

Documentation d'utilisation Programmation

TORIBIO Alexis

LOPEZ Tao

MANANT Clément

16/06/2024

B3 Robotique

Sommaire :

Sommaire :	2
Introduction.....	3
Outils nécessaires	4
Matériel physique.....	4
Logiciels.....	5
Web	5
Programmation.....	5
Récupération des éléments	6
Récupération du git	6
Récupération des dépendances.....	6
Mise en place des éléments	7
Lancement de l'application	7
Vérification du fonctionnement.....	7
Intégration du code sur la Raspberry.....	8
Utilisation du code côté client/admin	9
Fonctionnement côté client.....	9
Fonctionnement côté admin	10
Sources.....	14

Introduction

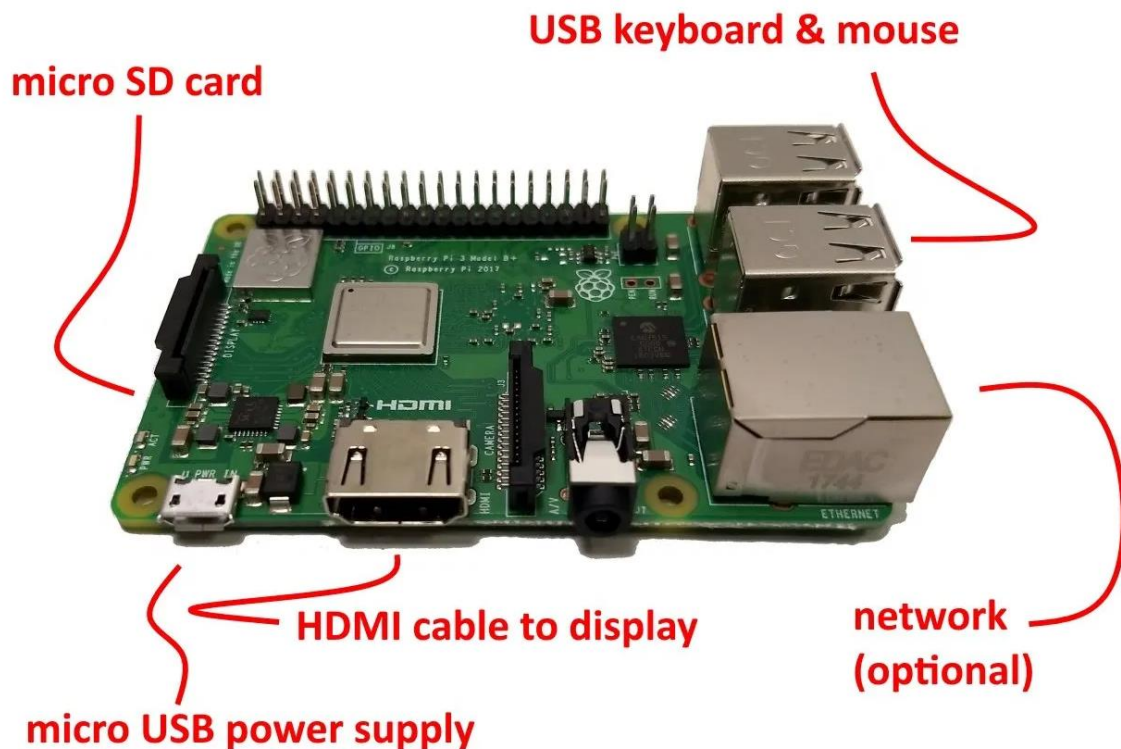
La partie informatique est très importante pour ce projet, que ce soit pour l'application web ou pour la programmation du robot. Il est nécessaire pour le projet que l'application soit intuitive et ergonomique, l'utilisateur doit pouvoir commander sa boisson rapidement. Le robot doit de son côté être fluide et précis dans ses mouvements pour ne pas renverser de liquide à des endroits imprévus.

