

Rapport Projet Développement Informatique

Gestion de Stock



Tuteur : BELAID Djamel

Dépôt GitLab : https://gitlabens.imtbs-tsp.eu/dzenan.cindrak/pro3600_17-gestiondestocks

Sommaire :

I - Cahier des Charges :	3
1. Présentation du projet :	3
2. Objectif :	3
3. Description fonctionnelle des besoins :	3
4. Objectifs :	3
5. Délais :	4
II - État de l'art :	4
1. Stock Controller :	4
2. Storage-Manager Free :	5
3. MonStock :	6
III - Analyse fonctionnelle :	7
1. Principe de fonctionnement :	7
2. Description des fonctions :	7
3. Description des classes :	8
IV – Maquettes :	13
V – Backend :	19
2. Description de la base de données :	20
VI – Conclusion :	21
1. Perspectives :	21
2. Conclusion générale :	22
3. Conclusions personnelles :	22
a. Vincent LABARRE :	22
b. Valentin BAILLEUL :	23
c. Mohamed CHAMROUK :	23
d. Dzenan CINDRAK :	24
VII - Annexes :	24
Annexe 1 : Compte-rendus de réunion :	24
Annexe 2 : Compte-rendus de l'autoformation :	38
1. Vincent LABARRE :	38
2. Valentin BAILLEUL :	39
3. Mohamed CHAMROUK :	40
4. Dzenan CINDRAK :	40
Annexe 3 : Plannings :	40
1. Vincent LABARRE :	40
2. Valentin BAILLEUL :	42
3. Mohamed CHAMROUK :	43
4. Dzenan CINDRAK :	44
VIII - Bibliographie :	45

I - Cahier des Charges :

1. Présentation du projet :

La gestion des stocks est fondamentale que ce soit dans une entreprise ou une association à but non lucratif. Celle-ci permet d'éviter l'achat de matériel inutile et donc de faire des économies. Ainsi nous est venu l'idée de développer une application permettant à une entité quelconque d'administrer ses stocks simplement. L'application sera développée pour mobile Android, cela permettra à une grande partie des gens de l'utiliser (80% des parts du marché sont détenues par Android).

2. Objectif :

Nous souhaitons développer une **application Android**, facile d'utilisation, qui a pour but d'aider à la gestion des stocks d'une entreprise ou de n'importe quelle autre entité. L'application s'adaptera aux stocks de l'entreprise en question et devra donc être flexible et fournira la possibilité d'ajouter toute catégorie d'objet (des fruits comme des câbles et des écrans).

3. Description fonctionnelle des besoins :

Premièrement, les stocks seront recensés sur une **base de données** distante et non embarquée sur le terminal mobile. La base de données sera implémentée sur un **serveur que l'entreprise possède**. La connexion à cette base se fera par authentification (soit par un mot de passe générique soit par utilisateur si un système d'authentification est mis en place cf. Objectifs secondaires).

Les champs nécessaires pour une gestion optimale sont :

- La catégorie (câble, écran, fruit ...)
- La référence de l'objet (le code barre de l'objet, identifiant unique)
- L'emplacement de l'objet (le lieu où est stocké l'objet. Chaîne de caractère libre)
- Utilisateur ayant ajouté l'objet (nom de l'utilisateur sous forme de chaîne de caractère)
- Commentaire sur l'objet (par exemple : non-compatibilité de certains équipements avec d'autres équipements, version du système d'exploitation de l'équipement ...).

4. Objectifs :

a) Objectifs principaux

L'**objectif principal est la gestion des stocks**. On peut alors définir les fonctions nécessaires pour cela.

Ainsi, nous aurons besoin impérativement des 4 fonctions suivantes :

- Ajouter du matériel dans la base de données.
- Retirer du matériel de la base de données.
- Afficher la base de données.
- Effectuer des recherches dans la base de données.

b) Objectifs secondaires

Les objectifs secondaires de l'application, que nous ferons si nous avons le temps, sont de **fournir une alerte en cas de faible quantité d'objet en stock** et mettre en place un **système d'authentification** permettant de savoir qui a modifié la base de données.

