

Remise 2

Équipe no 2

GAUDREAU Louis
CHAAR Jacob
BERNIER Alexandre

Bases de données avancées
GLO 4035

Travail présenté à
Jean-Thomas Baillargeon

Département d'informatique et de génie logiciel

Université Laval 2018

La description du problème et des données

Pour ce travail, nous avons choisi de développer le projet d'inventaire de savons. Notre mandat fut de créer une application ainsi qu'une base de données permettant de présenter l'état de l'inventaire d'une compagnie vendant des savons. Ce document permet de présenter des aspects techniques de ce travail.

Lorsque nous avons reçu les données, nous avons remarqué que l'on peut créer trois collections avec celles-ci. Nous avons donc distribué les données selon leurs collections respectives en les comparant grâce à des schémas de données. De plus, nous avons formaté les données afin qu'elles respectent leurs types de données respectives (ex. les totaux en float). Ces changements sont effectués dans le fichier utilities.py.

Nous avons ensuite créé une API permettant de retourner une interface HTML à l'utilisateur afin d'utiliser les différentes requêtes créées. Une section expliquant les différentes routes et ce qu'elles utilisent se retrouvent plus loin dans le rapport. Toutes les routes créées sont dans le fichier app.py.

Pour faire fonctionner l'application

```
docker pull grizzlys54/glo4035a18-projet2018-eq2
docker run --name testtp -p 80:80 grizzlys54/glo4035a18-projet2018-eq2
```

Note : Ce travail a été développé et testé pour une utilisation sur Google Chrome. Pour le bon fonctionnement de cette application, les « cookies » doivent être activés et aucun bloqueur de publicité ne doit fonctionner lors de l'utilisation de l'application.

Collections

- Purchases : contiens les achats
- Densities : contiens les densités des items
- Labors : contiens les utilisations de matière première des items

Requêtes

Pour chaque aspect de ce travail, nous avons préféré être larges avec le rendu des requêtes MongoDB afin d'exploiter la puissance du langage JavaScript et ainsi spécifier les requêtes avec celui-ci. Cela fait aussi en sorte que beaucoup de requêtes sont réutilisables. Toutes ces requêtes sont effectuées à travers les routes de l'API dans app.py.

Note : ces requêtes ont été faites en python. Elles sont traduites en mongodb et peuvent contenir des erreurs de traductions.

1. Purchases .find({})
2. Densities .find({})
3. Labors .find({})

4. Purchases.remove({})
5. Densities.remove({})
6. Labors.remove({})
7. Purchases.insert({"date": date d'achat, "item": item, "qte": quantité utilisée, "unit": unité utilisé, "total": total de l'achat, "stotal": sous-total, "tax": taxe})
8. Densities.insert({"item": item, "ml": millilitre, "g": gramme})
9. Labors.insert({"date": date, "item": item, "qte": quantité, "unit": unité, "type": type de matière, "jobld": identifiant du «labor»})
10. Purchases.update({"_id": ObjectId(id)}, {"date": date, "item": item, "qte": quantité utilisée, "unit": unité, "total": total, "stotal": sous-total, "tax": taxe})
11. Densities.update({"_id": ObjectId(id)}, {"item": item, "ml": millilitre, "g": gramme})
12. Labors.update({"_id": ObjectId(id)}, {"date": date, "item": item, "qte": quantité utilisé, "unit": unité utilisé, "type": type de matière, "jobld": identifiant du «labor»})
13. Purchases.find({"\$and": [{"date": date voulue}, {"item": {"\$regex": "/type de matière voulue /"} }]})
14. Labors.find({"\$and": [{"date": {"\$lte": reqData["date"]}], {"type": "usage"}]})
15. Purchases.find({"date": {"\$lte": reqData["date"]}})

Index utilisés

- Purchases :
 - Date : Les recherches de coûts totaux, de coût d'acquisition et de coût moyen se font par la date. Il est donc utile de mettre un index pour ce champ.
- Labors :
 - Date : Les recherches de coûts totaux, de coût d'acquisition et de coût moyen se font par la date. Il est donc utile de mettre un index pour ce champ.
- Density :
 - Item : nous l'avons mis unique afin qu'il n'y ait qu'une densité par item. Nous le vérifions aussi lors de l'ajout par JavaScript.

Sécurité

Mot de passe administrateur pour la suppression de toutes les données : admin123

API

Toutes les routes créées sont dans le fichier app.py.