Outils nécessaires

Matériel physique

Pour l'application web, seulement un ordinateur est nécessaire pour son fonctionnement. Cependant, pour la programmation du robot, beaucoup de matériel physique est nécessaire. Pour avoir un ordinateur utilisable avec la Raspberry, il faudra les éléments suivants :

- 1 Raspberry Pi 3B+
- 1 carte SD 16 Gb minimum flashée à l'aide du logiciel [Raspberry Pi Imager](#)
- 1 câble d'alimentation pour la Raspberry
- 1 souris
- 1 clavier
- 1 écran
- 1 câble HDMI



Vous pouvez vous aider de cette documentation pour plus de précisions sur le branchement de la Raspberry : [Branchement Raspberry Pi 3 B+](#)

Une fois le branchement de l'alimentation à la Raspberry, si la carte SD est récemment flashé l'ordinateur démarre et peut mettre un peu de temps avant d'être fonctionnel. À partir de là tous les logiciels sont déjà installés, il ne reste plus qu'à rajouter les librairies qui permettront de faire fonctionner le projet.

Logiciels

Web

L'élément principal pour l'application est [Visual Studio Code](#), un éditeur de code polyvalent qui va nous être utile notamment pour ses extensions. L'intégralité de l'application est en JS un langage très utilisé dans le web pour sa rapidité d'exécution ainsi que les nombreuses dépendances qui permettent une grande variété de projets.

Le framework [NodeJS](#) est utilisé pour le backend, grâce à sa grande communauté, sa performance est optimisée et il est très efficace de développer avec, à l'aide des nombreux tutoriels et documentations mises à jour régulièrement. Pour avoir une API propre et fonctionnelle, il est important de pouvoir la tester. Pour cela, l'extension VS Code [ThunderClient](#) permet de tester son API directement à travers VS Code, sans avoir à créer de compte.

Pour le frontend, le framework [VueJS](#) se montrait le plus pertinent grâce à sa facilité d'intégration à l'API déjà faite en NodeJS, mais aussi grâce à sa légèreté d'une vingtaine de kilo octets, qui ne lui enlève pas pour autant sa vitesse d'exécution.

Ainsi, pour les extensions importantes au projet, nous retrouvons ThunderClient, NodeJS et VueJS, en plus des extensions au choix de l'utilisateur qui permettent d'augmenter et de simplifier la production de lignes de code comme LiveServer par exemple qui permet un travail simultané sur le projet ou divers linters pour embellir le code.

Programmation

Du côté de la Raspberry, le logiciel installé par défaut [Thonny](#) fait parfaitement l'affaire pour écrire du code Python à exécuter sur la carte. Il faudra cependant rajouter quelques librairies pour le fonctionnement des moteurs et d'autres capteurs/actionneurs du montage électronique.

Pour cela, vous pouvez clone le git [RpiMotorLib](#) avec la commande : 'git clone <https://github.com/gavinlyonsrepo/RpiMotorLib.git>' et installer la librairie [RPi.GPIO](#) avec la commande 'pip install RPi.GPIO'. Il est évidemment plus pratique de le faire directement via un terminal sur la Raspberry, mais si vous n'avez pas internet sur la Raspberry, vous pouvez copier les dossiers à l'aide d'une clé USB dans le dossier /usr/local/lib/python3.9/dist-packages/ .

Récupération des éléments

Une fois l'environnement de travail installé proprement il faut récupérer les différents fichiers de code.

Récupération du git

Afin de récupérer le git qui répertorie les fichiers relatifs au projet, créez un dossier dans votre ordinateur qui sera utilisé pour le projet. Ouvrez ensuite un terminal dans ce dossier et écrivez : 'git clone <https://github.com/clementmanant/Robarathon.git>'

Si vous ne possédez pas déjà [git](#), vous pouvez l'installer et réessayer.

Pour faire fonctionner l'API, un fichier .env doit être rajouté à la racine du dossier api. Pour des questions de sécurité, il n'est pas disponible sur le git mais vous pouvez le demander en contact privé.

