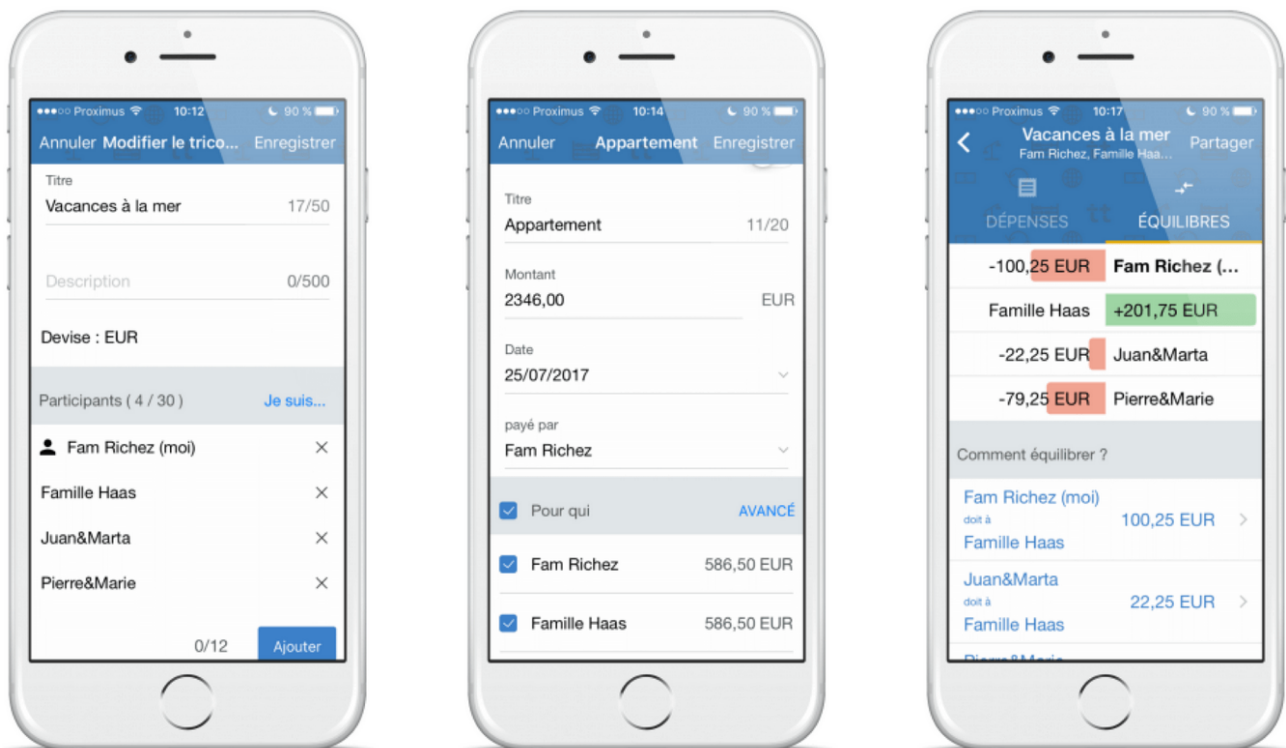


## Présentation

Le responsable de votre entreprise souhaite créer son propre système de gestion de comptes entre amis à la manière de [tricount](#) (ou [pumpkin](#), [sliptwize](#), [Flooz](#), [Splid](#) ou plus éloigné [lydia](#)). Cette application est idéale pour gérer un séjour entre amis ou gérer une période avec des co-locataires. Notamment elle permet de garder la trace de toutes les dépenses, savoir qui a payé quoi, et in fine, savoir qui doit quelle somme à qui pour solder les comptes. Seules les personnes qui se sont enregistrées (qui ont créé un compte avec login/mdp) peuvent créer un évènement et entrer des écritures. Les autres participants sont uniquement caractérisés par un pseudo. L'évènement est caractérisé par son titre, sa description et les pseudos des participants. Chaque dépense réalisée est indiquée dans l'application avec son libelle, le montant, la date et par qui elle a été payée (parmi les participants inscrits) et pour qui il a été payé (tous ou une partie des participants). L'application permet bien sûr, à tout moment, de calculer les dépenses cumulées de chacun mais surtout d'indiquer le nombre minimal de transactions à réaliser pour équilibrer les comptes entre tous les participants. Le créateur de l'évènement peut autoriser ou non les autres participants ayant un compte à ajouter eux-mêmes des écritures dans cet évènement (partage de responsabilités). Il n'y a pas de lien entre les budgets des événements, même si un individu participe à plusieurs événements, il est vu comme un nouveau participant à chaque fois. Il n'y a pas non plus d'équilibrage de budget entre les événements. C'est le créateur de l'évènement qui "ferme" l'évènement quand il le souhaite. Il n'est alors plus possible d'entrer une quelconque écriture. Evidemment, quand on a un compte, il est néanmoins possible d'accéder à tous les événements en cours ainsi qu'à l'historique des événements passés.