1. Route : request Get \--url
 - "adresse ip application'/"
 - Paramètres : aucun
 - Requête utilisée : aucune
 - Retourne l'accueil de l'application.
2. Route : request GET \--url
 - "adresse ip application'/transactions'
 - Paramètres : aucun
 - Requête utilisée : aucune
 - Retourne l'interface pour afficher les données de toutes les collections.
3. Route : request POST \--url
 - "adresse ip application'/transactions'
 - Paramètres : {donnée csv}
 - Requêtes utilisées : 7,8,9
 - Vérifie les données et les insères dans la base de données selon les données établies. Retourne une erreur 400.
4. Route : request DELETE \--url
 - "adresse ip application''/transactions'
 - Paramètres : {mot de passe administrateur}
 - Requêtes utilisées : 4,5,6
 - Supprime l'entièreté de la base de données si le mot de passe est correct, sinon retourne une erreur 401.
5. Route : request Get \--url
 - "adresse ip application'/transactions/purchases
 - Paramètres : aucun
 - Requête utilisée : 1
 - Retourne le contenu de la collection purchases.
6. Route : request Get \--url
 - "adresse ip application'/transactions/densities
 - Paramètres : aucun
 - Requêtes utilisées : 2
 - Retourne le contenu de la collection densities.
7. Route : request Get \--url
 - "adresse ip application'/transactions/labors
 - Paramètres : aucun
 - Requête utilisée : 3
 - Retourne le contenu de la collection labors.

8. Route : request Get \--url
 - ["adresse ip application'/transactions/add](#)
 - Paramètres : aucun
 - Requête utilisée : aucune
 - Retourne l'interface pour ajouter un item dans une collection.
9. Route : request Post \--url
 - ["adresse ip application'/transactions/add/purchase](#)
 - Paramètres: {"date": date, "item": item, "qte": quantité utilisé, "unit": unité, "total": total, "stotal": sous-total, "tax": taxe}
 - Requête utilisée : 7
 - Envoi un item dans la collection purchases. Retourne un Json de succès avec un statut 200 ou 400.
10. Route : request Post \--url
 - ["adresse ip application'/transactions/add/density](#)
 - Paramètres: {"item": item, "ml": millilitre utilisé, "g": gramme utilisé}
 - Requête utilisée : 8
 - Envoie un item dans la collection density. Retourne un Json de succès avec un statut 200 ou 400.
11. Route : request Post \--url
 - ["adresse ip application'/transactions/add/labor](#)
 - Paramètres: {"date": date, "item": item, "qte": quantité utilisée, "unit": unité utilisée, "type": type de matière, "jobId": identifiant du «labor»}
 - Requête utilisée : 9
 - Envoie un item dans la collection labors. Retourne un Json de succès avec un statut 200 ou 400.
12. Route : request Get \--url
 - ["adresse ip application'/transactions/modifyPurchase](#)
 - Paramètres : aucun
 - Requête utilisée : aucune
 - Retourne l'interface de modification des achats.
13. Route : request Post \--url
 - ["adresse ip application'/transactions/modifyPurchase](#)
 - Paramètres : {"id": id, "date": date, "item": item, "qte": quantité utilisé, "unit": unité utilisé, "total": total de l'achat, "stotal": sous-total de l'achat, "tax": taxe}
 - Requête utilisée : 10
 - Modifie un item dans la collection purchases. Retourne un Json de succès avec un statut 200 ou 400.
14. Route : request Get \--url
 - ["adresse ip application'/transactions/modifyDensity](#)

- Paramètres : aucun
 - Requête utilisée : aucune
 - Retourne l'interface de modification des densités.
15. Route : request Post \--url
- ["adresse ip application'/transactions/modifyDensity](#)
 - Paramètres: {"id":id, "item": item, "ml": millilitre utilisé, "g": gramme utilisé}
 - Requête utilisée : 11
 - Modifie un item dans la collection densitiesDb. Retourne un Json de succès avec un statut 200 ou 400.
16. Route : request Get \--url
- ["adresse ip application'/transactions/modifyLabors](#)
 - Paramètres : aucun
 - Requête utilisée : aucune
 - Retourne l'interface de modification des travaux effectués sur les savons.
17. Route : request Post \--url
- ["adresse ip application'/transactions/modifyLabors](#)
 - Paramètres: {"id": id, "date": date, "item": item, "qte": quantité utilisée, "unit": unité utilisée, "type": type de matière, "jobId": identifiant du «labor»}
 - Requête utilisée : 12
 - Modifie un item dans la collection densities. Retourne un Json de succès avec un statut 200 ou 400.
18. Route : request Get \--url
- ["adresse ip application'/totalCost/](#)
 - Paramètres : aucun
 - Requête utilisée : aucune
 - Retourne l'interface d'affichage pour le coût total à une date précise pour une catégorie de matériel.
19. Route : request Post \--url
- ["adresse ip application'/totalCost/](#)
 - Paramètres: {"date":date, "type": type de matière}
 - Requête utilisée : 13
 - Retourne le coût total à une date précise pour une catégorie de matériel.
20. Route : request Get \--url
- ["adresse ip application'/averageCost/](#)
 - Paramètres : aucun
 - Requête utilisée : aucune
 - Retourne l'interface d'affichage pour le coût moyen à une date précise pour une catégorie de matériel

