avec des vues journalières, hebdomadaire, mensuelle et annuel;
- la représentation graphique des dépenses et revenus selon les catégories ;
- L'affichage du solde du compte courant principale

Cette application vise à fournir une interface intuitive et personnalisable, facilitant ainsi la maîtrise des finances personnelles pour les utilisateurs. Vous pouvez consulter l'annexe pour un [exemple d'application similaire existante](#) qui offre des fonctionnalités similaires pour vous en inspirer.

## 6. Requis

### 6.1.Requis fonctionnels

#### Qualité évaluée :

##### 4.3 Procéder à la conception

*Critère d'évaluation : Intégrer les concepts de programmation en réseautique retenus au premier laboratoire en répondant aux besoins et en respectant les requis fonctionnels du projet courant.*

Au lancement de l'application, une trame est affichée pendant 5 secondes avec le logo de votre application et les noms des membres de votre équipe. Une fois le délai passé, l'utilisateur arrive sur la page principale, où il peut inscrire ces dépenses et ses revenus.

De manière spécifique, les fonctionnalités attendues de l'application sont :

#### 6.1.1 - Page principal - Accueil

- Le montant total du compte principal doit être visible à l'utilisateur;
- Un bouton permettant d'ouvrir le menu de navigation;
- Deux onglets sont disponibles: **Dépenses** et **Revenus**
- L'onglet **Dépenses** est choisi par défaut
- Une barre de navigation horizontale doit être disponible afin de filtrer les dépenses **journalières, hebdomadaires, mensuel et annuel**
- La période choisie par défaut est '**hebdomadaire**'
- Un graphique de type '**Pie chart**' colorié en fonction des catégories doit être affiché
- Dans le centre du 'Pie chart', il faut indiquer le montant total des dépenses ou des revenus pour la période choisie
- Un bouton '+' doit être disponible pour permettre à l'utilisateur d'ajouter une dépense ou un revenu
- La période visée doit également être affichée à l'écran. Par exemple, si on choisit 'Hebdomadaire', il faut indiquer la semaine visée. (e.g Jan 12 - Jan 18) pour la semaine du 12 au 18 janvier.
- Une liste des dépenses / revenus par catégories est indiquée pour la période visée

### 6.1.2 - Page d'ajout des dépenses ou des revenus

- Un champ est disponible afin d'y indiquer la valeur de la dépense ou du revenu
  - ◆ Ce champ doit uniquement accepter une entrée numérique
- L'utilisateur doit choisir la catégorie de la dépenses ou du revenu
- Un bouton doit être disponible pour permettre à l'utilisateur de se rendre à la page des catégories pour en ajouter une au besoin.
- L'utilisateur doit indiquer la date de la dépense/revenu
- Sauvegarder la dépense/revenu localement
  - ◆ *Informez vous sur la librairie 'Room', pour pouvoir enregistrer les données dans une BD locale*

### 6.1.3 - Menu de navigation

- Une option '**Accueil**' qui amène à la page principale
- Une option '**Catégories**' qui amène à la page des catégories
- Une option '**Rapports**' qui amène à la page des graphiques

### 6.1.4 - Page des catégories

- Deux onglets sont disponible dans cette page: Dépenses et Revenus
- L'onglet '**Dépenses**' est sélectionné par défaut
- La liste des catégories est affiché
- Un bouton '+' permettant d'ajouter une catégorie est disponible
- Cliquer sur le bouton '+' amène l'utilisateur à la page d'ajout d'une catégorie

### 6.1.5 - Page d'ajout d'une catégorie

- Un champ de type texte permettant d'entrer le nom de la catégorie
- Une option de type '**radio button**' permettant de choisir si la catégorie est de type '**Dépenses**' ou '**Revenus**'
- Une option permettant de choisir la couleur de la catégorie
- Un bouton permettant d'ajouter (sauvegarder) la catégorie

