



## **Mini Projet “Kungry”**

Informatique Mobile et Applications  
GIF-3101

# Table des matières

Table des matières	2
Objectifs	3
Introduction	3
Énoncé	3
Rapport 1 - Planification MP	4
Rapport 2 - Avancement MP	4
Rapport 3 - Bilan MP	5
L'Application "Kungry"	5
API et Documentation	8
Pagination	9
Dossier de remise (Rapports)	10
Dossier de remise (Application)	10
Barème de correction	11
Annexe A - Schéma de navigation de l'application	12
Annexe B - "Flow" de connexion	13
Annexe C - Exemple rapport d'analyse	14
Annexe D - Exemple Rapport d'avancement	15
Annexe E - Exemple Rapport de bilan	16

# Objectifs

À la fin de ce mini-projet, l'étudiant devrait:

- Développer une application mobile avec les contraintes réelles du marché(API, Design, etc)
- Implémenter soi-même les capteurs
- Être en mesure mettre en applications les nombreux concepts présentés dans le cours

# Introduction

Le mini-projet dans le cours d'Informatique Mobile et Application (IMA) se veut un projet qui solidifie les concepts de développements mobiles. Il se veut également une simulation d'un projet tout à fait réel qui impose un minimum de structure et de gestion.

# Consignes

- Vous **devez obligatoirement** respecter les éléments de conformité qui sont énumérés sur le portail des cours dans la section ""Contenu et activités" -> Guides -> Conformité".
- Un guide de procédure de remise est disponible sur le portail des cours dans la section "Contenu et activités" -> Guides. Veuillez vous y référer pour vous assurer d'une remise conforme.
- Le travail doit être réalisé en équipe de 4
- Le travail doit être réalisé en Kotlin avec un maximum d'environ 20% de JAVA.

# Énoncé

Les livrables du mini projet sont découpés en 2 grandes catégories: L'application et les rapports de gestion. Le nombre de rapports de gestion est au nombre de 3. Par contre, le format des rapports étant très simple, si vous utilisez un bon système de gestion et que vous le maintenez à jour à chaque fois que vous travaillez sur votre projet, vos rapports s'écriront pratiquement d'eux-mêmes.

## Rapport 1 - Planification MP

Le but de ce rapport est de présenter votre compréhension du mini projet et ses tâches associées. Ce rapport devrait permettre de voir le temps dont vous pensez avoir besoin pour réaliser l'application, découpé par fonction et par membre de l'équipe.

Concrètement voici les informations qui sont attendues dans ce rapport:

- Nom des membres de l'équipe incluant IDUL et NI
- Un tableau "*Temps fonctionnalité*" qui présente le Cahier de produit(tâche et description) et le temps associé pour chaque tâche et/ou sous-tâche.
- Un tableau "*Temps coéquipier*" qui présente le nom de chacun des coéquipiers et le temps estimé pour réaliser le projet.
- Chacun des 2 tableaux doit avoir un total et bien entendu le total doit être équilibré (être égal entre les 2 tableaux)

Le format est libre à vous, mais dans l'optique de vous simplifier la tâche un tableau d'exemple est présent à l'annexe C de ce document.

## Rapport 2 - Avancement MP

Le but de ce rapport est de présenter un avancement réel en lien avec votre Mini-Projet. Avec ce rapport vous devriez être en mesure d'avoir une idée plus réaliste du temps réel à investir pour pouvoir terminer le mini-projet.

Concrètement voici les informations qui sont attendues dans ce rapport:

- Nom des membres de l'équipe incluant IDUL et NI
- Un tableau qui reprend le tableau "*Temps fonctionnalité*" du premier rapport et qui ajoute 3 nouvelles colonnes: État, Temps réel et temps réestimé pour chaque tâche et/ou sous-tâche du cahier de produit.
- Un tableau qui reprend le tableau "*Temps coéquipier*" et qui ajoute 2 nouvelles colonnes: Temps réel et Temps réestimé.
- Encore une fois chaque tableau doit fournir des totaux pour chacune des colonnes qui contiennent des chiffres et bien entendu le total doit balancer (être égal entre les 2 tableaux).
- Un petit paragraphe qui expose votre "impression" en lien avec votre avancement, avancez-vous comme prévu? Est-ce que vous rencontrez des problématiques non prévues? Vous avez simplement été trop débordé dans d'autres cours et n'avez pas eu le temps?