On pourra trouver [ici](#) ou [là](#) une discussion sur les différentes solutions possibles.

## Travail à faire

1. Le travail peut être réalisé en binôme, mais les responsabilités doivent être établies. A la fin, une feuille des tâches avec son responsable doit être impérativement rendue (utilisation de <https://trello.com/> ou onglet "project" sous Github, ou "Issues" sous [GitLab](#))

2. Etablir le MCD et la base permettant de gérer cette application. C'est le coeur du problème ! Comment stocker l'information ? comment l'interroger ?
  3. Créer la base et la remplir avec quelques données pertinentes pour tester les cas critiques
  4. Faites sobre et modeste au départ. inutile de faire joli ... l'important est déjà d'avoir une base et la page de saisie d'écritures austère mais fonctionnelle pour que chacun puisse travailler ! Tout le reste peut être complété par la suite ... Authentification, notation, CSS, Pool, page d'équilibre financier, page d'administration d'un évènement, chiffrement SSL, etc ...
1. Une documentation décrivant le projet réalisé, les choix faits et les technologies employées devra être fournie en PDF. Elle comprendra notamment 4 parties totalement indépendantes :
    - (a) Déploiement. Décrit l'installation et le déploiement de ce projet. Plus cette documentation sera courte, mieux ce sera. Notamment, le déploiement sur une machine donnée, avec un nom de contexte donné et un nom de base donné doit nécessiter le moins de travail possible.
    - (b) Synthèse
      - i. Objectif du travail
      - ii. Ce qui a été fait
      - iii. Les améliorations à apporter
      - iv. En quoi ce projet était difficile
      - v. Conclusion
    - (c) La description technique du projet
      - i. Objectif techniques du travail
      - ii. Principe de réalisation
      - iii. Difficultés techniques rencontrées et solutions apportées.
      - iv. Conclusion
    - (d) La documentation utilisateur (tutorial)
      - i. Objectif de cette application
      - ii. Du point de vue de l'utilisateur

## Contrôle du travail

Le projet sera réalisé avec l'ensemble des technologies J2E. Le projet doit tourner en salle TP et vous devez pouvoir en faire une démonstration. L'attention sera particulièrement portée sur les technologies que vous mettrez en oeuvre pour réaliser ce site, notamment :

- Conception d'objets "Métier" réutilisables hors web.
- Spring ou au minimum organisation MVC "à la main"
- JPA ou au minimum organisation DAO "à la main"
- Feuilles de styles externalisées (CSS, Bootstrap, Pure, ...)
- "responsive design" : Fonctionnement sur téléphone portable aussi bien que sur station de travail
- Entêtes mises en commun (fichiers includes, les logos DA2I et Lille1 doivent figurer sur toutes les pages ...)
- Système de trace d'activité du serveur (Logger, Valve...)
- Ergonomie générale du site,
- Code coté client vérifiant autant que possible les informations saisies (JavaScript)
- Validation HTML des pages par Tidy ou par le W3C. Présence de l'icône de test sur chaque page.
- Eviter l'injection SQL
- Système d'authentification automatique(Spring Security , Realm, ....)
- Accès à la base via un pool de connexions paramétrable
- Cryptage des informations via un serveur sécurisé (SSL,...)
- Centralisation de tout ce qui pourrait changer (logo, nom de la formation, base de données etc .. web.xml ou application.properties)

Une fois cela réalisé, rendre sur Moodle une archive portant vos noms (ex DurandDupont.zip) et contenant la doc, un fichier README pour le déploiement, les scripts SQL si nécessaire (création automatique préférable) et le WAR de votre application.