



maidbot

Dossier de présentation

Professeur encadrant
Mr. GOUGEON Pierre



LYCEE
EUGENE
LIVET
NANTES



GICQUIAUD Noam
HERY Emmanuel
MENSEAU Yoann
GICQUEL Léo

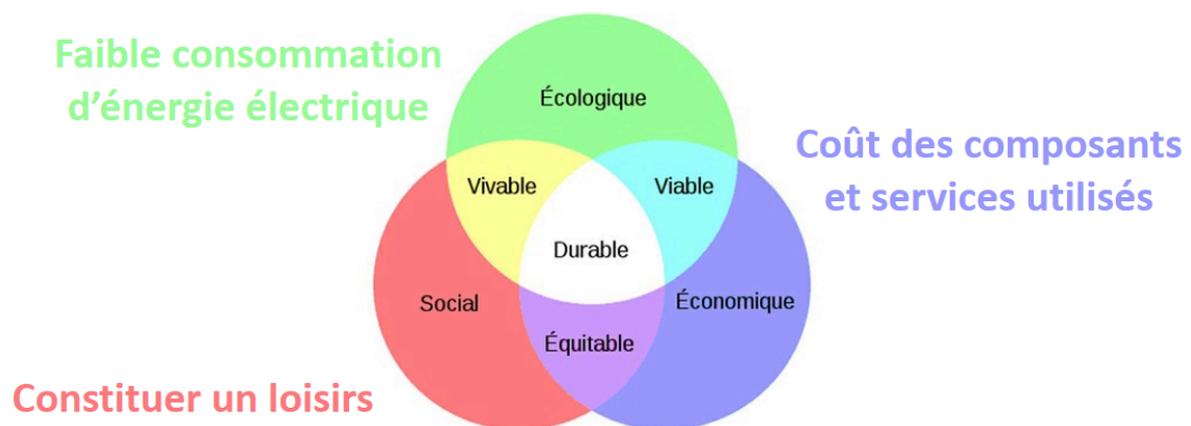
Sommaire

1. Analyse du besoin	
<i>a. Cahier des Charges</i>	2
<i>b. Diagramme d'Exigences</i>	3
2. Conception	
<i>a. Schéma Architectural du système</i>	4
<i>b. Diagramme de Cas d'Utilisation</i>	5
<i>c. Décomposition de la Chaîne d'Information</i>	6
<i>d. Diagramme de Séquence</i>	7
<i>e. Organisation du travail</i>	8
3. Réalisation	
<i>a. Solutions envisagées</i>	9
<i>b. Algorithme d'une page</i>	10
4. Validation, synthèse	
<i>a. Soucis techniques rencontrés</i>	15
<i>b. Solutions retenues</i>	16
<i>c. Futur de ce projet</i>	16
<i>d. Enrichissement personnel</i>	16

1. Analyse du besoin

a. Cahier des Charges

Le système est à **destination de bars**. Dans sa conception, il doit respecter au mieux le **Développement Durable**, reposant sur trois bases que sont le Social, l'Environnement et l'Economie, c'est-à-dire ici : être un **loisirs** pour les usagers, **consommer le moins d'énergie possible**, proposer des **services à faible coût**.

