

**GNG1503A**  
**Automne 2023**  
**A05-FA52**

**LIVRABLE D:**  
**Conception préliminaire**

*MEMBRES D'EQUIPE*

1. Abomo Dozeng Arnaud Joguy
2. Imad El Bacha
3. Ahmed Yassine Ben Ayed
4. Aminata Aliou
5. Chahd Machkour
6. Tania Pillay
7. Lina Azzouzi

*NUMEROS D'ETUDIANTS*

300391035  
300316248  
300393076  
300276865  
300309383  
300342827  
300343213

Date de soumission: 15 octobre 2023

Faculté de génie

**Université d'Ottawa**

## Table des matières :

1	Introduction.....	4
2	Problème .....	5
2.1	Problème 1 : Gestion d’inventaire.....	5
2.2	Problème 2 : Pauvre réseau d’organisation et de visualisation pour les activités futures .	5
2.3	Problème 3 : Composantes physique .....	5
3	Génération de Concepts.....	6
3.1	Problème 1 - Software Backend(Logiciel de Gestion d’inventaire) .....	6
3.2	Problème 2 : Software Front end .....	7
3.3	Problème 3 : Hardware.....	9
4	Analyse de solution .....	11
4.1	Tables d’Étalonnage et Matrice Décisionnelle .....	11
4.2	Schema de Conception préliminaire.....	13
5	Solution finale.....	14
6	Conclusion et Travail future .....	16
7	Notre lien Wrike .....	17
8	Références.....	17

## Sommaire des tableaux :

### 4.1Tables d’Étalonnage et Matrice Décisionnelle

**Table 1** : Étalonnage des solutions possibles

**Table 2** : Matrice décisionnelle

## Sommaire des figures :

### 4.2 Schéma de Conception préliminaire

*Fig 1: esquisses + Diagrammes Synoptiques de la solution 1*

*Fig 2: esquisses + Diagrammes Synoptiques de la solution 2*

*Fig 3: esquisses + Diagrammes Synoptiques de la solution 3*

## Résumé

Ceci est un document de conceptualisation. Ce document se sert des critères de conception étudié dans notre livrable précédent. L'objectif principale est de produire les solutions les plus probables en faveur de notre projet de gestion d'inventaire.

# 1 Introduction

Ce document constitue la suite du livrable C, dans lequel nous avons entrepris une analyse des critères de conception, nous les avons classifiés en exigences fonctionnelles et non fonctionnelles, et effectué un étalonnage en vue d'établir des spécifications cibles. Le présent rapport exploite ces informations pour générer un ensemble de concepts préliminaires. Nous examinerons, évaluerons et finalement sélectionnerons les concepts clés que nous avons l'intention de développer davantage.

Le processus commence donc par la présentation des problèmes que nous devons résoudre, suivie de l'énumération des différentes solutions (Hardware, software Backend, Software Frontend) proposées par chaque membre de l'équipe. Ces solutions seront soumises à une analyse en profondeur dans le but de choisir les plus pertinentes à l'aide d'une matrice décisionnelle, en vue de les combiner pour créer notre solution finale.

## 2 Problème

### 2.1 Problème 1 : Gestion d'inventaire

Le client (Plates-formes Numériques et Interopérabilité (PN&I)) a expliqué qu'ils ont présentement un système de gestion d'inventaire qui tend vers la défaillance, c'est à dire qu'ils ont de plus en plus de perte d'équipement, une perte de temps pour refaire l'inventaire et une perte d'argent pour au final avoir un suivi inconsistant de ces articles.

### 2.2 Problème 2 : Pauvre réseau d'organisation et de visualisation pour les activités futures

Le client a aussi parlé du fait qu'au sein de leur organisation ils sont, le plus souvent, informés en retard si un article vient à être en rupture de stock. Alors ils aimeraient vraiment un système d'alerte par notifications qui les alerte d'un état alpha ou beta de leur objet dans l'inventaire.