### 6.1.6 - Page des rapports

- Deux onglets sont disponible: **Dépenses** et **Revenus**
- Par défaut, l'onglet '**Dépenses**' est choisi
- Une barre de navigation horizontale doit être disponible afin de filtrer les dépenses/revenus **journalières**, **hebdomadaires**, **mensuel** et **annuel**
- La période choisi par défaut est '**hebdomadaire**'
- Le graphique affiché est de type '**stacked bar chart**' où on affiche les dépenses en \$ de la période visée en utilisant les couleurs des catégories respectives.
- Le graphique doit afficher l'information sur six unités de temps. Par exemple, si le filtre choisi est 'journalier', le graphique doit montrer les dépenses ou revenus des 6 derniers jours incluant le jour courant. Si le filtre choisi est 'hebdomadaire', le graphique doit afficher les dépenses par semaine, des 6 dernières semaines. La même logique s'applique pour le filtre mensuel et annuel.
- Lorsqu'un utilisateur clique sur une barre du graphique, les détails des dépenses / Revenus sont affichés au bas du graphique

**Référez-vous à l'annexe pour un exemple visuel des requis susmentionnés.**

## 6.2 Requis non fonctionnels

- L'application sera développée pour une plate-forme Android. À des fins de compatibilité, la version de la plate-forme à considérer est **Android 7 (API 24 - Nougat)** et versions supérieures. Le minSdkVersion est donc **Android 7 (API 24 - Nougat)**;
- L'application doit être accessible sur les terminaux mobiles notamment les tablettes et téléphones.
- L'application doit être facile d'utilisation. Un soin doit être accordé au système de navigation et à l'interface en général;
- La production d'un code lisible et suffisamment documenté (commentaires du code) est vivement attendue. Le développement doit se faire avec l'IDE Android Studio.

## 7. Livrables

### 7.1 Langages et bibliothèques autorisés

- L'application doit être développée uniquement en Java ou en Kotlin.
- Usage permis de librairies externe. Si vous utilisez ces librairies, veuillez justifier dans la section « Présentation » de votre rapport la raison de leur utilisation.

Tout non-respect de ces consignes entraînera la note de 0.

### 7.2 Soumission

Outre la présentation de l'application (démonstration), les livrables à fournir à terme seront regroupés dans une archive (ZIP) ayant le format suivant:

**INF8405\_TP1\_matriculeW\_matriculeX\_matriculeY\_matriculeZ**

Votre archive contiendra les fichiers suivants :

- Le projet Android Studio de l'application incluant les fichiers sources (.java), autrement dit le dossier en entier contenant votre projet;
- Le rapport au format PDF;
- **Le fichier contenant votre application (.apk).**

En cas d'absence d'un élément parmi les livrables, le travail ne sera pas corrigé et vaudra à toute l'équipe la note de 0. Le non-respect des autres consignes est sujet à pénalité.

## 7.3 Rapport

### Qualité évaluée :

#### 7.1 Lire et rédiger de la documentation

*Critère d'évaluation : Rédiger un rapport technique documentant efficacement le travail d'ingénierie réalisé dans ce projet en utilisant différentes formes de langage (naturel, informatique, etc.)*

Le rapport, d'une longueur maximale de 3 pages (excluant la page de présentation), doit comporter les éléments suivants :

- **Page présentation** qui doit contenir le nom ou le logo de l'école, le libellé et l'identifiant du cours, la session, le numéro et l'identification du projet, la date de remise, les matricules et noms des membres de l'équipe, la mention « Soumis à : **nom et prénom du chargé de laboratoire** ».
- **Introduction** en vos propres mots pour mettre en évidence le contexte et les objectifs du TP.
- **Présentation** de vos travaux. Une explication de votre solution mettant en lumière la prise en compte des principaux requis du système. Si vous utilisez des configurations particulières des bibliothèques ou des projets, précisez-les également.
- **Difficultés rencontrées** lors de l'élaboration du TP et les éventuelles solutions apportées.
- **Critiques et améliorations** : Il serait intéressant d'inclure vos suggestions pour améliorer le laboratoire.
- **Conclusion** : Expliquez en quoi ce laboratoire vous a été utile, ce que vous avez appris, si vos attentes ont été comblées, etc.