Récupération des dépendances

Avec un terminal, rendez-vous dans le dossier que vous venez de copier. Ensuite allez dans le dossier de l'API en faisant 'cd api/'. Installez les dépendances nécessaires grâce à 'npm install'. Faites de même côté app :

'cd ../' pour revenir dans le dossier précédent.

'cd app/' pour se rendre dans le dossier app.

'npm install' pour installer les dépendances nécessaires au fonctionnement de l'application.

Mise en place des éléments

Lancement de l'application

Pour lancer l'application, rendez-vous dans un terminal du dossier Robarathon cloné précédemment. Allez dans un premier temps du côté de l'API avec un 'cd api/'. Faites 'npm run dev' pour lancer le serveur API. Vous pouvez ensuite vous rendre dans le dossier de l'application pour la lancer aussi :

'cd ../app/' pour se rendre dans le dossier app.

'npm run serve' pour le lancement du serveur de l'application.

Vérification du fonctionnement

Pour vérifier que tout est en ordre, vous pouvez d'abord regarder si vous avez accès à la base de stockage des boissons. Pour cela ouvrez une nouvelle page dans un navigateur et copier '<http://localhost:3000/api/products/>'. Vous devriez obtenir un long texte ayant un format semblable à celui-ci :

```
[{"_id":"65ef8f8114fb417ef11d6055","name":"Mojito","quantity":100,"price":8,"createdAt":"2024-03-11T23:10:57.165Z","updatedAt":"2024-06-12T15:10:35.552Z","__v":0,"steps":[{"ingredient_id":"65f9a35238c2a5c39894c29e","quantity":5,"ordre":1,"_id":"6602a02e842125ffa6bdf97"},{"ingredient_id":"66029a2b842125ffa6bdf7c","quantity":2,"ordre":2,"_id":"6602a02e842125ffa6bdf98"},{"ingredient_id":"66029a53842125ffa6bdf7f","quantity":2,"ordre":3,"_id":"6602a02e842125ffa6bdf99"}],"available":true,"imageUrl":"https://drinkch.rokka.io/new-composition-1200/22dafc.jpg"},
```

...

```
{"_id":"6627714e878dc9f2bf17b1bb","name":"Vodka Redbull","quantity":50,"price":5,"available":true,"steps":[{"_id":"66730dbc721b261eaeabd1ec","ingredient_id":"660292e0842125ffa6bdf48","ordre":1,"quantity":4},{"_id":"66730dbc721b261eaeabd1ed","ingredient_id":"66276ae3c47742f2c83b7b27","ordre":2,"quantity":8}], "imageUrl":"https://drinksguide.dk/wp-content/uploads/2022/10/Vodka-Red-Bull.jpg"}]
```

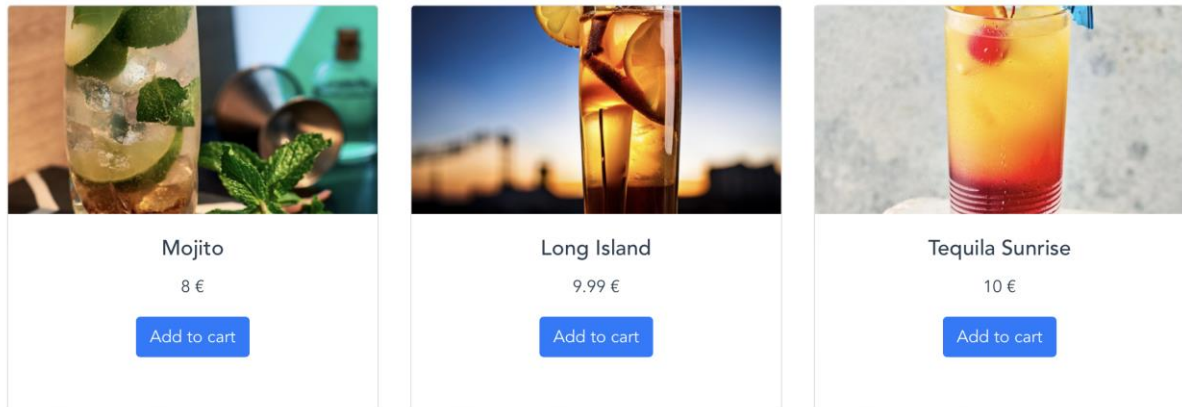
Vous pouvez ensuite tester si l'autre base de données connectée à l'API fonctionne en copiant ce lien dans une page de navigateur : '<http://localhost:3000/api/redis/list>'. Si c'est votre première utilisation, vous êtes censés voir [] sur votre page.