### 2.3 Problème 3 : Composantes physique

Le client a parlé de leur système actuel qui demande un gros effort humain car les solutions sont pratiquement toutes physiques. C'est-à-dire que le renouvellement de l'inventaire se fait manuellement donc il faut compter les articles pour savoir la quantité qui entre/sort de l'inventaire. De plus pour trouver un objet il faut aveuglement faire confiance aux données récoltées lors du renouvellement de l'inventaire et se déplacer de rangés en rangés pour trouver l'objet recherché.

### 3 Génération de Concepts

Pour essayer d'apporter les solutions aux problèmes posés nous allons avoir les divisions suivantes :

- Software Backend : se rapporte au logiciel à créer pour la gestion d'inventaire (Problème 1)
- Software Front end : constitue l'interface utilisateur (Problème 2)
- Hardware : constitue la partie physique de nos solutions pour remédier au problème de détection des articles une fois qu'ils sont mis dans leur emplacement (Problème 3)

#### 3.1 Problème 1 - Software Backend(Logiciel de Gestion d'inventaire)

##### **Solution1 A (Arnaud): Java**

Pour une solution logicielle backend je propose l'utilisation du langage de programmation java qui est reconnu pour sa sécurité, ce qui est essentiel lorsque nous travaillons avec des données sensibles telles que des informations d'inventaire. Il est de bonnes performances donc important pour gérer des bases de données d'inventaire volumineuses et des opérations en temps réel. Il est distribué sous une licence open source, ce qui signifie que nous pouvons l'utiliser sans frais ni restrictions majeures.

##### **Solution1 B (Imad): Arduino**

Une solution Backend en IDE Arduino, qui s'occupera de la sécurité des données grâce à plusieurs facteurs, tels qu'un contrôle d'accès avec authentification à deux facteurs (par mot de passe et biométrie). Le chiffrement des données en transit à l'aide de protocoles sécurisés tels que SSL/TLS doit aussi être inclus. Et entre autres des mises à jour et correctifs réguliers, une gestion des messages d'erreurs, ... Tout cela suivant la conformité légale.

##### **Solution1 C (Chahd): Python**

Une solution backend en Python, qui utilise une base de données SQL pour le suivi des mouvements de stock. Cette solution utilise le Framework Django pour créer un modèle de base de données pour les produits et les mouvements de stock. Chaque mouvement de stock est enregistré dans la base de données avec des informations telles que le produit concerné, la quantité, le type de mouvement (entrée, sortie, etc.), et la date.

##### **Solution1 D (Ahmed): Language C**

La solution à ce problème est un codage en langage c, car c'est bien là son rôle développer des applications, et des systèmes d'exploitation. Et même s'il est rustique et primaire, il reste tout de même la base qui a permis aux informaticiens de créer le monde virtuel dans lequel nous vivons tout en se peaufinant avec l'âge.

**Solution1 E (Tania): Arduino**

La solution à ce problème est en utilisant Arduino Uno car elle est très conviviale et très sécurisée de tel sorte que seules les personnes qui y ont accès peuvent l'utiliser. Il doit être possible de récupérer les données en cas de désastres.

**Solution1 F (Aminata): JAVA**

Pour la partie logicielle de ce système il serait plus adapté de servir du l'environnement de développement java car il est gratuit, flexible et nous permet de trouver des solutions et des alternatives diverses pour un problème de donné. De plus la base de données pour ce langage informatique est extensible en fonction de l'espace qui lui est réservé car il ne cesse pas grandir proportionnellement aux donnés.

**Solution 1 G (Lina): Python**

Python est un langage de programmation très facile à apprendre. Il possède une bibliothèque vaste ce qui nous permettrait d'effectuer plusieurs commandes. Malgré sa polyvalence, il peut être moins adapté à certains domaines à cause de sa vitesse d'exécution. Cependant, python est disponible sur plusieurs plateformes, ce qui ne nous restreindra pas sur le choix de celle-ci.