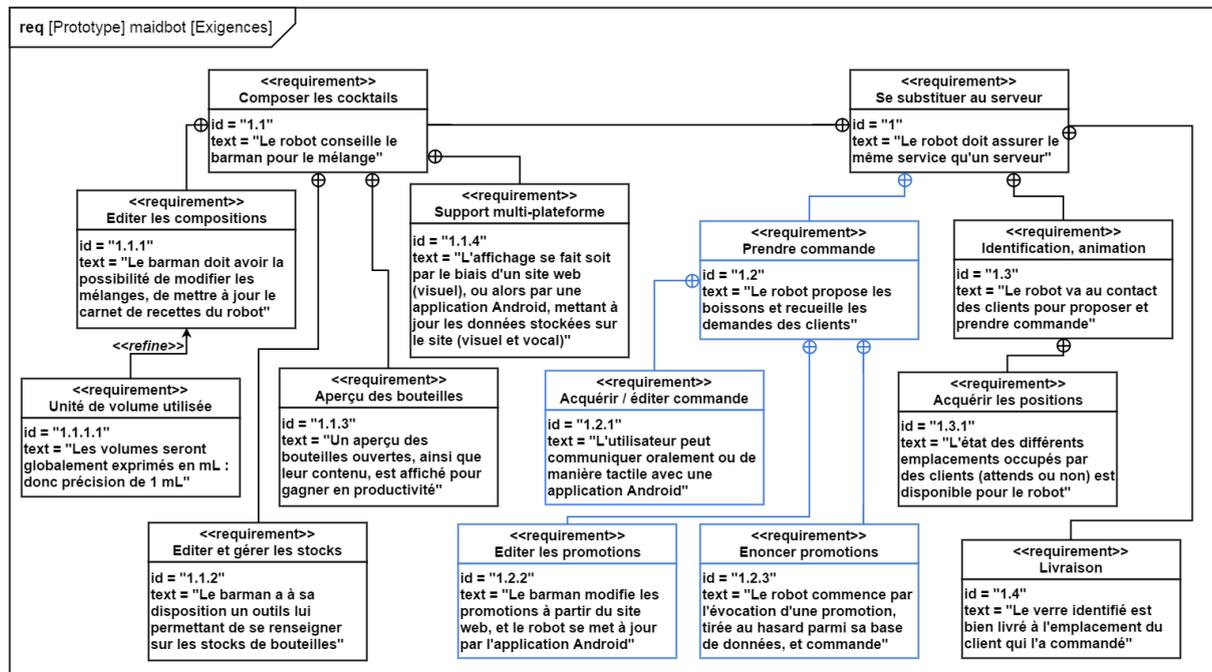


Il **offre un divertissement aux clients**, tout en acquérant les commandes. En effet, ce projet se concrétise sous la forme d'un **robot**, allant automatiquement **au contact des personnes**, et **proposant des rafraîchissements** par synthèse vocale. Cela permet de **se démarquer** par rapport aux autres établissements.

Pour le personnel du bar, plusieurs **pages web** doivent être disponible. Elles serviront à la **gestion des recettes et stocks** de boissons, ainsi qu'à l'**édition des promotions et commandes**. De cette façon, le personnel est averti de toute rupture de stock, a la possibilité de modifier les compositions de boissons, et peut gérer l'aspect commercial : réductions exceptionnelles, ou encore rectification d'erreur de commande, due à la reconnaissance vocale du client.

b. Diagramme d'Exigences

En accord avec le Cahier des Charges précédemment développé, voici ci-dessous un Diagramme d'Exigences suivant la norme SysML.



Ce diagramme sert donc de **base de travail à l'équipe**, car il **énonce les exigences à satisfaire**. Bien évidemment, étant réalisé au tout début de ce projet, des **écarts seront constatés par rapport au prototype final**.

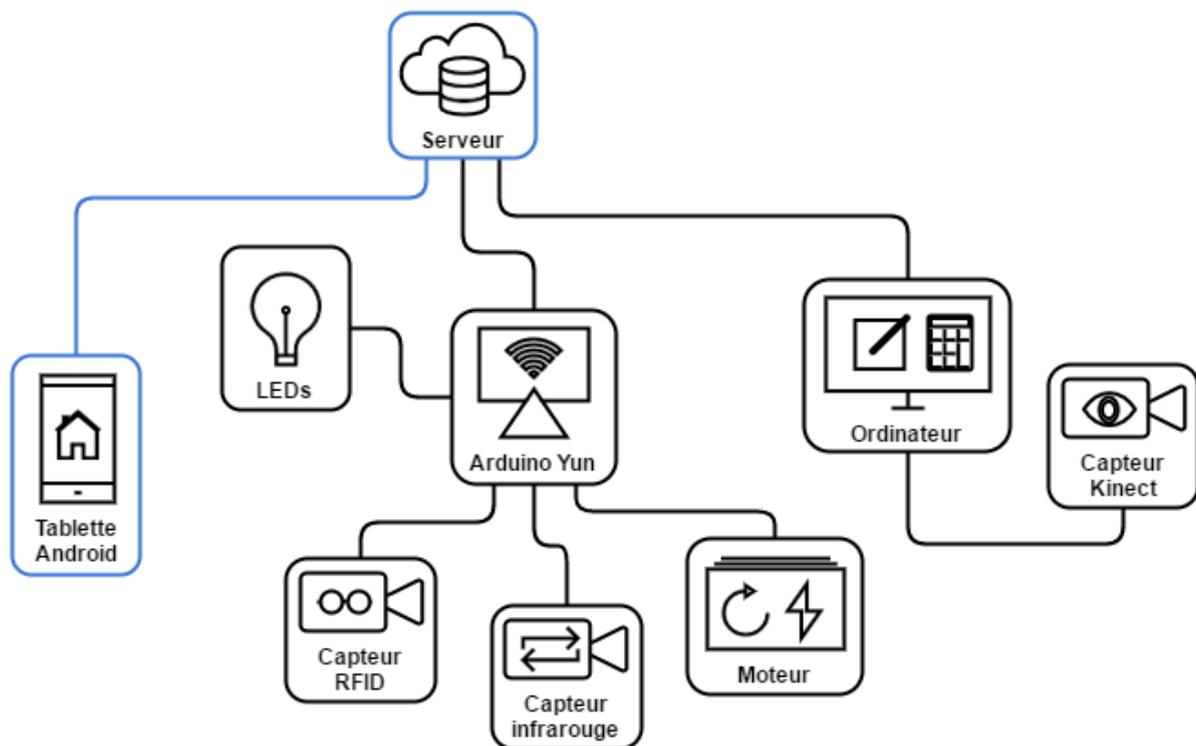
Partant de la **fonction principale "Se substituer au serveur"**, nous pouvons repérer **4 branches principales**, qui représenteront les **4 piliers** du projet. Le premier (de la gauche vers la droite) est l'**édition des recettes et des stocks**. Ensuite vient la **prise de commande**. Puis, l'**identification des clients en attente**, disposés à effectuer une commande. Enfin, la **livraison** : le robot va vers le client, et revient au besoin (récupération de verre ou autre).