Si ces deux liens fonctionnent correctement, la partie API marche et on peut vérifier la partie application en copiant ce lien dans une page de navigateur : '<http://localhost:8080/>'.

Si vous avez accès à la page d'accueil, c'est parfait.

[Home](#) | [Admin](#) | [Cart](#)

Products



Mais on peut aller un peu plus loin pour vérifier que l'interaction entre l'application et l'API se fait bien. Ajoutez un cocktail à votre panier, rendez-vous dans votre panier et validez-le. Vous pouvez ensuite retourner sur le lien '<http://localhost:3000/api/redis/list>' qui est censé afficher un résultat semblable à celui-ci :

```
[{"user_id":"userId","cocktail":{"_id":"65ef9c5e87c4d52f67705294","name":"Tequila Sunrise","quantity":30,"price":10,"createdAt":"2024-03-12T00:05:50.947Z","updatedAt":"2024-03-26T10:28:03.916Z","__v":0,"steps":[{"ingredient_id":"66029a95842125ffa6bdf83","quantity":5,"ordre":1,"_id":"6602a333842125ffa6bdfbf"}, {"ingredient_id":"66029ad7842125ffa6bdf87","quantity":3,"ordre":2,"_id":"6602a333842125ffa6bdfc0"}, {"ingredient_id":"66029ae5842125ffa6bdf89","quantity":2,"ordre":3,"_id":"6602a333842125ffa6bdfc1"}]}, {"available":true,"imageUrl":"https://images.immediate.co.uk/production/volatile/sites/30/2018/04/tequila-sunrise-18167a1.jpg?quality=90&resize=556,505"}]}
```