Le format est libre à vous, mais dans l'optique de vous simplifier la tâche un tableau d'exemple est présent à l'annexe D de ce document.

## **Rapport 3 - Bilan MP**

Le but de ce rapport est de présenter un bilan de votre réalisation du Mini Projet.

Concrètement voici les informations qui sont attendues dans ce rapport:

- Nom des membres de l'équipe incluant IDUL et NI
- Un tableau qui reprend le tableau "*Temps fonctionnalité*" du premier rapport et qui ajoute 2 nouvelles colonnes: Temps réestimé lors du rapport d'avancement et temps réel pour chaque tâche et/ou sous-tâche du cahier de produit.
- Un tableau qui reprend le tableau "*Temps coéquipier*" et qui ajoute 2 nouvelles colonnes: Temps réestimé lors du rapport d'avancement et Temps final.
- Encore une fois chaque tableau doit fournir des totaux pour chacune des colonnes qui contiennent des chiffres et bien entendu le total doit balancer (être égal entre les 2 tableaux).
- Un petit paragraphe qui expose les conclusions du projet. Aviez-vous prévu assez de temps? Qu'aviez-vous minimisé? Qu'aviez-vous exagéré?

Le format est libre à vous, mais dans l'optique de vous simplifier la tâche un tableau d'exemple est présent à l'annexe E de ce document.

## **L'Application "Kungry"**

**Prenez tout d'abord connaissance de la maquette haute-fidélité:**

<https://adobe.ly/3aOhdZ6>

L'application Kungry est une application pour aider quelqu'un qui a faim à trouver un restaurant qui l'intéresse. L'application se découpe principalement en 3 onglets :

- Carte des restaurants
- Liste des restaurants à proximité
- Mon profil

La complexité de navigation est relativement simple dans cette application et des sous-niveaux sont disponibles uniquement lorsqu'un utilisateur consulte un profil de restaurant. Il pourra voir les commentaires associés à ce restaurant et pourra laisser un

commentaire s'il a un compte dans l'application. Il pourra également se créer un compte Kungry de manière très simple.

Afin de simplifier votre compréhension de l'application, nous avons inclus un schéma complet de l'application à l'Annexe A et B et avons fourni une description textuelle de la mécanique de la page ici même.

La présentation visuelle de chaque page doit être *très près de la* page fournie dans la maquette. Considérant que les spécifications du design sont fournies et que les images/icônes sont téléchargeables directement du lien de maquette vous avez tout en main pour reproduire avec exactitude l'application. Si vous rencontrez des difficultés d'implémentation qui semble insurmontable, utilisez votre créativité et proposez une alternative au client (en l'occurrence ce professeur).

Les capteurs doivent être implémentés directement par vous, à l'exception près de la position GPS dans la carte qui elle peut venir directement du SDK de Google. Il n'y a pas de restriction en termes de librairie, vous pouvez donc utiliser ce que vous jugez utile à l'avancement de votre TP.

## **1.0 Carte des restaurants**

Cette Section permet de voir les restaurants à proximité dans une carte qui affiche également notre position courante. Chaque restaurant est représenté par une "pin" qui est clivable. Une fois sélectionnée elle permet d'afficher au bas de l'écran un résumé des informations du restaurant sélectionné. Toute la zone résumée est cliquable et permet d'ouvrir la page de détails d'un restaurant.

### **1.1 Détail d'un restaurant**

La page de détail d'un restaurant permet d'afficher les différentes informations associées à un restaurant incluant des liens cliquables (site web et téléphone) qui doivent effectuer les actions respectives associées. La page présente également au maximum les 5 évaluations les plus récentes et un bouton permet d'afficher une page qui affiche toutes les évaluations qui ont été laissées sur ce restaurant. Il est également possible de laisser une évaluation, par contre il faut être connecté pour pouvoir effectuer cette action.

### **1.2 Liste complète des évaluations**

La page complète des évaluations permet de voir toutes les évaluations laissées en lien avec un restaurant. Sur une évaluation

s'il y a une photo d'attaché, il est possible d'ouvrir cette photo pour la voir en plein écran. Finalement, il est possible à partir de la liste des évaluations de laisser une évaluation sur le restaurant qui est actuellement consulté. Prenez note qu'il y a un concept de pagination sur cette liste.