Les différentes tâches ont été réparties de la manière suivante (les numéros correspondent aux ids des exigences du diagramme précédent) :

- **1.1 : Composer les cocktails** **GICQUIAUD Noam**
- **1.2 : Prendre commande** **HERY Emmanuel**
- **1.3 : Identification, animation** **MENSEAU Yoann**
- **1.4 : Livraison** **GICQUEL Léo**

2. Conception

a. Schéma Architectural du Système

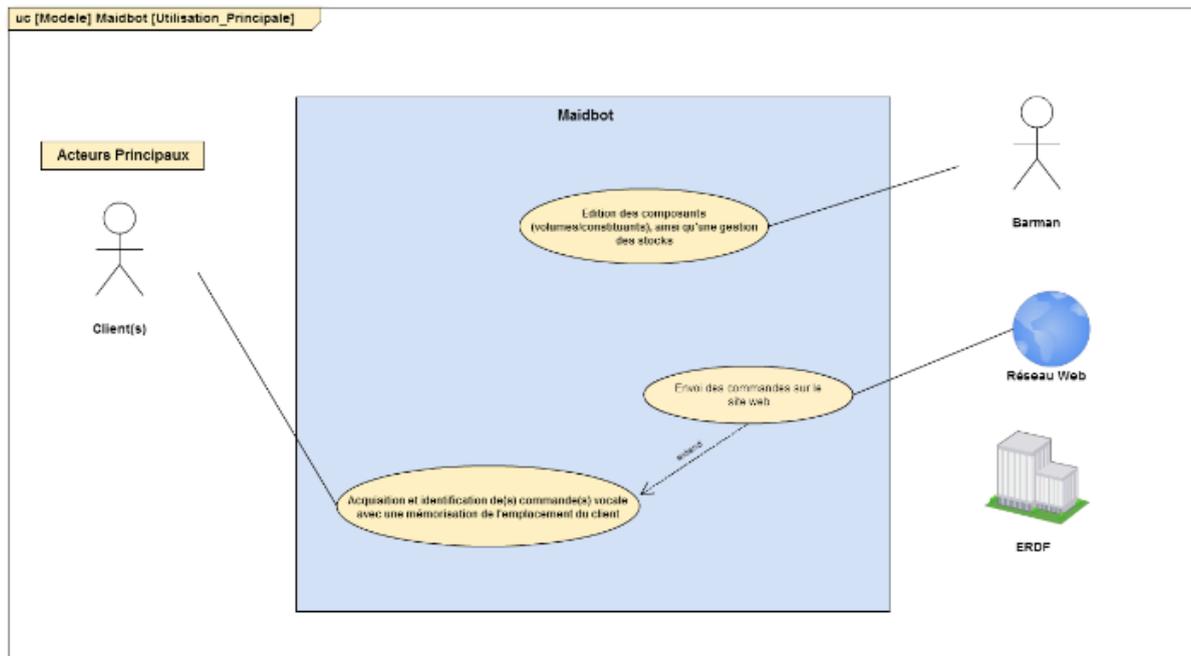


Il a été choisi de centraliser le système **autour d'un serveur**, permettant ainsi **plusieurs échanges de flux** d'informations diverses et variées, comme par exemple l'emplacement d'un client vers lequel le robot doit se rendre, ou encore les recettes des boissons proposées par l'établissement.

Les membres d'équipe ont des tâches, associées à des composants, qui peuvent être simplifiées ainsi : **gestion d'un serveur web** (*GICQUIAUD Noam*), **application sur un terminal Android** (*HERY Emmanuel*), **animation automatisée d'un robot avec une carte programmable Arduino**, intégrant une puce Wi-Fi (*GICQUEL Léo*), et enfin **détection des clients avec un capteur Kinect connecté à un ordinateur disposant d'un accès à l'Internet** (*MENSEAU Yoann*).

b. Diagramme de Cas d'Utilisation

Ci-suit un **diagramme de Cas d'Utilisation SysML**.



Malheureusement, ce diagramme n'est pas assez révélateur des services rendus par le système vis-à-vis des acteurs, on explique et rectifie donc ci-après les différents cas d'utilisation, avec les interactions homme/machine.

