

Kata LBCFLBA «Les Bons Comptes Font Les Bons Amis»

Table of Contents

- À propos
- Vue d'ensemble
- Informations générales
- Matériel et logiciel à disposition
- Descriptif du projet
 - Technologies
 - Maquettes
 - Exemple de cas d'emploi
 - Fonctionnement général
 - Écrans de l'application
 - * Écran d'accueil
 - * Écran de création d'ardoise
 - * Écran de saisie et consultation
 - * Écran de rapport
 - * Écran d'aide
 - * Écran à propos
- Livrables
 - Planification
 - Code
 - Dossier de projet
 - * Rapport
 - * Annexes
 - Journal de travail
 - Application et code
- Points techniques évalués spécifiques au projet
- Conseils
 - Conseils sur le déroulé de la documentation
- Acceptation du cahier des charges

À propos

«Les Bons Comptes Font Les Bons Amis» est un exercice de programmation destiné aux apprenti·e·s [informaticien·ne·s CFC en voie développement d'applications](#). Il est fait pour se dérouler sous forme de [Travail pratique individuel \(TPI\)](#), dont le cadre est fixé par l'article 20 de l'[Ordonnance du SEFRI sur la formation professionnelle initiale](#) et l'évaluation faite selon les [critères d'évaluation ICT](#), détaillés dans le [document fourni par iCQ-VD](#).

Les sources de ce document se trouvent sur <https://github.com/ponsfrilus/kata-lbcflba>.

Contributions, remarques et commentaires bienvenues via <https://github.com/ponsfrilus/kata-lbcflba/issues>.

Vue d'ensemble

Le but de ce travail est de développer une application permettant de gérer les dépenses dans un groupe d'amis, par exemple pour une soirée, un voyage ou une sortie. En créant une "ardoise" virtuelle, les utilisateurs peuvent entrer leurs frais respectifs et le système permet de facilement résumer le remboursement d'un utilisateur aux autres.

Le projet est nommé «Les Bons Comptes Font Les Bons Amis», abrégé *LBCFLBA* pour la suite.

Informations générales

Le Kata LBCFLBA est prévu pour être réalisé en une semaine (40 heures), sous la forme d'un mini-TPI ou TPI blanc. Il est néanmoins recommandé d'être flexible et d'adapter, en fonction de la personne, la durée prévue pour la réalisation.

La répartition du temps suggérée est la suivante :

	Horaire	Heures	Jours
Analyse	10%	04:00	$\frac{1}{2}$
Implémentation	40%	20:00	$2\frac{1}{2}$
Tests	10%	04:00	$\frac{1}{2}$
Documentation	30%	12:00	$1\frac{1}{2}$
<i>Total</i>	<i>100%</i>	40:00	5

L'horaire est de 8 heures par jour :

	Horaire	Pause	Durée
Matin	08:40 - 13:00	20 minutes	4h00
Après-midi	14:00 - 18:15	15 minutes	4h00
Total			8h00

Matériel et logiciel à disposition

La réalisation de ce travail nécessite uniquement un laptop et un accès à Internet. L'utilisation de logiciels libres est fortement recommandée.

Descriptif du projet

Le but de ce projet est de développer l'application LBCFLBA, une ardoise virtuelle permettant de facilement tracer et répartir les coûts entre les participants d'un événement.

Technologies

Ce cahier des charges ne précise aucune technologie pour la réalisation de cette application. Il est attendu de la part du·de la candidat·e une réflexion sur la meilleure manière de livrer une telle application dans le temps imparti. Il est donc présumé ici que le candidat choisira une solution dans sa zone de confort et fournira, à l'issue du mini-TPI, un site Internet ou une application mobile, voire les deux en même temps.

Maquettes

Les maquettes des écrans proposées dans ce document sont présentées uniquement pour illustrer une possible réalisation du projet. Elles permettent de se mettre dans la peau des utilisateurs·trices et de s'imaginer une version de la solution. Elles ne font pas office de référence pour la création de l'application.

Exemple de cas d'emploi

Alice, Bob, Carole et David ont prévu une sortie au Montreux Jazz Festival pour fêter l'anniversaire de David. Pour cette sortie, tous les frais sont pris en charge par Alice, Bob et Carole en guise de cadeau à David.