### **1.3 Création d'une évaluation**

La page de création d'une évaluation est une page accessible uniquement une fois connectée. Elle permet de sélectionner un nombre d'étoiles pour notre évaluation et de manières facultatives d'ajouter une photo en utilisant la méthode de votre choix ou un commentaire qui sera associé à notre évaluation. À chaque évaluation correctement ajoutée, le nombre total d'évaluations associé et affiché dans notre compte sera incrémenté.

## **2.0 Liste des restaurants à proximité**

La section de liste des restaurants à proximité doit afficher une liste des restaurants du plus près au plus loin en distance à vol d'oiseau. Pour ce faire un API permet de recevoir la position (latitude, longitude) courante d'un appareil et s'occupe de retourner la liste ordonnée. La présentation de l'information dans chaque cellule est disponible dans la maquette et chaque cellule est cliquable et permet d'ouvrir la page de détails de restaurant tout comme la carte l'offre. Prenez note qu'il y a un concept de pagination sur cette liste.

## **3.0 Mon compte**

La page mon compte fait partie des sections uniquement accessibles pour un utilisateur connecté. Cette page permet uniquement de consulter les informations en lien avec mon profil et de me déconnecter. Une déconnexion devrait effacer le token qui est sauvegardé en mémoire sur le téléphone.

### **3.2 Création de compte**

La page de création de comptes permet de créer un compte à partir de simplement un courriel et un mot de passe. Si le courriel existe déjà dans la base de données, une erreur de type HTTP 409 sera retournée. Si la création réussit avec succès, un Token sera alors retourné. Ce token qui expire après 24h, nécessitera d'être persisté dans l'application via les SharedPreferences. En théorie le type de token utilisé nécessiterait un concept de refresh token pour rafraîchir un token et en générer un nouveau sans nécessiter un mot de passe. Par contre, dans le cadre du Mini-Projet cela ne sera pas nécessaire de supporter cette mécanique. Autrement dit,

si le token est expiré vous pouvez simplement afficher la page de connexion qui elle demandera la connexion.

### 3.1 Connexion

La page de connexion est pratiquement identique à celle de création de comptes à l'exception près du libellé du bouton. Lors d'une connexion avec succès un Token est alors retourné, ce dernier doit être inclus lors des appels protégés dans le header d'Authorization. Le détails est fourni dans la documentation et la script Python de test.

## API et Documentation

Une documentation d'API est disponible avec le projet sous la forme d'un Swagger. Cette documentation est disponible à cette URL <https://kungry.ca/docs/>

Également afin de vous fournir le maximum d'outil pour vous permettre de progresser le plus aisément possible nous avons inclus un script python qui permet d'effectuer un exemple d'appel réseau aux API et d'effectuer une action. Cela vous permet donc de tester certains cas et également de voir le détail de l'appel à l'API.

Pour faire fonctionner le script il est nécessaire d'avoir Python2 ou 3 avec la librairie `requests` d'installé. Les scripts sont interactifs, il suffit donc de sélectionner l'un des chiffres parmi les options offertes pour que l'applicatif s'exécute. Voici un exemple d'exécution.

```
Log action in file (Yes/No) [No]:
Choose the action you wish to execute:
1. Account Info
2. Add Review Image
3. All Restaurant Reviews
4. Create Account
5. Create Review
6. Login
7. My Reviews
8. Refresh Token
9. Restaurant Details
10. Restaurants En
11. Restaurants Fr
12. Search Restaurant Combo
13. Search Restaurant Geolocation
14. Search Restaurant Text
15. Exit
Select one:6

=====
----REQUEST-----
METHOD : POST
URL : https://kungry.ca/api/v1/account/login
HEADERS : {'User-Agent': 'API Test', 'Content-Type': 'application/json', 'Content-Length': '266'}
BODY : {"email": "test1581783183@test.com", "password": "test1234", "client_id": "ST04WED2NTDDxJLs80Dios5M1SHwsrRlydsMalt0", "client_secret": "Y0VWGpjSnHdSAYDxGBR2CI009ZYM10PJGnH3ijkKwrUMVvwlprU
mLf6F6kx0u0C1UKTAE1SAeZn36V9QYBvTtrLMrtUxVux0GwleQGyApCzad69L36TdLXFqAG1tpk"}

----RESPONSE-----
URL : https://kungry.ca/api/v1/account/login
STATUS CODE : 404
HEADER : {'Connection': 'keep-alive', 'X-Frame-Options': 'SAMEORIGIN', 'Allow': 'POST, OPTIONS', 'Content-Type': 'application/json', 'Vary': 'Accept-Language, Cookie', 'Server': 'nginx/1.10.3 (U
buntu)', 'Content-Language': 'en', 'Date': 'Thu, 27 Feb 2020 03:32:14 GMT', 'Content-Length': '150'}
DURATION : 0:00:00.192919
RAW RESPONSE : {"content":null,"meta":{"status_code":404,"error":{"display":"Such account does not exists.", "details":[{"detail":"Such account does not exists."}]}}
=====
```