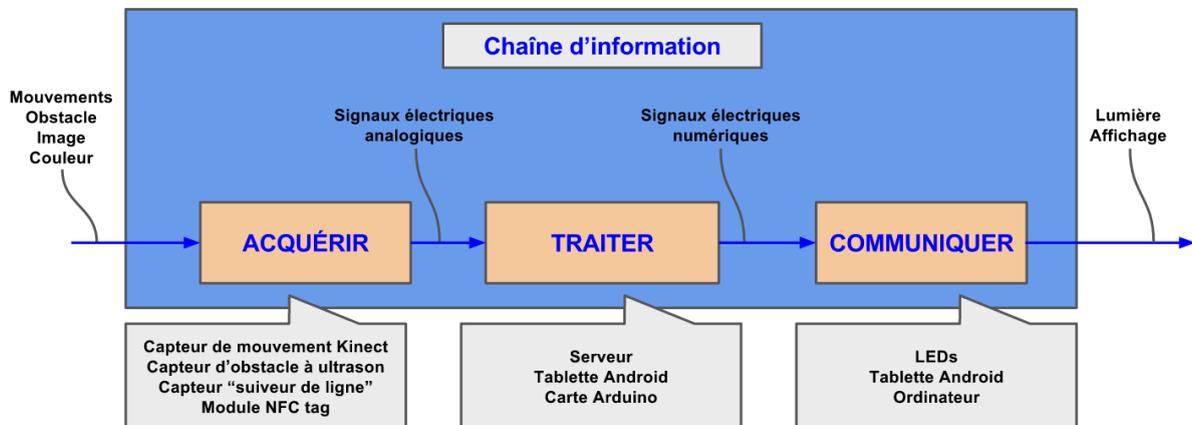
Tout d'abord, nous trouvons **deux acteurs principaux : le client, le barman**. On dénombre aussi **deux acteurs secondaires, les réseaux Web et ERDF**. Forts de la connaissance des acteurs, nous pouvons donc énoncer les interactions :

- **Barman** → *Gestion du bar*
 - *Édition promotions et commandes*
 - *Édition des compositions*
 - *Gestion des stocks*
 - *Par une page web*
 - **Réseau Web + ERDF**

- **Client** → *Commander*
 - *Par une application Android*
 - **Réseau Web + ERDF**

c. Décomposition de la Chaîne d'Information

Ci-après une **décomposition fonctionnelle de la Chaîne d'Information**.



Une fois encore, il est difficile de décrypter correctement ce diagramme, on tentera donc une approche axée sur les différentes fonctions exposées.

➤ **ACQUÉRIR**

- Cette fonction est assurée par 4 capteurs **Kinect**, d'**obstacle** par envoi d'ultrasons (capteur abandonné), **suiveur de ligne** et **module NFC**, remplacé plus tard par RFID suite à plusieurs soucis techniques, et un **terminal Android ayant l'application développée pour ce projet**.
- Les informations concernées sont les **mouvements de personnes**, les **obstacles**, les **contrastes** (pour le suiveur de ligne), **données radio**, et bien sûr l'**enregistrement vocal des commandes**.

➤ **TRAITER**

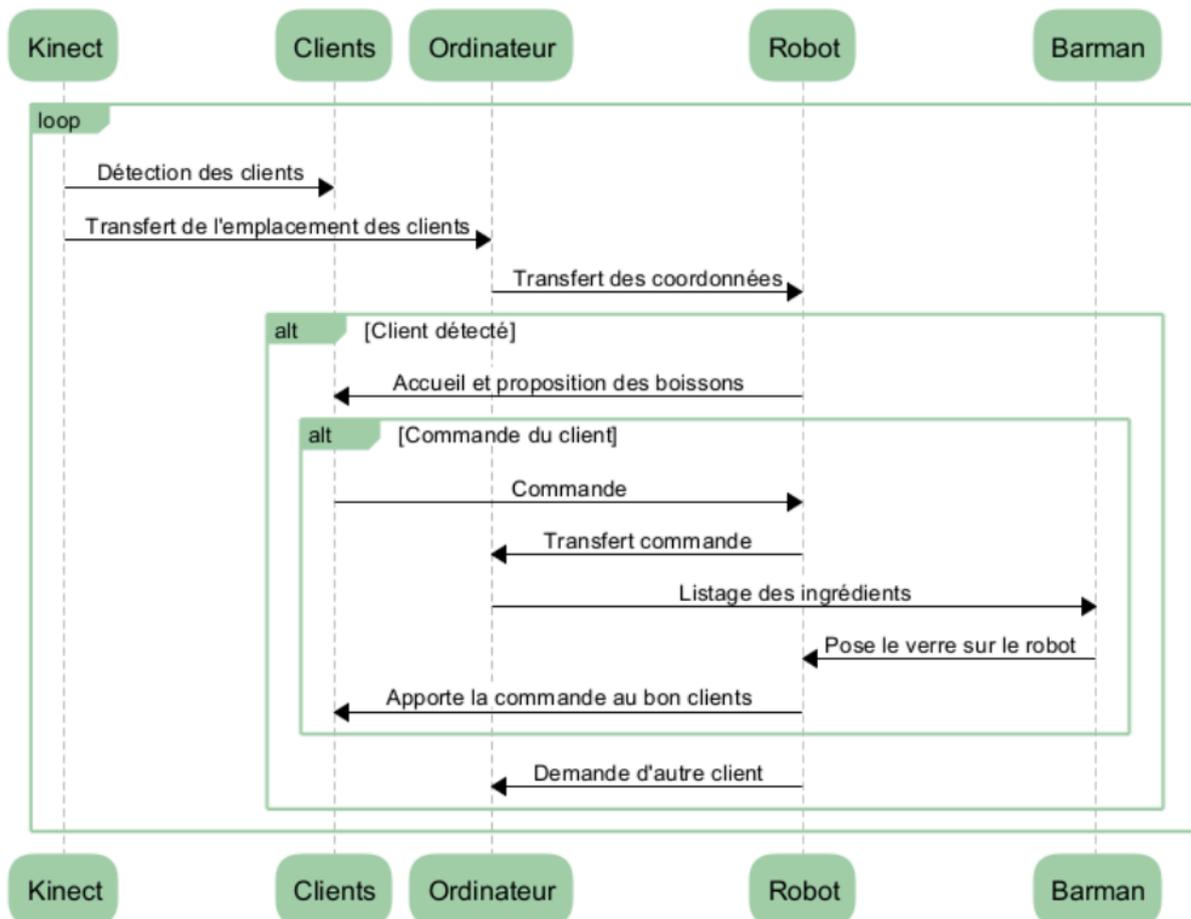
- On trouve le **serveur**, qui récupère par le mouvement des personnes pour en déduire les emplacements de clients en attente, la **tablette**, qui envoie le son au serveur Google pour récupérer la commande au format texte, et la **carte Arduino**, qui récupère le reste des informations et décide des actions à mener grâce à l'algorithme entré.