### 3.2 Problème 2 : Software Front end

**Solution2 A (Arnaud): Excel**

Je propose Excel de la suite Microsoft. C'est un logiciel de gestion d'inventaire largement utilisé et familier à la plupart des utilisateurs et employés de bureau. Il ne nécessite généralement pas de formation intensive pour commencer à créer et à gérer des feuilles de calcul d'inventaire. Ce qui permettra une réduction des coûts du projet. En outre il est disponible et utilisable hors ligne. Il permet en plus d'ajouter, supprimer ou mettre à jour des données d'inventaire en temps réel, ce qui vous permet de suivre l'état actuel des inventaires de notre client.

**Solution2 B (Imad): Arduino**

Une solution serait Arduino pour développer l'interface utilisateur. On pourrait par exemple Créer un tableau de bord central où les utilisateurs peuvent accéder à toutes ces fonctionnalités en un seul endroit, y inclure des icônes, des menus déroulants et des boutons d'action pour simplifier la navigation. En ce qui concerne la visualisation des données, Créez des graphiques à barres ou à secteurs pour représenter visuellement la répartition des articles dans l'inventaire.

**Solution2 C (Chahd): APP**

La solution serait : Une application Mobile; Développer une application mobile pour iOS et Android, permettant aux différents utilisateurs de gérer l'inventaire à partir de leurs smartphones ou tablettes en utilisant Flutter par exemple. Pour le Design Visuel, il faudra choisir une palette de couleurs appropriée et un style visuel cohérent pour l'application. Utiliser des icônes et des éléments graphiques pour améliorer l'expérience visuelle.

**Solution2 D (Ahmed): Microsoft Power bi**

La partie software du système, donc la gestion des informations doit se faire à mon avis à travers power bi. Il surpasse excelle dans tous les domaines. Car si Excel est un tableur traditionnel, avec fonctionnalités fixe, power Bi est un outil d'analyse avancée offrant une variété de fonctionnalités pour l'interprétation et la comparaison de données.

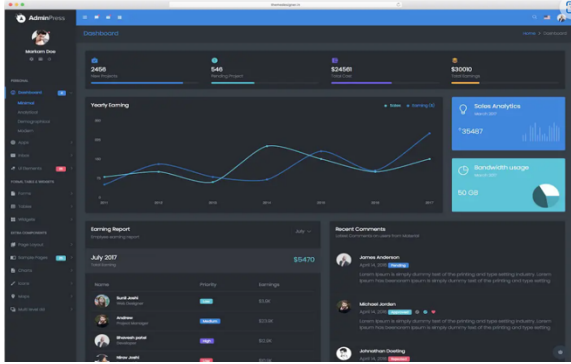
**Solution2 E (Tania): Arduino**

La solution à ce problème est l'utilisation d'Arduino Uno comme interface. C'est facile à utiliser et facile à comprendre, le code est ainsi facile à modifier. C'est aussi facile à utiliser pour tester des prototypes et les accessoires de Arduino Uno sont facilement disponibles.

**Solution2 F (Aminata): WEB**

Pour la partie interface de notre solution, on pourrait se servir du langage html pour produire une interface web. C'est l'outil avec la méthode la plus simple et basique en termes de programmation dans le but de créer une interface web pour un logiciel quelconque. HTML est généralement utilisé pour générer des interfaces modernes et très développées de nos jours Par exemple<sup>1</sup> :





**Image 1** : Capture d'écran de l'interface d'Admin Press

### **Solution 2 G (Lina): App**

Une application qui peut être téléchargée sur tablette (apple ou autre) et sur téléphone (apple ou autre). L'interface est simple à utiliser et les notifications permettent au client de savoir toutes les informations nécessaires sur les items. Malgré le fait que cette méthode a besoin de plusieurs mises à jour, elle peut être très efficace pour gérer la fréquence des commandes d'items.

### 3.3 [Problème 3 : Hardware](#)

#### **Solution3 A (Arnaud): NFC**

**NFC** (Near Field Communication) est un protocole de communication donc je recommande pour une solution coté Hardware car il est essentiel pour une gestion en temps réel, il permet une collecte d'informations rapide sur les articles d'inventaire. Son utilisation permettra de réduire les erreurs de saisie manuelle et d'améliorer la fiabilité des informations d'inventaire donc notre client voudrai.