Alice prend les billets de train Lausanne Montreux aller retour :

$$4 \times 2 \times 6.50 = 52.-$$

Bob commande les billets d'accès au festival pour la soirée :

$$4 \times 88 = 352.-$$

Il en profite pour acheter l'affiche de la 56e édition faite par [Camille Walala](#), il l'offrira à David en souvenir de la soirée :

$$69.-$$

Une fois sur place, Carole achète des crêpes et des boissons (4 crêpes salées, 2 sucrées, 4 boissons) :

$$106.-$$

Pendant le concert, Carole va au bar et ramène des bières pour tout le monde :

$$4 \times 6 = 24.-$$

Plus tard, Alice va chercher des verres d'eau gazeuse :

$$4 \times 4.5 = 18.-$$

Le montant de la soirée s'élève à 620.-. Sachant que David ne doit rien payer (les frais lui sont offerts pour son anniversaire), comment partager les frais entre Alice, Bob et Carole ?

➔ Pour effacer toutes les ardoises, Alice et Carole sont débitrices de Bob de respectivement 137.- et 77.- (Bob est créancier pour un montant de 214.-).

Fonctionnement général

La praticité de l'application est le mot d'ordre.

Il y a deux cas d'emplois :

1. L'utilisateur utilise l'application pour créer une nouvelle ardoise ;
2. L'utilisateur est invité à rejoindre une ardoise existante.

Ces deux cas d'emplois sont expliqués plus en détails dans la description des différents écrans ci-dessous.

Écrans de l'application

Écran d'accueil

L'écran d'accueil de l'application doit permettre à l'utilisateur de rapidement comprendre à quoi l'application sert, de créer une ardoise et de naviguer entre les différents écrans.

Il se compose de :

- Un titre
- Un texte sommaire (une accroche)
- Un bouton pour créer une ardoise
- Un lien vers l'aide/FAQ
- Un lien vers à propos



Figure 1: “Maquette de page d’accueil”

Écran de création d'ardoise

Lorsque l'utilisateur veut créer une nouvelle ardoise, on lui demande :

- Le titre de l'ardoise (obligatoire)
- La description de l'ardoise (facultative)
- Une date de fin (facultative)

La confirmation de la création de l'ardoise génère un hash unique permettant au créateur de l'ardoise de la partager.



Figure 2: "Maquette de page de création"

Écran de saisie et consultation

Lorsqu'un utilisateur accède à une ardoise avec un lien, il doit d'abord s'identifier : il peut soit cliquer sur le nom d'un participant existant soit créer un nouveau participant.

Il peut ensuite interagir avec tous les éléments avec l'autorité de la personne qu'il a sélectionnée :

1. Consulter les dépenses existantes
2. Ajouter une nouvelle dépense à l'ardoise
3. Modifier une dépense existante
4. Supprimer une dépense

Cet écran affiche également un classement permettant de voir qui devrait être la prochaine personne à payer une dépense (dans le but de tendre vers une équité des



Figure 3: “Maquette de page d’identification”

dépenses).

Un bouton pour accéder au détail du rapport est disponible.



Figure 4: “Maquette de page de saisie”

Écran de rapport

Cet écran propose un état des comptes et comment résoudre l’ardoise (qui doit combien à qui). Il est fait de manière intelligible* et facilement partageable.

Écran d’aide

L’écran d’aide explique comment utiliser l’application et liste les questions posées fréquemment (FAQ). Un lien vers les issues du dépôt est proposé aux utilisateurs.

Écran à propos

L’écran à propos est une impressum qui explique qui, quand, quoi, comment, affiche un numéro de version et un lien vers les sources. On y fait appel aux contributeurs.

Livrables

Planification

En fin de première journée de travail, le·la candidat·e envoie (au format [PDF](#)) aux intéressé·e·s une planification initiale détaillant les tâches à accomplir durant le projet. Le niveau de granularité du découpage devrait être de 2 à 4 heures, mais peut descendre plus bas si la durée du TPI est écourtée.

Tout au long du projet, le·la candidat·e mettra à jour la planification réelle.

En fin de projet, le·la candidat·e veillera à ajouter les planifications initiale et réelle dans son rapport, et prendra le soin d'en commenter les différences.

Code

Git !

Dossier de projet

Rapport

Un [canevas de dossier de projet](#) est à disposition du·de la candidat·e.

Le rapport prête une attention particulière à détailler tous les [points techniques évalués spécifiques au projet](#), prouvant que l'élément a été traité de manière professionnelle par le·la candidat·e.