➤ **COMMUNIQUER**

- Affichage des informations par écran (ordinateur, application Android)

d. Diagramme de Séquence

Voici un **diagramme SysML de Séquence**, résumant le comportement du système en fonction de la situation, avec les différents acteurs.



Le système suit bien évidemment un **processus en boucle** : il actualise toute sorte de données, dont les principales seront évoquées tout à l'heure. Tout d'abord, le capteur Kinect **détecte les clients**. Ensuite, tout les **emplacement des clients en attente sont transférés** (on néglige ici le serveur en l'assimilant à l'ordinateur). Puis, le robot **acquiert les divers emplacements sauvegardés** auprès de l'acteur précédent, on remarque donc un itinéraire "*Client* → *Kinect* → *Ordinateur* → *Robot*". Si un client est détecté, le robot **accueille le client**. Cette condition vérifiée, si le **client commande**, le robot la **transfère vers l'ordinateur**. Un **petit listage des ingrédients est affiché** au barman, qui **pose ensuite la boisson commandée** sur le robot, qui l'**apporte au client ayant émis le souhait de l'obtenir**. Sinon, si le client ne souhaite pas ou plus commander, le robot **demande un autre client**.

e. Organisation du travail

Un tableau, mise en forme équivalente d'un diagramme de Gantt, avec les estimations et temps de travail réel, organisé en 4 grandes phases.

	Prévisionnel	Cumul	Réel	Cumul
Phase 1: Analyse du besoin	12 h	12h	7 h	7h
<i>Décrire la réalisation par un schéma architectural à partir du diagramme d'exigence</i>	2 h	2h	2 h	2h
<i>Proposer un diagramme de cas d'utilisation du projet du projet de la réalisation</i>	2 h	4h	1 h	3h
<i>Proposer une décomposition fonctionnelle de la chaîne de l'information et/ou de la chaîne de l'énergie, adaptée à la réalisation</i>	2 h	6h	1 h	4h
<i>Identifier les différents constituants de la réalisation, mis à disposition</i>	4 h	10h	1 h	5h
<i>Justifier le choix d'un capteur selon sa notice et selon le contexte du projet</i>	2 h	12h	2 h	7h
Phase 2: Conception	12 h	24h	17 h	24h
<i>Etablir un diagramme de Gantt ou tableau des activités prévisionnel</i>	2 h	14h	1 h	8h
<i>Décrire la réalisation à l'aide de diagrammes SysML pertinents</i>	4 h	18h	8 h	16h
<i>S'assurer du fonctionnement des constituants à disposition</i>	3 h	21h	6 h	22h
<i>Proposer un schéma structurel de la réalisation</i>	3 h	24h	2 h	24h
Phase 3: Réalisation	54 h	78h	59h	83h
<i>Essayer ou simuler des éléments de solutions</i>	20 h	44h	56h	80h
<i>Associer grandeurs physiques aux variables logicielles et commenter les programmes</i>	7 h	51h	3h	83h
<i>Mettre en oeuvre la chaîne de l'information au sein du prototype</i>	15 h	66h	0h	83h
<i>Réaliser des mesures pertinentes pour valider le fonctionnement de la chaîne</i>	12 h	78h	0h	83h
Phase 4: Validation	12 h	90h	6h	89h
<i>Mettre les différents diagrammes en conformité avec la réalisation</i>	6 h	84h	5h	88h
<i>Comparer des grandeur à l'aide d'un graphe/ tableau de mesures. Analyser les écarts vis-à-vis des résultats attendus selon le cahier des charges / diagramme d'exigences</i>	4 h	88h	0h30	88h30
<i>Proposer une amélioration à l'organisation du travail des coéquipiers</i>	2 h	90h	0h30	89h

On constate une **avance sur la première phase**, qui a été **rentabilisée tout au long de la seconde phase**. L'identification des constituants a en effet pris 3 heures de moins que prévu. Il aurait peut-être fallu consacrer plus de temps au diagramme de cas d'utilisation, ainsi qu'à la décomposition de la chaîne d'énergie.