#### **Solution3 B (Imad): QR code**

La solution serait les QR codes. Les QR codes ne nécessitent pas d'alimentation électrique. Ils sont simplement des images imprimées ou affichées, de sorte qu'il n'y a pas de consommation d'énergie à proprement parler. Aussi, Les QR codes sont statiques, ce qui signifie qu'ils ne sont pas directement connectés à un système informatique. Cependant, ils peuvent vous rediriger vers une application ou un site web avec un simple scan.

**Solution3 C (Chahd): RFID**

Les lecteurs RFID peuvent se présenter sous la forme de dispositifs portables déployés en nombre dans chaque entrepôt. Ces lecteurs sont conçus pour interagir avec les puces RFID en captant les signaux radiofréquences émis par ces dernières à travers des antennes intégrées, ce qui leur permet de récupérer les informations contenues dans ces puces.

**Solution3 D (Ahmed): QR code**

Je dirais que la meilleure solution sera l'usage de QR code. Car bien que leur identification reste manuelle et requiert l'intervention humaine, il reste tout de même efficace. Il ne coûte pas de l'argent, ne s'entretiennent pas et ne peuvent bugger ou être piraté.

**Solution3 E (Tania): RFID**

La solution serait d'utiliser RFID. Je pense que la solution à ce problème est de choisir pour chaque items les puces qui conviennent et quelles doivent être placées à des endroits qui vont aider à le lire facilement par un lecteur RFID.

**Solution3 F (Aminata): NFC**

Pour la partie logicielle de notre solution il serait favorable de penser à la technologie NFC en d'autres termes Near Field Communication (Communication en champs proche). Cette technologie permet de transmettre des informations D'un appareil à un autre de façon automatique très rapidement sans avoir recours à un appariement. Elle facilite la circulation de plusieurs donné par jumelage et le fait qu'elle soit dotée d'une technologie Bluetooth lui permet un transfert de données audio. Étant qu'elle peut fonctionner de trois façons différentes, dans le cadre spécifique au projet nous nous servons du mode pair à pair qui lui est un mode qui permet l'échange d'information entre deux terminaux équipés du NFC.

**Solution 3 G (Lina): RFID**

Les puces RFID(Radio Frequency Identification) seraient un atout à avoir lors de la résolution de ce problème de conception. Elles permettront au client de faire l'inventaire sans à avoir à effectuer un contact physique avec l'item. Malgré le fait que leur prix peut être élevé, elles réduiraient considérablement l'erreur humaine et économisera beaucoup de temps au personnel. Leur traçabilité se fait de façon précise et rapide.

## 4 Analyse de solution

Après avoir analysé les trois solutions pour chaque sous-système, il est évident de constater que pour la partie Software Frontend, les trois meilleurs sous-systèmes sont Microsoft Power BI, Interface Web (via HTML), et Interface Excel. Ils sont tous très populaires, ce qui donne accès à beaucoup plus de sources à exploiter qu'avec Arduino ou la conception d'une application tierce. De plus, ils sont tous dirigés par de grandes entreprises (Microsoft pour Power BI et Excel, W3C pour HTML) et présentent ainsi plus de crédibilité pour être utilisés, car ils présentent moins de risques en termes de sécurité ou de failles à exploiter. Ensuite, en ce qui concerne la partie Software Back-End, nous avons décidé de porter notre choix sur les trois sous-systèmes suivants : Arduino, Python, et Java, en raison encore une fois de leur popularité et de la vaste disponibilité des ressources qu'ils offrent (Arduino faisant ici exception en termes de popularité, mais présentant tout de même des ressources nombreuses et variées fournies cette fois par l'université). De plus, ils sont tous gratuits à utiliser et ne représentent donc pas de contraintes budgétaires, contrairement à d'autres langages. De plus, l'expertise technique est à prendre en compte, car, contrairement aux trois langages cités précédemment, aucun membre de notre équipe ne saurait manipuler le langage C, ce qui représenterait plus une contrainte qu'une solution. Finalement, pour la partie Hardware, nous avons pris en considération les suggestions de chaque membre de l'équipe et avons remarqué que nos arguments se ressemblaient et se portaient sur les mêmes sous-systèmes, à savoir RFID, QR code, et NFC. Nous avons constaté que RFID et NFC présentaient à peu près les mêmes avantages (un échange d'informations client/serveur simplifié et interactif, une localisation des articles rapide et précise, etc.). Le QR code étant déjà utilisé, nous pensons pouvoir le développer afin de le rendre plus fonctionnel et nécessitant moins d'intervention humaine.