21. Route : request Get \--url
- "adresse ip application'/leftQuantity/
 - Paramètres : aucun
 - Requête utilisée : aucune
 - Retourne l'interface d'affichage pour les quantités restante d'un item à une date précise pour une catégorie de matériel.
22. Route : request Post \--url
- "adresse ip application'/leftQuantity /
 - Paramètres: {"date":date}
 - Requête utilisée : 14
 - Retourne les «labors» de type «usage» à la date demandé
23. Route : request Post \--url
- "adresse ip application'/leftQuantity /purchase
 - Paramètres: {"date":date}
 - Requête utilisée : 15
 - Retourne les achats de la date demandée.

Interfaces

Chaque interface de l'application fut créée en HTML5 et JavaScript selon la norme ES6.

Accueil



L'accueil de l'application. Permet de naviguer vers les autres pages.

Listes des transactions (Inventory)

Home

Inventory

Total cost

Average Acquisition cost

Left quantity

Transactions

Add item

Clear database

Purchases - Densities - Labors

Purchases

Id		Item	Unit	Quantity	Date	Subtotal	Tax	Total	Modification	Remove
5bfc15f44a0eb40001817a99		Consumable - HE - Pin Blanc	ml	123	2018-11-23	123 \$	2 \$	125 \$	<div>Modify</div>	<div>Remove</div>

Density

Id		Item	g	ml	Modification	Remove
5bfc24270b62120001d8601b		Consumable - Base Oil - Jojoba Oil	0.868	1	<div>Modify</div>	<div>Remove</div>
5bfc24270b62120001d86021		Consumable - Base Oil - Sweet Almond Oil	0.91	1	<div>Modify</div>	<div>Remove</div>
5bfc24270b62120001d86029		Consumable - Base Oil - Coconut Oil	0.903	1	<div>Modify</div>	<div>Remove</div>
5bfc24270b62120001d8602a		Consumable - Base Oil - Avocado Oil	0.92	1	<div>Modify</div>	<div>Remove</div>
5bfc24270b62120001d8602e		Consumable - Base Oil	0.9	1	<div>Modify</div>	<div>Remove</div>
5bfc24270b62120001d86030		Consumable - Base Oil - Olive Oil	0.916	1	<div>Modify</div>	<div>Remove</div>
5bfc24270b62120001d86036		Consumable - Base Oil - Argan Oil	0.912	1	<div>Modify</div>	<div>Remove</div>
5bfc24280b62120001d86039		Consumable - Base Oil - Castor Oil	0.961	1	<div>Modify</div>	<div>Remove</div>
5bfc4539cc0080001e49ec2		Consumable - HE - Tea Tree	0.9876	5	<div>Modify</div>	<div>Remove</div>
5bfc130f79c7d2000141eeec2		Consumable - HE - Pin Blanc	123	0.1	<div>Modify</div>	<div>Remove</div>

Labors

Id		Job Id	Item	Date	Unit	Quantity	Type	Modification	Remove
5bfc24270b62120001d86019		4	Consumable - Base Oil - Coconut Oil	2018-08-07	g	30	usage	<div>Modify</div>	<div>Remove</div>
5bfc24270b62120001d8601a		3	Consumable - HE - Bergamote	2018-03-10	ml	10	usage	<div>Modify</div>	<div>Remove</div>
5bfc24270b62120001d8601d		3	Consumable - Soude Cautique	2018-03-10	g	110	usage	<div>Modify</div>	<div>Remove</div>
5bfc24270b62120001d8601e		1	Consumable - HE - Eucalyptus Radiata	2018-01-03	ml	5	usage	<div>Modify</div>	<div>Remove</div>
5bfc24270b62120001d8601f		4	Consumable - Additive - Argile Jaune	2018-08-07	g	10	usage	<div>Modify</div>	<div>Remove</div>

Permet de lister les données de toutes les collections. En cliquant sur «Add item», l'utilisateur sera transféré vers la page d'ajout d'un item. En cliquant sur «Clear database», l'utilisateur sera invité à entrer un mot de passe permettant de supprimer les données de toutes les collections. En cliquant sur «Modify», l'utilisateur sera transféré vers la page de modification de l'item associé au bouton. En cliquant sur «Remove», l'utilisateur peut supprimer l'item voulu après confirmation. Les «purchases» et les «labors» sont classés par date d'ajout et les «densities» par nom d'items.