La **réalisation a dépassé de 5 heures** sur le temps prévu, en laissant donc moins pour la période de validation. L'équipe, moi y compris, est restée focalisée sur les solutions, et non le reste, à savoir les explications, le lien entre les parties, et les temps de tests "grandeur nature".

Le projet n'est donc pas abouti, cependant il suffirait d'un peu de temps supplémentaire pour terminer le prototype, et songer à un déploiement commercial, ainsi qu'une campagne publicitaire.

3. Réalisation

a. Solutions envisagées

Premièrement, intéressons-nous au développement de l'application Android. Dans un **soucis d'économie**, on privilégie les **outils gratuits**, s'ils permettent de **programmer rapidement et efficacement**. Ce qu'est précisément App Inventor 2.



Développé par Google, puis légué au MIT, cet IDE (traduire *environnement de développement*) possède plusieurs avantages notables, comme la construction d'un installateur sous la forme d'un fichier *.apk, permettant de transmettre rapidement l'application par tout type de communication. En outre, il intègre plusieurs services utiles à l'évolution du projet : **reconnaissance et synthèse vocale**, **accès web** (requêtes POST et GET, faciles à traiter par un serveur), et **éléments graphiques** tels que des images, boutons, texte, afficheur web, etc.

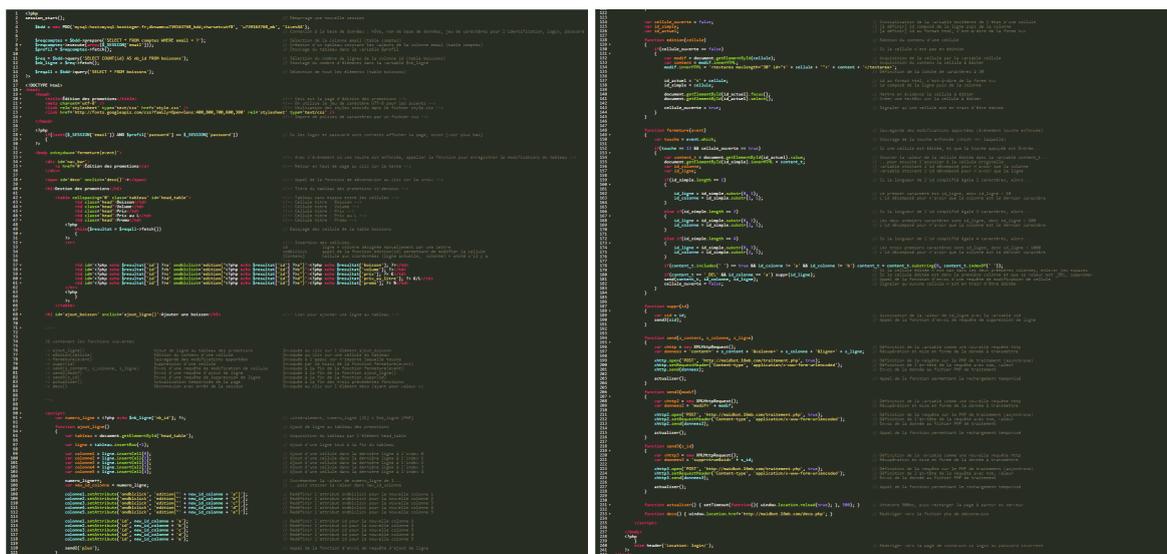
Ensuite, se pose la question de l'**hébergeur** du site, la **gratuité** étant toujours préférée selon les services proposés. De ce fait, *Hostinger.fr* a été choisi.



En plus de sa gratuité, **2 GO d'espace disque** sont disponibles, ce qui est largement suffisant pour l'affichage de quelques textes et images peut gourmands. Par compte, 2 sites peuvent même être créés ! Le **langage php est supporté**, ce qui est indispensable pour des opérations complexes, tel que l'acquisition d'une donnée par le simple envoi d'une requête http. **2 bases de données MySQL** peuvent être créées, et **gérées avec l'outil phpMyAdmin**. Niveau débit, le maximum atteint **10 Mb/s soit 1,25 Mo/s**, ce qui reste acceptable du fait du faible nombre de personnes connectées, c'est-à-dire seulement le barman, et le client. Toute inquiétude concernant la perte de donnée est presque inexistante grâce à la **technologie RAID-1** : le site est stocké sur 2 disque en miroir, réduisant les risques. Enfin, la puissance de calcul est plus que suffisante, puisque le serveur est équipé d'un **Intel Xeon E3-1230 disposant de 8 coeurs** (soit pour ce modèle 4 coeurs physiques + 4 coeurs virtuels), ayant une **fréquence d'horloge de 3,2 GHz** associés à **8 Go de RAM**. Afin de donner un ordre d'idée, cela est l'équivalent d'un ordinateur grand public, mais deux fois plus performant.

b. Algorithme d'une page

Le code de la page principale est tout aussi long que complexe (pas moins de 242 lignes en langage PHP, HTML et JavaScript). C'est pourquoi il va être synthétisé ci-dessous en algorithme, afin de repérer les fonctions auquel il peut répondre. L'intégralité du code ne sera pas disponible en annexe (économie de papier).