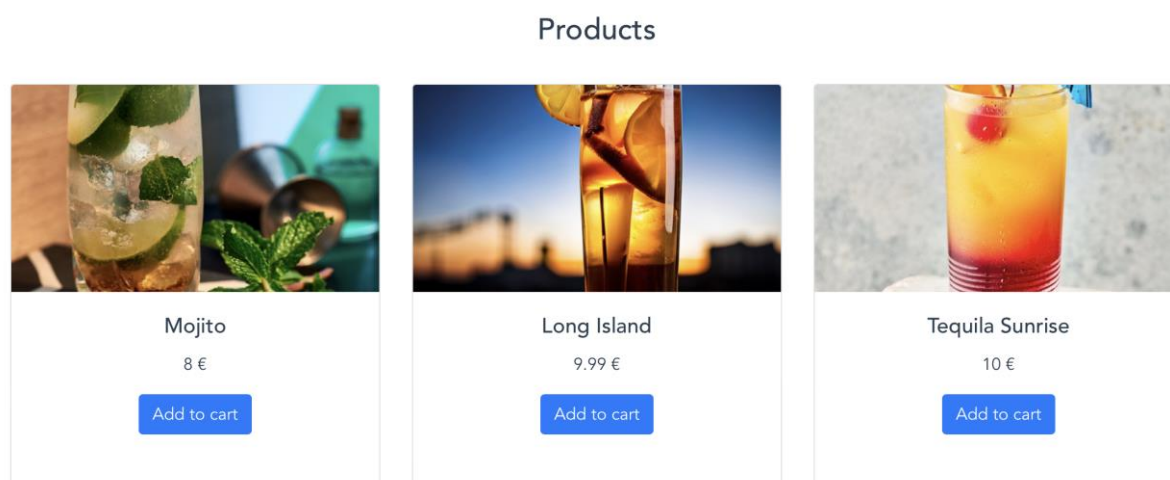
Intégration du code sur la Raspberry

Pour le code nécessaire au fonctionnement du robot contrôlé par la Raspberry, un seul fichier est nécessaire, le fichier `code_rasp.py` à la racine du git. Pour le récupérer, vous pouvez soit cloner tout le git depuis la Raspberry si vous avez un accès à Internet, soit le copier dans la Raspberry à l'aide d'une clé USB. Une fois le fichier récupéré, ouvrez-le avec Thonny sur la Raspberry. Vous pouvez lancer le script à l'aide de la flèche verte. Si le branchement électronique est effectué, les moteurs se mettront à tourner. Si ce n'est pas le cas, vous verrez uniquement les étapes de conception dans le terminal Thonny.

Utilisation du code côté client/admin

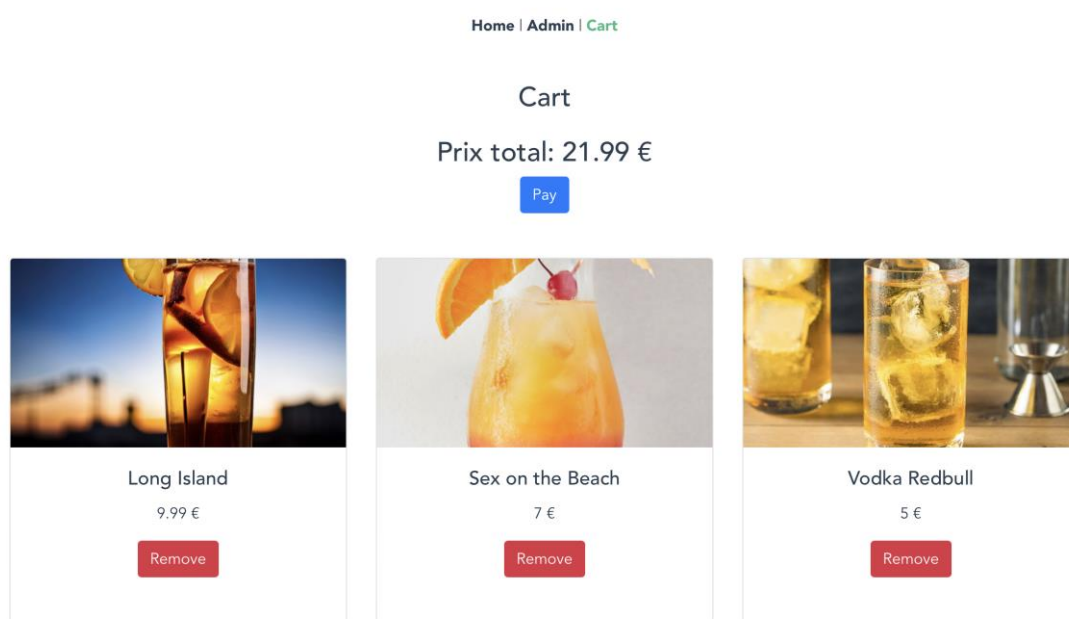
Fonctionnement côté client

En arrivant sur l'application, vous tomberez tout de suite sur la carte du bar, contenant de nombreux produits qui vous seront affichés, ainsi que leur prix.



Le fonctionnement est simple, appuyez sur Add to cart pour mettre le cocktail dans votre panier, le choix n'est pas décisif à ce moment-là, vous pourrez tout de même faire des modifications avant de valider votre panier. Lorsqu'un cocktail est ajouté, un pop-up apparait confirmant que le cocktail à bien été ajouté au panier.

Pour vérifier votre panier, allez dans l'onglet Cart dans la barre de navigation en haut de l'écran. Vous y verrez tous les cocktails que vous avez ajouté, ainsi que le prix total de votre panier. Si vous souhaitez supprimer un cocktail de votre panier cliquez simplement sur le bouton Remove en-dessous de ce cocktail. Le prix de la boisson sera déduit du prix total et il ne sera plus affiché dans votre commande.