## 8. Évaluation

La grille d'évaluation est la suivante :

Rubriques	Points
<b>Évaluation de l'exécutable</b> : respect des requis, aucun arrêt abrupte, convivialité et des interfaces, etc.	<b>12</b>
Évaluation de l'implémentation : gestion adéquate de toutes ressources (création, utilisation, libération), gestion des erreurs, logique de développement, <b>documentation du code</b> , etc.	<b>3</b>
Rapport	<b>4</b>
Conformité de la remise (remise du fichier .apk)	<b>1</b>
<b>Total des points</b>	<b>20</b>

## 9. Annexe

Pour commencer, prenez le temps de consulter les tutoriels Android suivant afin de créer une nouvelle application, l'émuler sur un émulateur Android et la débbugger:

Create an interactive Dice Roller app

<https://developer.android.com/codelabs/basic-android-kotlin-compose-build-a-dice-roller-app#0>

Run your first app on the Android Emulator

<https://developer.android.com/codelabs/basic-android-kotlin-compose-emulator?hl=en#0>

Use the debugger in Android Studio

<https://developer.android.com/codelabs/basic-android-kotlin-compose-intro-debugger?hl=en#0>

Je vous invite également à consulter Android Codelabs, ce site contient beaucoup de tutoriels guidé pouvant vous aider dans la création de votre application:

<https://developer.android.com/get-started/codelabs>

**Exemple d'application de gestion des dépenses**

Les figures ci-bas sont tirées de l'application Money Manager & expenses disponible sur Google Playstore ici:

[https://play.google.com/store/apps/details?id=ru.innim.my\\_finance&hl=en\\_CA](https://play.google.com/store/apps/details?id=ru.innim.my_finance&hl=en_CA)