5. Délais :

Livrable	Description	Date limite
1	<ul style="list-style-type: none">• Cahier des charges• Analyse fonctionnelle• État de l'art	10/02/2020
2	<ul style="list-style-type: none">• Maquettes	16/02/2020
3	<ul style="list-style-type: none">• Prototype	16/03/2020
4	<ul style="list-style-type: none">• Programme final• Tests• Rapport	18/05/2020

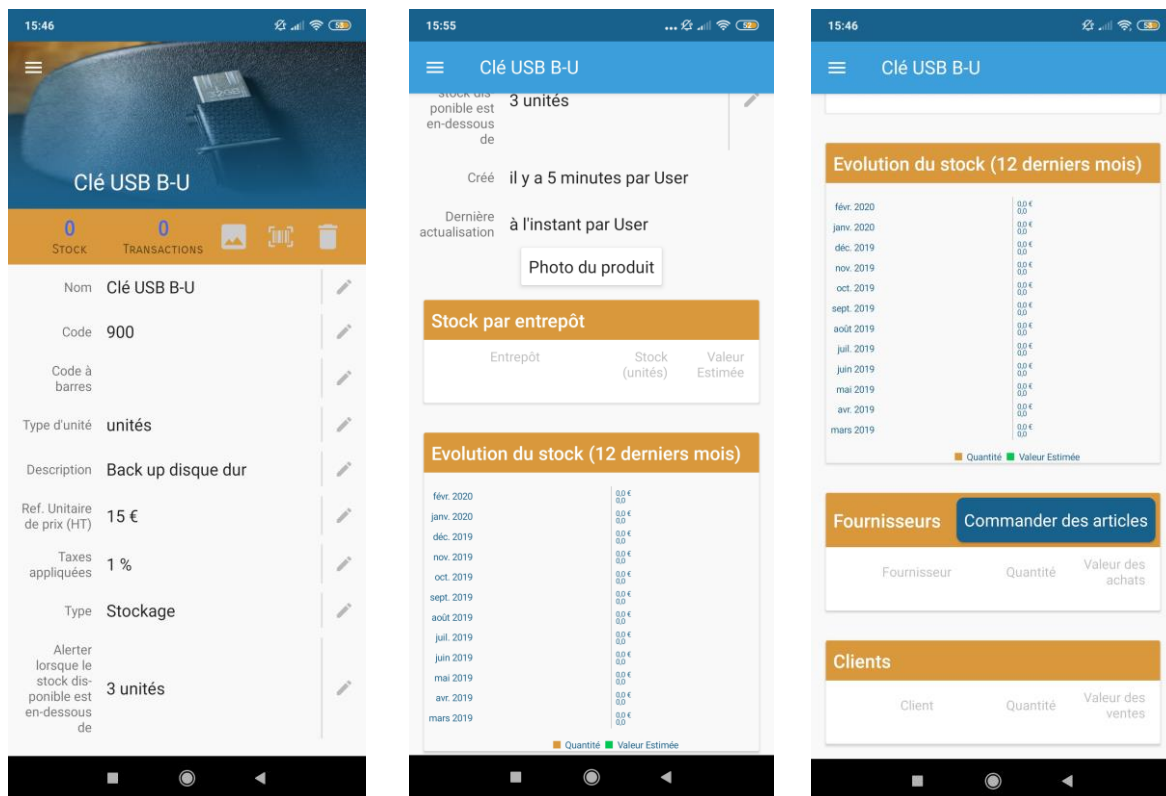
II - État de l'art :

1. Stock Controller :



Cette application disponible gratuitement (payante pour obtenir certaines fonctionnalités dont le prix varie entre 2€99 et 19€99) propose une gestion de stocks en ligne et hors ligne à l'aide d'un compte internet (email). Elle permet de créer plusieurs entrepôts dans lesquels les produits sont stockés par leur type.

Les images ci-dessous illustrent l'ajout d'un produit dans un stock : on peut associer à ce produit une photo et son code-barre, puis on le place dans l'entrepôt adéquat. On peut ensuite suivre son évolution ; voir ses différents fournisseurs et acheteurs (qui sont à rentrer au préalable dans l'application).