Vous pouvez valider votre commande avec le bouton Pay. Cela vous indiquera que la commande à bien été envoyée à la liste d'attente du robot qui la préparera dès que possible. Votre panier sera vidé après l'envoi de la commande.

Fonctionnement côté admin

Lorsque vous êtes sur la page d'accueil de l'application, vous trouverez un onglet admin dans la barre de navigation. C'est dans cet onglet que vous pouvez gérer la carte et les ingrédients.

Home | **Admin** | Cart

Admin Products

Admin Ingredients

Vous aurez ainsi accès aux produits et aux ingrédients. Dans la partie ingrédients, vous pouvez trouver tous les ingrédients accessibles par le robot.

Ingredients

Add ingredient



Rhum

65f9a35238c2a5c39894c29e

Edit



Coca

65f9a35638c2a5c39894c2a0

Edit



Gin

65f9a36a38c2a5c39894c2a4

Edit

Vous y trouverez leur nom et leur identifiant accompagné d'une image. Vous pouvez changer ce nom et cette image à l'aide du bouton Edit si cela ne vous convient pas.

Edit Ingredient

Name

Rhum

Image

<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQRPAbGSdx>

Submit

Validez avec le bouton Submit une fois les changements effectués.

Il est aussi possible d'ajouter de nouveaux ingrédients avec le bouton Add ingredients.

Add ingredient

Name

Image

Submit

Écrivez le nom de l'ingrédient à rajouter, copiez le lien d'une image et enregistrez avec Submit. La page Ingredients sera mise à jour et vous retrouverez l'ingrédient ajouté à l'instant.

Le concept est le même pour la page Products.

Products

Add Product



Mojito

8 €

Edit



Long Island

9.99 €

Edit



Tequila Sunrise

10 €

Edit

Les différents produits de la carte sont affichés avec leur prix. Le bouton Edit permet de modifier les informations concernant chaque cocktail.

Edit product

Name

Quantity

Price

Image

Steps

Input steps using the following format :

```
[ { "ingredient_id": "{ingredientId}", "ordre": 1, "quantity": 4 },  
  { "ingredient_id": "{ingredientId}", "ordre": 2, "quantity": 8 } ]
```

Submit

Pour les premiers champs, rien ne change comparé à la page Edit Ingredient. Cependant, les étapes de conception doivent être dans un format particulier pour que le robot puisse lire les informations demandées. {ingredientId} correspond à l'identifiant d'ingrédient vu précédemment. L'ordre correspond à l'ordre dans lequel les ingrédients doivent être rajoutés dans le verre et la quantity correspond à la quantité qui sera servie de cette ingrédient. Une fois le produit modifié à souhait, le bouton Submit permet d'enregistrer les modifications.

Il est aussi possible d'ajouter un nouveau cocktail grâce au bouton Add Product qui ouvrira un formulaire vierge semblable à celui pour modifier un cocktail.

Add product

Name

Quantity

Price

Image

Steps

Input steps using the following format :

```
[ { "ingredient_id": "{ingredientId}", "ordre": 1, "quantity": 4 },  
  { "ingredient_id": "{ingredientId}", "ordre": 2, "quantity": 8 } ]
```

Submit

Remplissez-le de la même manière et appuyez sur Submit pour valider. Le produit sera ajouté à la carte de produits disponibles.

Sources

GitHub du projet : <https://github.com/clementmanant/Robarathon>

Raspberry Pi Imager : <https://www.raspberrypi.com/software/>

Branchement Raspberry : <https://www.freva.com/fr/configuration-materielle-hardware-du-raspberry-pi-3-b/>

Visual Studio Code : <https://code.visualstudio.com/>

NodeJS : <https://nodejs.org/en/>

ThunderClient : <https://www.thunderclient.com/>

VueJS : <https://vuejs.org/>

Thonny : <https://thonny.org/>

Installation git : <https://www.git-scm.com/downloads>

RpiMotorLib : <https://github.com/gavinlyonsrepo/RpiMotorLib>

RPi.GPIO : <https://pypi.org/project/RPi.GPIO/>