Finalement en lien avec l'API et la documentation, il est nécessairement de savoir le Client ID et le Client Secret de



l'application. Ces éléments sont nécessaires pour établir des communications sécurisées.

Client ID: STO4WED2NTDDxjLs8ODios5M15HwsrRlydsMa1tO

Client Secret:

YOVWGpjSnHd5AYDxGBR2CIB09ZYM1OPJGnH3ijkKwrUMVvwLpr  
UmLf6fxku06CIUKTAEI5AeZN36V9QYBYvTtrLMrtUtXVuXOGWle  
QGYyApC2a469I36TdIXFqAG1tpK

## Pagination

Sur certaines méthodes d'APIs, il y a des concepts de pagination qui sont présent. Dans le présent projet les APIs de GET /restaurant et GET /restaurant/{ID}/reviews ont de la pagination. Vous pouvez également reconnaître ses APIs avec leur JSON qui contiennent un en-tête qui permet de savoir combien d'éléments il y a et si une prochaine page est disponible. Voici un exemple d'en-tête.

```
{
  "content": {
    "count": 16,
    "next": 2,
    "previous": null,
    "results": []
  }
}
```

Le détail de l'en-tête de ce JSON se lit comme suit:

- Count : permet de dire combien d'éléments au total sont disponibles
- Next : indique le numéro de la page suivante. Si aucune page suivante n'est disponible, la valeur sera null.
- Previous: indique le numéro de la page précédente. Si aucune page n'est avant la page courante retournera un null.
- Results: Array qui contient nos objets.

Pour accéder à une page précise, il suffit de passer un argument "page" dans l'URL avec le numéro de page que l'on désire accéder. Par exemple si on désire accéder à la page #2 de la liste des restaurants on effectuerait cette requête GET /api/v1/restaurant/?page=2. Par défaut si aucun numéro de page n'est saisi la page #1 sera affichée. Autrement dit la requête GET /api/v1/restaurant/?page=1 reviendra au même que /api/v1/restaurant/

Lors de l'utilisation de la pagination, le but n'est pas d'avoir une boucle qui itère sur chaque page avant d'effectuer l'affichage, mais bien d'utiliser l'affichage pour donner l'impression à l'utilisateur que les données ont toutes été chargées, mais en réalité elles le sont au fur et à mesure. Pour se faire, une des techniques possibles

est d'ajouter une cellule "loading" dans notre liste qui est la dernière cellule et lorsque celle-ci devient visible et/ou deviendra visible on charge la prochaine page. Ainsi on donne accès rapidement au contenu à l'utilisateur et ce dernier peut voir tout le contenu, et ce sans limite.

Il est possible de voir l'exécution d'une boucle de chargement de pages multiples dans le script python fourni avec ce TP en exécutant la méthode "All Restaurant Reviews".

## **Dossier de remise (Rapports)**

La nomenclature de votre dossier de remise doit être de cette forme et uniquement de cette forme:

- Uniquement un fichier nommé "Rapport.pdf"

La remise de votre travail doit être effectuée sur le portail des cours. Une boîte de dépôt est disponible dans la section "Évaluations et résultats".

## **Dossier de remise (Application)**

La nomenclature de votre dossier de remise doit être de cette forme et uniquement de cette forme:

- Fichier zip avec votre nom d'équipe(Ex: Equipe 1) qui contient
  - L'application en format APK nommé "MP.apk"
  - Le bundle git nommé "MP.bundle"

La remise de votre travail doit être effectuée sur le portail des cours. Une boîte de dépôt est disponible dans la section "Évaluations et résultats".