### 4.1 Tables d'Étalonnage et Matrice Décisionnelle

**Table 1** : Étalonage des solutions possibles

Solutions	Solution 1	Solution 2	Solution 3
Software Front end	Application (Microsoft Power BI)	Interface Web	Interface Excel
Software Backend	Arduino	Python	Java
Hardware	RFID	NFC	QR code

**Table 2 : Matrice décisionnelle**

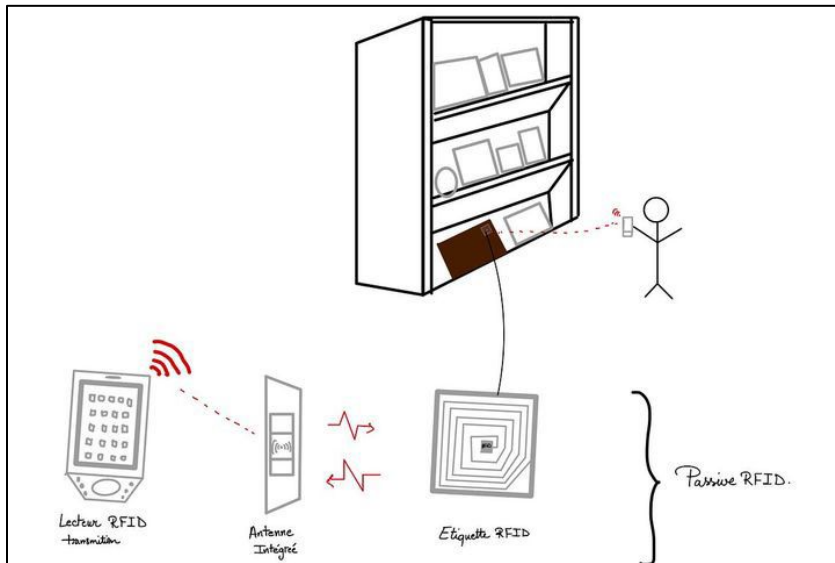
L'importance sera évaluée sur un intervalle de 1 à 5 de telle sorte 5 désigne une importance critique et 1 désigne une importance faible.

Les solutions qui satisfont les critères seront aussi évaluées sur un intervalle de 1 à 3 de telle sorte que 3 désigne : une satisfaction complète du critère tandis que 1 désigne que le critère n'est pas satisfait

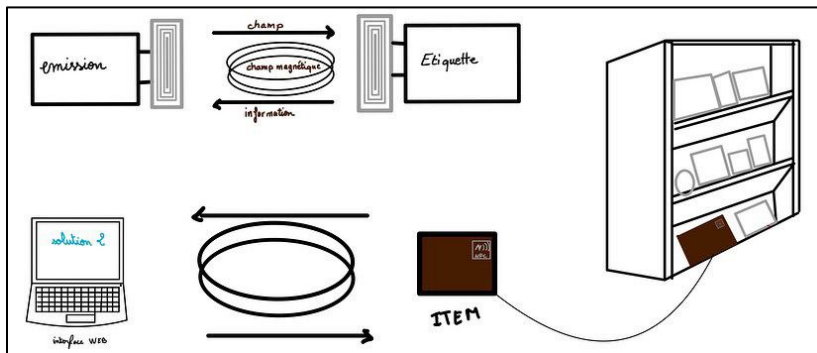
<b>Solutions</b>	<b>Importance</b>	<b>Solution 1(RFID)</b>	<b>Solution 2(NFC)</b>	<b>Solution 3(QR Code)</b>
<b>Critères de conception</b>				
Coût	2	2	3	3
Taille	2	2	2	2
Échange d'information client/serveur	4	3	3	2
Contrôle à distance des entrepôts	5	3	2	1
Réduction de l'intervention humaine	3	2	2	1
Localisation des articles	5	3	3	1
Actualisation instantanée de l'inventaire au besoin de l'utilisateur	4	3	2	1
Système d'information par notification	4	2	2	1
Sécurisation et accessibilité des informations	4	2	3	1
	<b>Total</b>	<b>84</b>	<b>81</b>	<b>43</b>