Les termes techniques et les acronymes utilisés dans le rapport sont référencés dans un glossaire figurant dans le rapport.

Les choix technologiques sont justifiés dans le rapport. Les outils et les technologies utilisées sont l'objet de descriptions explicatives dans le rapport.

Le candidat démontre sa compréhension du système en fournissant un schéma d'architecture dont la description détaille l'interaction entre les différents systèmes.

Le document doit évoluer chaque jour. Il sera envoyé dans l'état aux intéressé·e·s deux fois par semaine, au format [PDF](#). Dans le cas d'un mini-TPI, le rapport est envoyé chaque jour.

Annexes

Le rapport contient tous les documents nécessaires à la compréhension du déroulement du projet en annexes. Cahier des charges, planifications, journal de travail, résumé du rapport TPI, etc. doivent être annexés au document.

Journal de travail

Le journal de travail doit permettre de retracer les activités du·de la candidat·e tout au long du déroulement du projet. Durée des tâches, PV des discussions, problèmes rencontrés, choix, solutions, liens vers la documentation, les références, sources d'informations, aide extérieure, heures supplémentaires, etc. doivent être consignés dans ce document (c.f. [critères d'évaluation B2](#)).

Le journal de travail est présent dans le dossier de projet, en annexe au rapport.

Le document doit évoluer chaque jour. Il sera envoyé dans l'état aux intéressé·e·s deux fois par semaine, au format [PDF](#). Dans le cas d'un mini-TPI, le journal de travail est envoyé chaque jour.

Application et code

Le·la candidat·e communique l'adresse de son dépôt Git aux intéressé·e·s et le maintient à jour quotidiennement (plusieurs *commits* par jour). Le dépôt est agrémenté d'un fichier `README.md` au format [Markdown](#), qui explique l'utilisation du projet et sa mise en œuvre. (Voir aussi l'objectif «simplicité des instructions de mise en œuvre», ci-dessous). Le lien vers le dépôt est présent dans la documentation.

Points techniques évalués spécifiques au projet

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat·e sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (correspondant aux [critères d'évaluation A14 à A20](#)) :

1. La qualité du repository [Git](#) : messages de commits explicites et lisibles, permettant de retracer l'évolution du code (plusieurs commits par jour, création de branches de fonctionnalités), fichier `README.md` présentant le projet et son déploiement.
2. Un code exempt de sections copiées/modifiées (principe [DRY: Don't Repeat Yourself](#)) et respectant le [style de programmation](#) des langages utilisés.
3. La simplicité des instructions de mise en œuvre, qui permettent aux intéressé·e·s d'essayer le projet sur leur propre équipement au fur et à mesure de sa progression. Idéalement, les instructions se limitent à deux étapes (`git clone` et `docker-compose up`).
4. Les différentes [méthodes HTTP](#) sont implémentées à bon escient en fonction de l'action réalisée sur la ressource indiquée. Les [codes de réponse HTTP](#) utilisés permettent aux clients d'avoir une information sur le résultat de leurs requêtes.
5. Le front-end est soigné, la liste des mangas paginée, triable et la possibilité de faire une recherche dans la table est présente.
6. Le rapport démontre que le·la candidat·e a étudié le modèle des données : un diagramme entité-association ([ERD](#)) est présent dans le rapport. Le·la candidat·e décrit et critique le diagramme et les différentes tables.
7. L'utilisateur·trice a accès à une page de documentation de l'API, qui explique les types de données, les valeurs de retour, les différentes possibilités d'interactions avec l'API. Le respect de [OAS](#) et l'utilisation des fonctions de documentation de [Swagger](#) sont nécessaires pour obtenir le score maximal sur ce point.

Conseils

Conseils sur le déroulé de la documentation

Le rapport, le journal de travail et le dépôt Git doivent être mis à jour en continu pour rendre compte des accomplissements à chaque étape ci-dessus.

Aucune «hypothèque» de temps de travail péjorant la documentation ni le code, ne seront tolérées.

△ Il est impératif de souligner l'importance de la documentation dans cet exercice, c'est principalement sur cette dernière que les candidat·e·s sont évalués △

Acceptation du cahier des charges

	Lu et approuvé le	Signature
Candidat·e	_____	_____
Expert·e n°1	_____	_____
Expert·e n°2	_____	_____
Chef·fe de projet	_____	_____