Fig. 1 : Un exemple de page principale

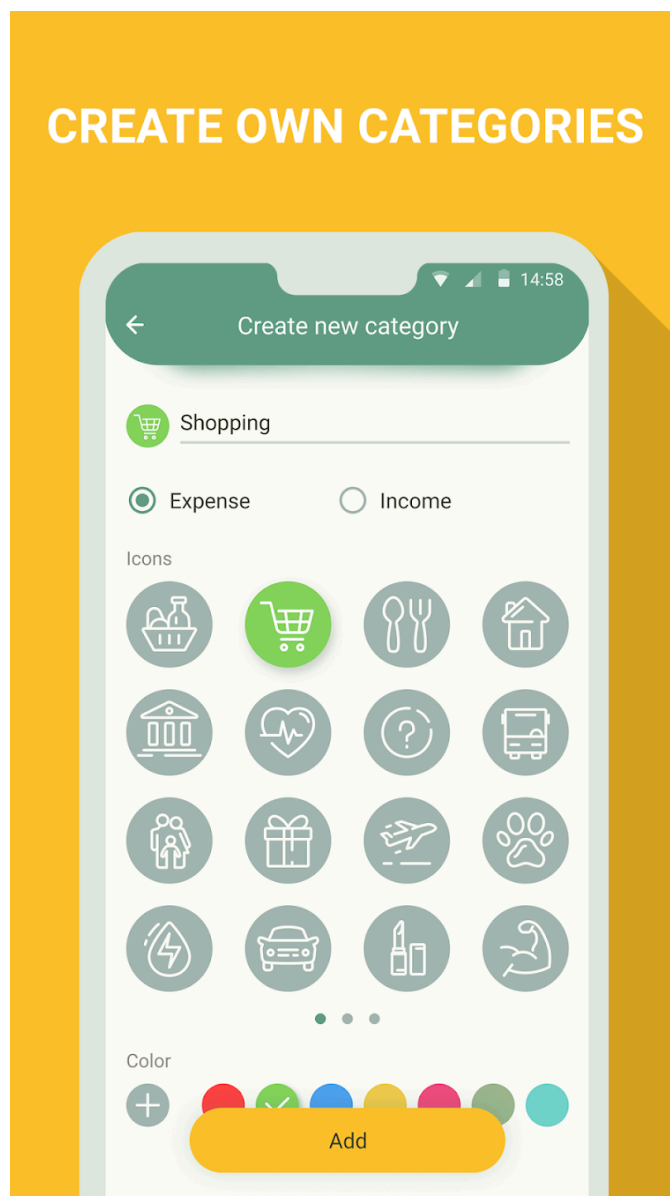


Fig. 2 : Exemple de la page de création des catégories