Code de la page maidbot.16mb.com
(adresse fournie par l'hébergeur)

ÉDITION DES PROMOTIONS ✕

GESTION DES PROMOTIONS

BOISSON	VOLUME	PRIX	PRIX AU L	PROMO
BIÈRE (PRESSION)	50 CL	1.60 €	3.20 € / L	31.5 %
SCHWEPES AGRUM'	33 CL	3.00 €	9.09 € / L	6.25 %
<input type="text" value="Non renseigné"/>		€	€ / L	%

AJOUTER UNE BOISSON

Aperçu dans un navigateur web

Début Algorithme

Connexion à la base de donnée associée à maidbot.16mb.com
Récupération de données des tables (comptes et promotions)
Balise html <head> : titre, jeu de caractères, police d'écriture

Si *mot de passe correct* **alors**

Si *touche enfoncée* **alors** appeler fonction fermeture()

Si *clic sur le titre* **alors** afficher le haut de page

Si *clic sur la croix* **alors** appeler fonction deco()

 Création de la première ligne d'un tableau

Tant que fin de la table non atteinte

 Ajouter ligne au tableau

 Récupérer et insérer les données de la table

Fin Tant que

Si *clic sur le texte "Ajouter ligne"* **alors** appeler fonction ajout_ligne()

 numero_ligne prend la valeur du nombre de lignes du tableau

Fonction ajout_ligne

 Ajouter ligne au tableau

 numero_ligne est incrémenté de 1

 Edition des attributs de chaque cellule de la nouvelle ligne :

 | appeler la fonction edition au double clic sur la cellule

 Appel de la fonction send2

Fin Fonction

cellule_ouverte prend la valeur faux

Fonction edition

Si cellule_ouverte est faux **alors**

Transformer la cellule en textbox
cellule_ouverte prend la valeur vrai

Fin Si

Fin Fonction

Fonction fermeture

touche prend la valeur du code ASCII de la touche appuyée

Si touche égale 13 (Entrée) **alors**

Si le texte tapé contient des espaces ET ne se situe pas
| dans les colonnes a ou b **alors** enlever les espaces

Si le texte tapé égale '_DEL' ET se situe dans la 1ere
| colonne a **alors** appeler la fonction suppr

Appeler la fonction send
cellule_ouverte prend la valeur faux

Fin Si

Fin Fonction

Fonction suppr

sid prend la valeur de la ligne à supprimer
Appeler la fonction send avec sid pour paramètre

Fin Fonction

Fonction send

donnees prend la valeur 'content=' + s_content + '&colonne=' +
| s_colonne + '&ligne=' + s_ligne

Envoi d'une requête POST à maidbot.16mb.com/traitement.php
Appeler la fonction actualiser

Fin Fonction**Fonction** send2

donnees2 prend la valeur 'modif=plus'

Envoi d'une requête POST à maidbot.16mb.com/traitement.php
Appeler la fonction actualiser

Fin Fonction**Fonction** send3

donnees3 prend la valeur 'suppr=true&sid=' + sid

Envoi d'une requête POST à maidbot.16mb.com/traitement.php
Appeler la fonction actualiser