## **Gestion des équipes**

Clause "Shutgun" instaurée dans le cadre du mini-projet

- Cette clause permet à une équipe de demander le retrait d'un de ses coéquipiers.
- Vous pouvez vous prémunir de cette clause au minimum 14 jours avant la remise du code source. Lorsqu'il restera moins de 14 jours pour la remise, il sera considéré trop tard et toute demande sera automatiquement refusée.

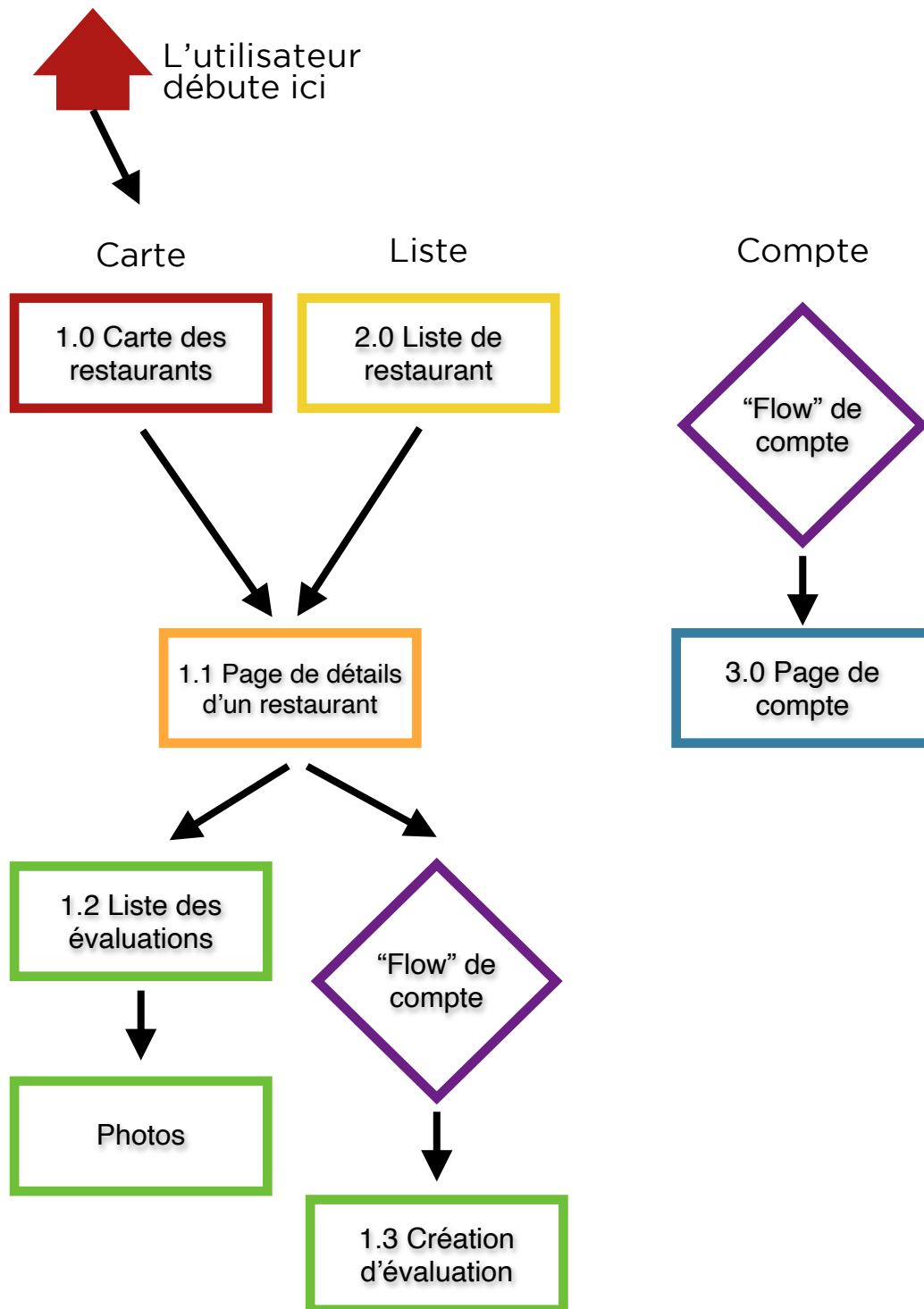
- Pour faire effet une équipe doit présenter des motifs raisonnables et des preuves de bon vouloir.
  - Il vous est donc recommandé de garder des traces de vos commits, courriels pour planifier une réunion, etc.
- Advenant un scénario où un élève est expulsé d'une équipe, il sera du devoir de l'étudiant expulsé et de l'équipe restante de remettre le même travail que convenu précédemment dans le rapport de la phase 1.
- Advenant une demande frauduleuse d'une équipe ou de présentation de preuve fabriquée une équipe pourra être dissoute et une pénalité allant jusqu'à 75% de la note finale du travail pourra être appliquée.