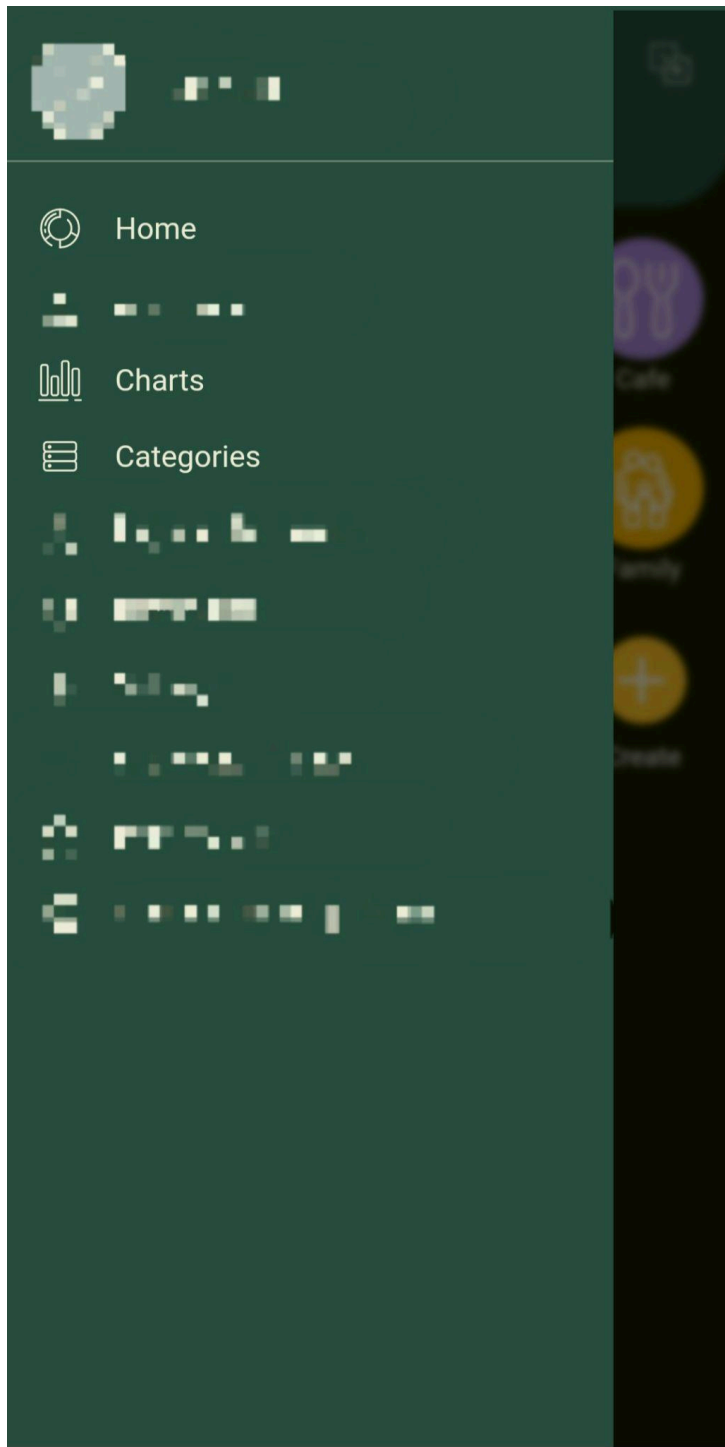


Fig. 3 : Exemple du menu

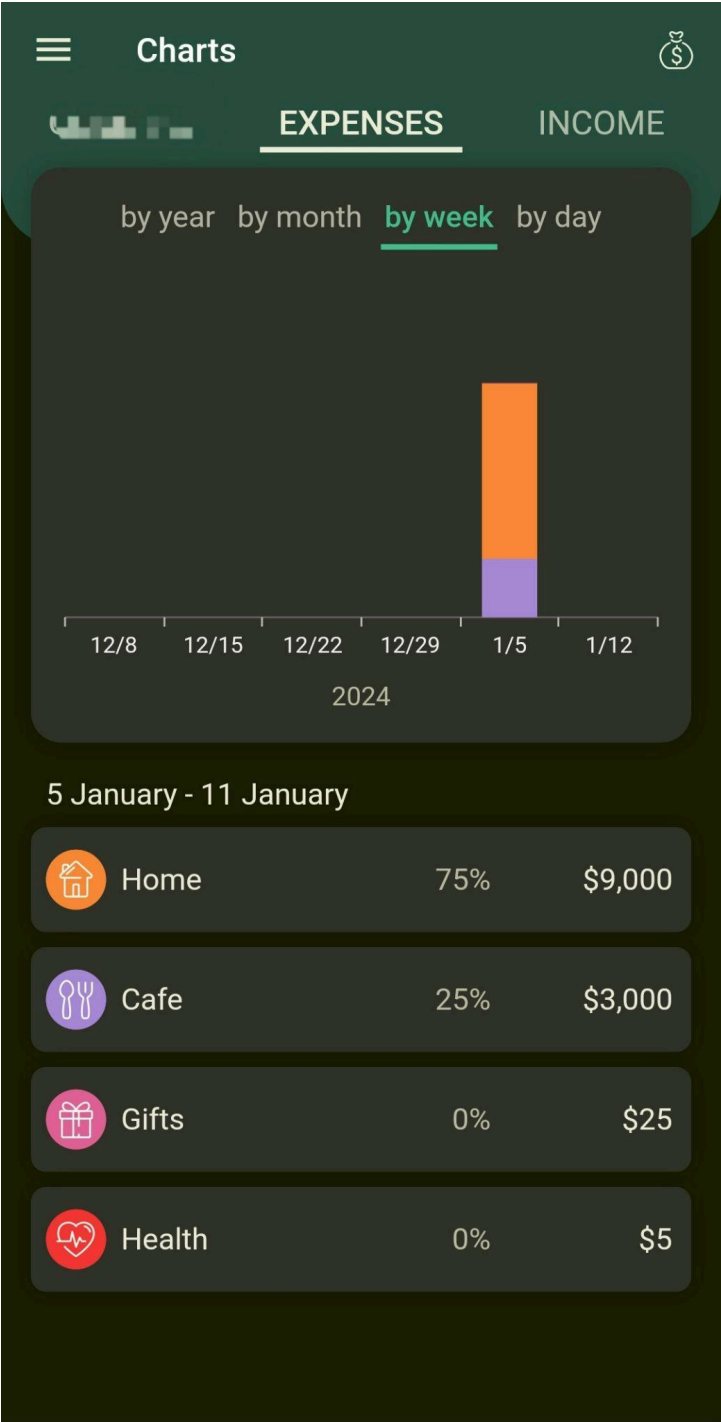


Fig. 4: Exemple de la page des rapports

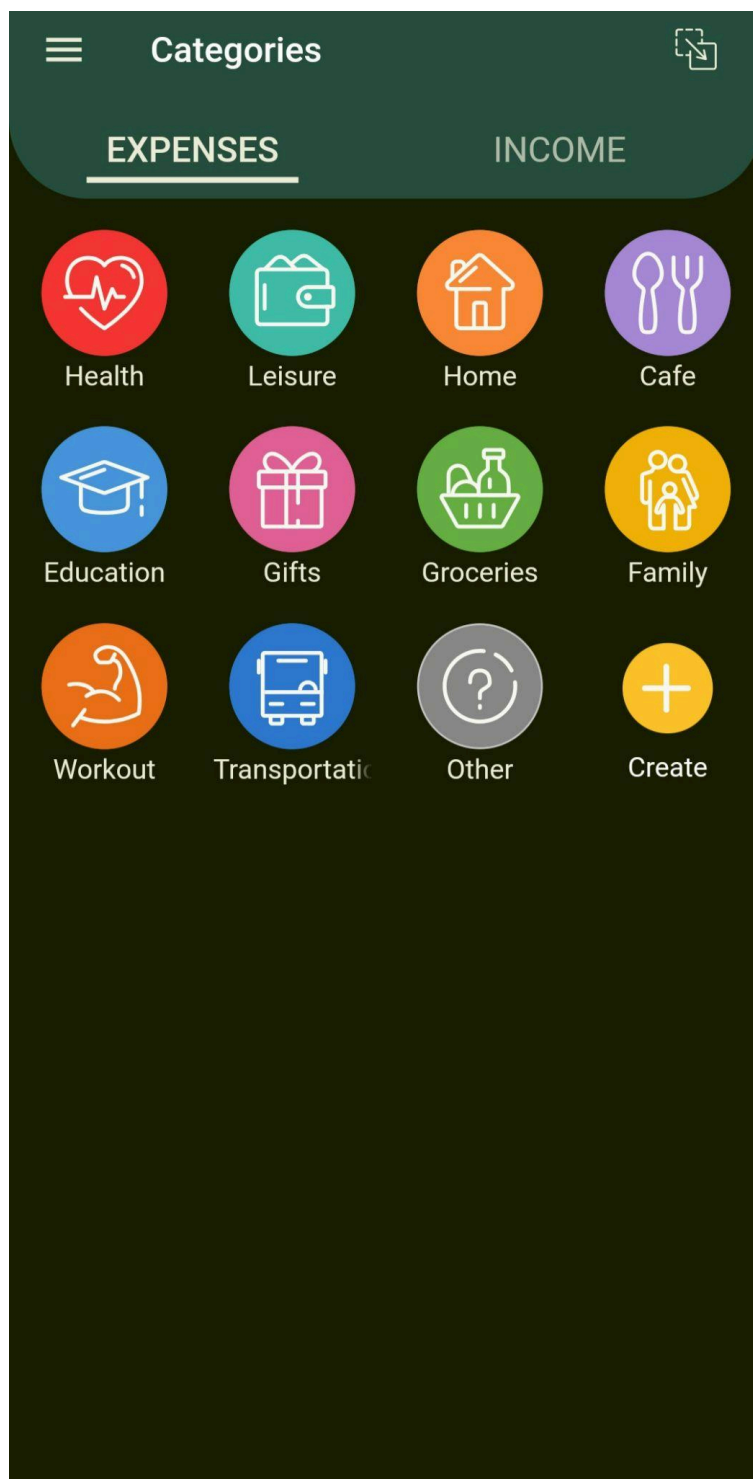