## 4.2 Schéma de Conception préliminaire

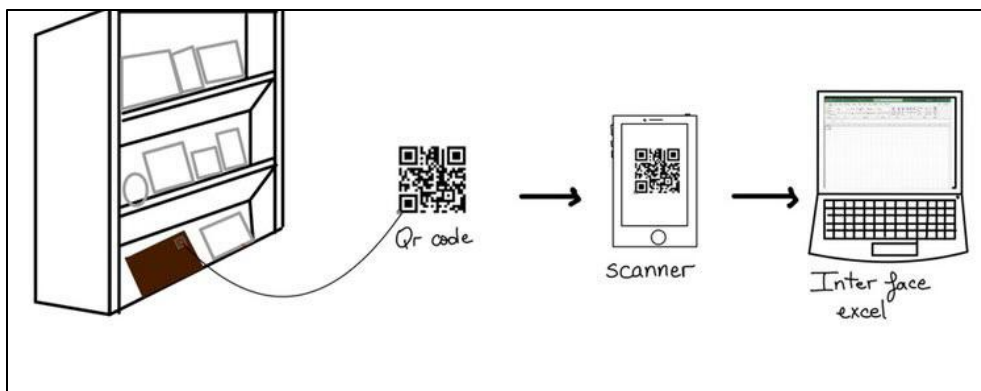
**Figure 1:** esquisses + Diagrammes Synoptiques de la solution 1



**Figure 2:** esquisses + Diagrammes Synoptiques de la solution 2



**Figure 3:** esquisses + Diagrammes Synoptiques de la solution 3



## 5 Solution finale

Suite à l'étude minutieuse des idées de chaque membre du groupe, et à l'aide de la matrice décisionnelle, notre choix final se portera alors sur la solution 1 avec le système RFID comme Hardware pour ce projet.

Notre solution finale sera donc composée d'une partie software backend (partie logiciel) qui sera réalisée à l'aide d'Arduino, d'une partie Software front end (Interface et lieu de navigation) qui seront réalisés à l'aide de Microsoft Power BI et enfin d'une partie Hardware qui sera gérée par un système RFID.

Pour la partie logiciel, les cartes électroniques Arduino sont faciles d'utilisation et son Open Source (c'est-à-dire les possibilités de libre redistribution, d'accès au code source et de création de travaux dérivés<sup>2</sup>) et Open Hardware. Leurs tarifs sont abordables et les parties de codes accessibles sont faciles d'accès au sein de la communauté, ce qui pourrait nous faciliter la tâche en termes de recherche d'inspiration à l'égard de notre logiciel. Alors pour se faire nous aurons une carte Arduino et un environnement de développement Arduino avec lesquelles effectuer les démarches nécessaires pour avoir des possibilités de générations d'un logiciel.

Pour comprendre notre choix il faut savoir que ce logiciel va optimiser la collecte : car toutes les informations sont stockées dans un même endroit et de façon organisée. Il est sécuritaire. En effet l'outil attribue les droits d'accès à tous les niveaux. Les données sont à l'abri et seulement les utilisateurs qui ont un droit d'accès peuvent consulter des bases de données.

En ce qui concerne la partie application et interface nde de l'API REST Power BI (C'est une interface de programmation d'application qui respecte des contraintes d'architecture de REST qui est Représentationnal State Transfer) qui permet de à deux logiciels de communiquer malgré leurs différences entre les systèmes d'exploitation en utilisant des requêtes HTTP<sup>3</sup> (HyperText Transfer Protocol). Un des avantages majeurs de Microsoft Power BI est son « pack de contenu » : les utilisateurs peuvent y jeter un coup d'œil même si la base de données est absente. Les graphiques, les schémas et les tableaux sont visibles sans toucher aux données brutes. Cela préserve ces dernières des erreurs humaines, car les chiffres sont donnés au repos. Tous ceci nous permet d'être sûr de notre choix, d'affirmer que Microsoft BI est la meilleure alternative possible.