Modification d'un item

Home	Inventory	Total cost	Average Acquisition cost	Left quantity
------	-----------	------------	--------------------------	---------------

Modify purchase

ID:

Item : Consumable - -

Unit:

Qty:

Date:

Subtotal:

Tax:

Total:

[Modify](#)

Cette interface permet de modifier un item dans la collection choisie dans l'interface de la liste des transactions. Les données de l'item seront automatiquement affichées. Après avoir modifié les données, l'utilisateur clique alors sur le bouton «Modify» et les données s'envoient au serveur. Une alerte informera ensuite à l'utilisateur si la modification a bien été faite.

Ajout d'un item

Home

Inventory

Total cost

Average Acquisition cost

Past images

Add item

Item type: Purchase ▾

Item : Consumable - HE ▾ ▾

Unit: ml ▾

Qty:

Date:

Subtotal:

Tax:

Total:

Cette interface permet d'ajouter un item dans n'importe laquelle des collections (purchase, labor ou density). Il faut simplement choisir le type d'item voulu et les informations vont s'ajuster selon le type. Lorsque l'utilisateur clique sur le bouton «Add purchase», les données s'envoient au serveur. Une alerte informera ensuite à l'utilisateur si l'ajout a bien été fait.

Coût total

Home

Inventory

Total cost

Average Acquisition cost

Left quantity

Total cost

Select the date to display the total cost:

Date:

2018-11-01	
Type	Total cost
Consumable - Emballage	70 \$
Consumable - HE - Apple	60 \$
Consumable - HE - Pin Blanc	50 \$
Consumable - HE	110 \$
Consumable	180 \$

Cette interface permet d'afficher le coût total selon une date précise pour une catégorie de matériel. L'utilisateur doit entrer les informations voulues et cliquer sur le bouton «Get total». Un tableau s'affichera avec les coûts totaux. S'il n'y a pas de totaux pour cette date ou catégorie, une mention l'affichera.

Coût moyen d'acquisition

Home	Inventory	Total cost	Average Acquisition cost	Left quantity
------	-----------	------------	--------------------------	---------------

Average cost

Select the date, unit and the consumable category:

Date:

Unit:

Item:

2018-12-12				
Type	Average value per acquisition unit (in ml)	Average value per acquisition unit (in L)	Average value per unit of use (per g)	Average value per unit of use (per 100g)
Consumable - HE - Apple	0.250000 \$	250.00 \$		
Consumable - HE - Bergamote	0.294300 \$	294.30 \$	0.294300 \$	29.43 \$
Consumable - HE - Frangipani	0.237500 \$	237.50 \$		
Consumable - HE - Lemongrass	0.220000 \$	220.00 \$		
Consumable - HE - Lime	0.413667 \$	413.67 \$		
Consumable - HE - Menthe Poivree	0.527600 \$	527.60 \$		
Consumable - HE - Orange Amere	0.755333 \$	755.33 \$		
Consumable - HE - Palmarosa	0.390000 \$	390.00 \$		
Consumable - HE - Patchouli	0.505700 \$	505.70 \$	0.505700 \$	50.57 \$
Consumable - HE - Peppermint	0.329900 \$	329.90 \$		
Consumable - HE - Rosemary	0.428333 \$	428.33 \$		
Consumable - HE - Ylang Ylang	0.237500 \$	237.50 \$		

Cette interface permet d'afficher le coût d'acquisition des items selon une date précise pour une catégorie de matériel et d'unité. L'utilisateur doit entrer les informations voulues et cliquer sur le bouton «Get average cost ». Un tableau s'affichera avec les coûts moyens selon différentes unités de mesure. Si un item n'est pas lié à une unité de mesure, le coût moyen associé à cette mesure ne s'affichera pas dans le tableau. S'il n'y a pas de densité pour une catégorie trouvée, le facteur sera alors de 1 g/ml par défaut.

Quantité restante

Home	Inventory	Total cost	Average Acquisition cost	Left quantity
------	-----------	------------	--------------------------	---------------

Quantity left

Select the date to display the quantity left:

Date:

N.B. If there's no density factor for an item, the g/ml factor will be 1

2018-11-01		
Consumable	Quantity left (ml)	Quantity left (g)
Consumable - HE - Pin Blanc	123	123
Consumable - HE - Apple	123	123
Consumable - Emballage	123	123

Cette interface permet d'afficher la quantité restante d'items selon une date précise pour une catégorie de matériel. L'utilisateur doit entrer les informations voulues et cliquer sur le bouton «Get left quantity». Un tableau s'affichera avec les quantités restantes en millilitres et en gramme selon le facteur g/ml trouvé dans la base de données. S'il n'y a pas de densité pour une catégorie trouvée, le facteur sera alors de 1 g/ml par défaut. S'il n'y a pas de quantité restante lors de cette date, le tableau sera vide.