Fig. 5: Exemple de la page des catégories

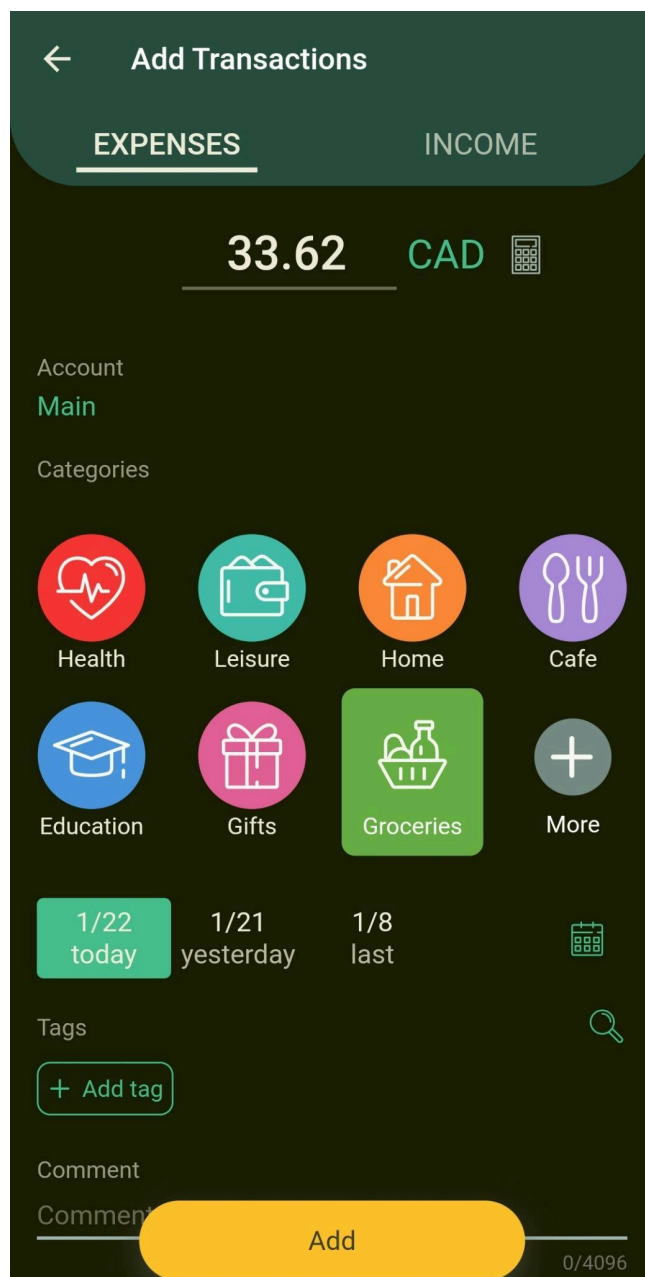


Fig. 6 : Exemple de la page d'ajout d'une transaction