Notre solution physique se focalisera sur le système de gestion à matériels RFID : il s'agit des puces, antennes et lecteurs RFID. RFID ou Radio Frequency Identification possède un système constitué d'un capteur ou étiquette (adaptateur d'une puce passive qui permet d'établir un lien entre cette puce et un lecteur RFID), d'une antenne intégrée dans la puce et d'un lecteur qui permet de détecter la position de la puce en émettant un son. Ce lecteur sera à écran tactile avec un clavier physique et aura la possibilité de lire des codes-barres.

Étiquettes RFID			
Métriques	Relation	Valeur	Unité
Poids	=	5	g
Longueur	<=	8	cm
Largeur	>=	6	cm
Capacité de stockage	>=	8	kilooctet

Antennes RFID intégrée dans la puce			
Métriques	Relation	Valeur	Unité
Portée de lecture ou bande passante	<=	8.5	m
Impédance	=	50	Ohms
Largeur de faisceau	=	65	degré
Game de frequence	=<...<=	865 - 960	MHz

Lecteur RFID à écran tactile			
Métriques	Relation	Valeur	Unité
Plateforme	=	Android	N/A
Poids	<=	360	Grammes(g)
Longueur	<=	132.8±0.5	cm
Largeur	<=	18.1	cm
gain de crête	=<...<=	3 < 6	DBi (decibel dipole)
Game de frequence	=<...<=	865 – 870	MHz

## 6 Conclusion et Travail future

Nous avons récolté les idées et concepts préliminaires de toute l'équipe lors de la phase de remue ménage, par la suite nous les avons analysés et arrêtés trois sous-systèmes de la solution finale que nous avons développé également en concepts préliminaires. Nous avons utilisé une matrice décisionnelle sur une échelle de 1 à 3 pour déterminer la solution finale qui est le RFID en se basant sur les critères de conception telle que le cout, taille, échange d'information clients/servers, contrôle à distance de l'entrepôt, réduction de l'intervention humaine, localisation des articles, actualisation instantanée de l'inventaire au besoin de l'utilisateur, le système d'information par notification et la sécurisation et l'accessibilité des informations. Nous avons appris que le RFID peuvent utiliser plusieurs types d'étiquettes telles que les étiquettes actifs, passifs, cependant celles de NFC est plutôt passifs et le code QR est tout simplement une image à scanner. Le poids de l'importance est de 1 à 5 où 5 = très important et 1= moins important.

Pour nos projets futurs, nous devons faire le plan et déterminer le cout du projet, développer les prototypes 1, 2, 3 et les présenter à l'évènement qui aura lieu lors de la journée de conception qui aura lieu le 30 novembre 2023. Il nous faudra aussi crée un manuel d'utilisateur et bien le documenter.



## 7 Notre lien Wrike

<https://www.wrike.com/open.htm?id=1231945349>

## 8 Références

1. 27 meilleurs modèles d'administration HTML5 / CSS3 pour la conception de matériel, 2019 | Webypress

2. « code source ouvert » [archive], Grand Dictionnaire terminologique, Office québécois de la langue française

3.

<https://www.bing.com/ck/a?!&&p=84ab5ca60b44f4e5JmltdHM9MTY5NzMyODAwMCZpZ3VpZD0xMTFmNjUxMy0wOWIOLTY5NzctMTZkZC03NjNjMDgyMDY4OTEmaW5zaWQ9NTczOA&pntn=3&hsh=3&fclid=111f6513-09b4-6977-16dd-763c08206891&psq=c%27est+quoi+une+api+rest&u=a1aHR0cHM6Ly93d3cudXB0cmVuZHMuZnIvcXUtZjZkZC03NjNjMDgyMDY4OTEmaW5zaWQ9NTczOA&ntb=1>