Fin Fonction**Fonction** actualiser

Attendre 500 ms
Recharger la page

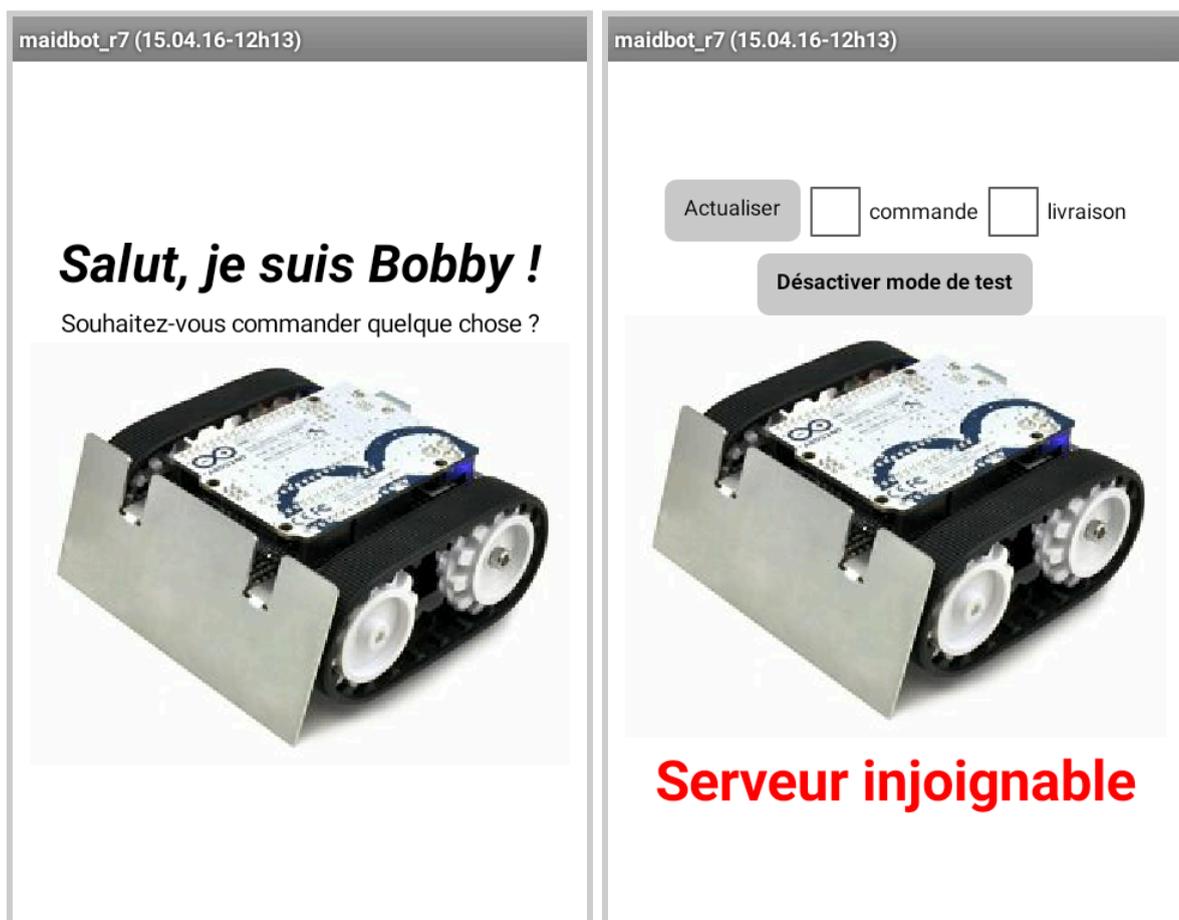
Fin Fonction

4. Validation, synthèse

a. Soucis techniques rencontrés

Durant la recherche de solutions, j'ai été confronté à de nombreux problèmes d'ordre technique. Ils seront donc énumérés, avec confirmation de résolution ou non.

Pendant le développement de l'application Android, la **source des erreurs a été principalement la connectivité web**. Il a donc fallu trouver des solutions permettant d'être **informé de l'état de la connexion** à l'Internet, et mettre en place quelques **outils simulant des valeurs** associées à des variables bien précises. Ainsi, un message clignotant rouge sur blanc est affiché, si le serveur n'envoie pas de réponse à une requête lui étant adressée. De plus, un mode pseudo-débugage a été incorporé. Il est seulement accessible par une page de connexion, elle-même cachée à l'utilisateur. Les **soucis techniques ont donc été résolus** dans la mesure du possible, le but du projet n'étant pas d'administrer un réseau, ainsi que les accès.



Mode Normal (gauche) / Mode Débogage (droite)

b. Solutions retenues

Les services utilisés, à savoir *App Inventor 2* et *Hostinger.fr*, m'ont permis de mener à bien ce projet, les **solutions sont donc retenues**. De part leur gratuité, elles représentent une économie considérable.

Pourquoi **utiliser un hébergeur**, plutôt que de stocker le site (pages, base de données, fichiers textes) sur un réseau local ? Cela nous aurait effectivement permis de gagner en rapidité, cependant la **première solution est bien plus écologique et économique** sur le long terme. En effet, les établissements peuvent, de cette manière, se passer d'ordinateurs et serveurs, et donc réduire leur consommation d'électricité ainsi que leur facture.

c. Futur de ce projet

N'étant **qu'un prototype**, le système ne peut, à ce moment présent, tout simplement **pas prétendre à une utilisation commerciale**, du fait que certains accès ne sont pas assez sécurisés. Par exemple, un autre client peut récupérer la commande de celui qui l'a originellement demandée.

d. Enrichissement personnel

Je suis très satisfait du projet dans sa globalité, malgré le fait que le prototype ne soit pas entièrement fonctionnel, et ne réponde pas à toutes les exigences. Tout au long de l'année, j'ai acquis des connaissances et savoir-faire, que j'ai pu réinvestir. J'ai aussi beaucoup appris en recherchant par moi-même les solutions. J'estime donc pouvoir dire que j'ai gagné en autonomie, en esprit d'équipe, et en compétences techniques. Ce fut une expérience très enrichissante.