## Barème de correction

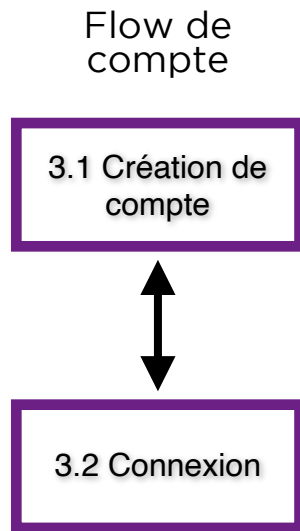
35% de la note finale est allouée pour le Mini-Projet et les rapports

	Note
Rapport 1 - Planification	/5
Rapport 2 - Avancement	/5
Rapport 3 - Bilan	/5
Qualité du code source (Nom des objets, respect des normes de programmations Kotlin, réutilisation de code)	/3
Concept de programmation	/5
- Liste intégré avec ViewHolder	/2
- GPS intégré directement	/3
Réalisation de l'application testé via APK	/42
- Carte des restaurants	/7
- Liste des restaurants à proximité	/5
- Page de détail d'un restaurant	/7
- Liste des évaluations	/4
- Création d'une évaluation	/4
- POST d'une photo	/4
- Mon Compte	/3
- Création de compte /Connexion	/3
- Mécanique de connexion/authentification	/5
Respect des consignes (signature de l'APK, format de remise)	/5
Total	/70

## Annexe A - Schéma de navigation de l'application



## Annexe B - “Flow” de connexion



## Annexe C - Exemple rapport d'analyse

Exemple de tableau de cahier de produit avec temps estimé.  
(*Temps fonctionnalité*)

Tâche	Description	Temps estimé
Connexion	Intégration de la page de connexion pour permettre la connexion avec un nom d'utilisateur et mot de passe.	15h
...	...	...
TOTAL		25h

Exemple de tableau résumé de temps estimé par coéquipier.  
(*Temps coéquipier*)

Nom	Temps estimé
Mathieu Juneau	15h
...	...
TOTAL	25h

## Annexe D - Exemple Rapport d'avancement

Exemple de tableau de cahier de produit avec estimation initiale, état courant et réestimation. (*Temps fonctionnalité*)

Tâche	Description	État	Temps estimé	Temps réel	Temps ré-estimé
Connexion	Intégration de la page de connexion pour permettre la connexion avec un nom d'utilisateur et mot de passe.	Débuté	15h	5h	25h
Reset password	...	À faire / Débuté/ Terminé	...	...	
...	...	...	...	...	
TOTAL			25h	10h	50h

Exemple de tableau résumé par coéquipier qui présente l'estimation initiale, le temps réel passé et la réestimation. (*Temps coéquipier*)

Nom	Temps estimé	Temps réel	Temps ré-estimé
Mathieu Juneau	15h	5h	25h
...	...		
TOTAL	25h	10h	50h

## Annexe E - Exemple Rapport de bilan

Exemple de tableau de cahier de produit avec estimation initiale, réestimation de mi-projet et temps final. (*Temps fonctionnalité*)

Tâche	Description	Temps estimé	Temps ré-estimé	Temps final
Connexion	Intégration de la page de connexion pour permettre la connexion avec un nom d'utilisateur et mot de passe.	15h	25h	5h
Reset password	...	...		...
...	...	...		...
TOTAL		25h	50h	10h

Exemple de tableau résumé par coéquipier qui présente l'estimation initiale, la réestimation de mi-projet et le temps final. (*Temps coéquipier*)

Nom	Temps estimé	Temps ré-estimé	Temps final
Mathieu Juneau	15h	25h	5h
...	...		
TOTAL	25h	50h	10h