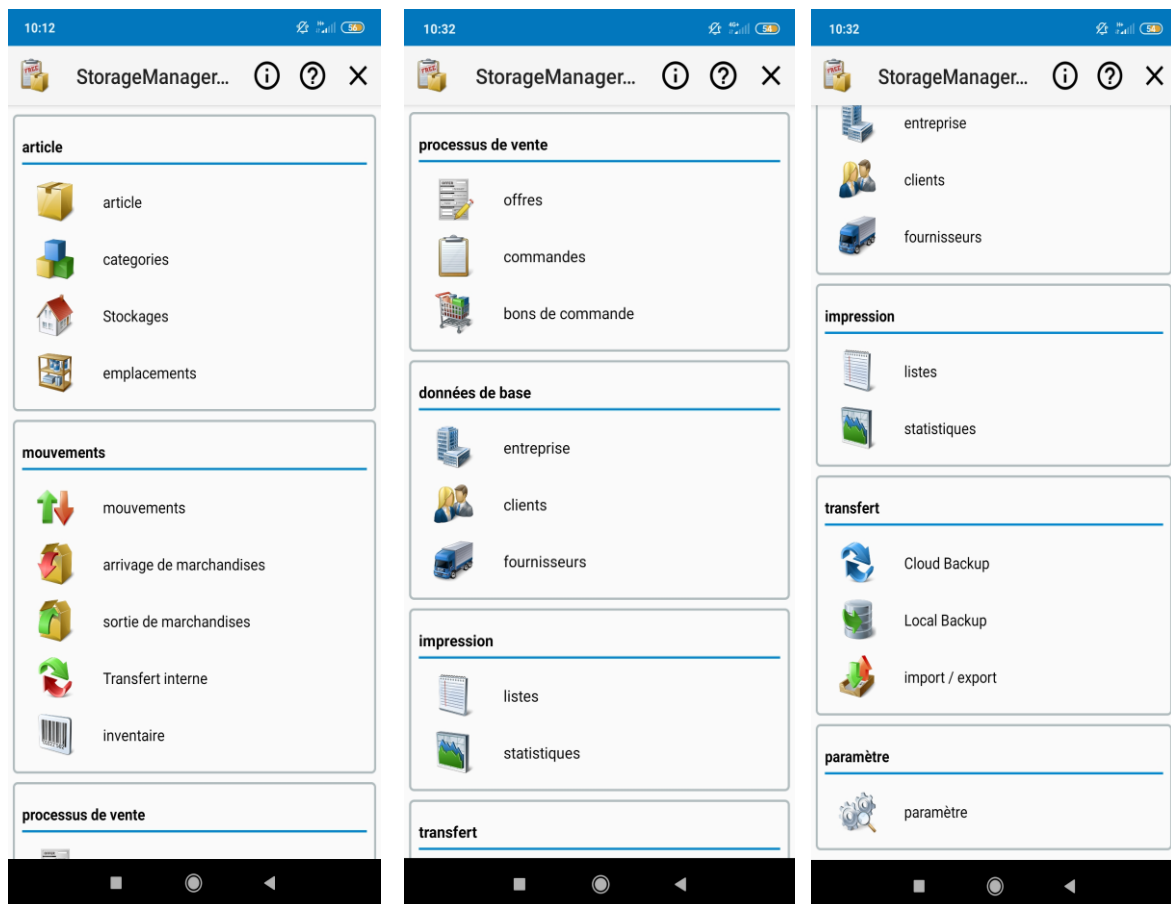
De plus, elle offre un panel de fonctionnalités importants, on peut analyser les transactions et les transferts faits sur la semaine, le mois voire l'année. Les statistiques portent sur toutes les sections comme par exemple les ventes par client ou encore les achats par date.

Après exploration de l'application (d'une quinzaine de minutes), on constate une prise en main rapide mais les nombreuses options offertes peuvent rendre son utilisation assez complexe. Il est à noter que plusieurs personnes peuvent accéder à la gestion de stocks en même temps.

2. Storage-Manager Free :



Cette application offre la plupart des options classiques d'application de stockage (ajout d'un produit dans un entrepôt particulier par exemple). Cependant, si l'on se réfère aux commentaires des utilisateurs, il semblerait qu'il y ait un manque de fonctionnalités pour les statistiques et le suivi des produits. Le référencement des produits peut se faire par code-barre ou avec un code QR généré automatiquement.



Il est à noter que l'application ne peut pas être utilisée à plusieurs en même temps.

3. MonStock :



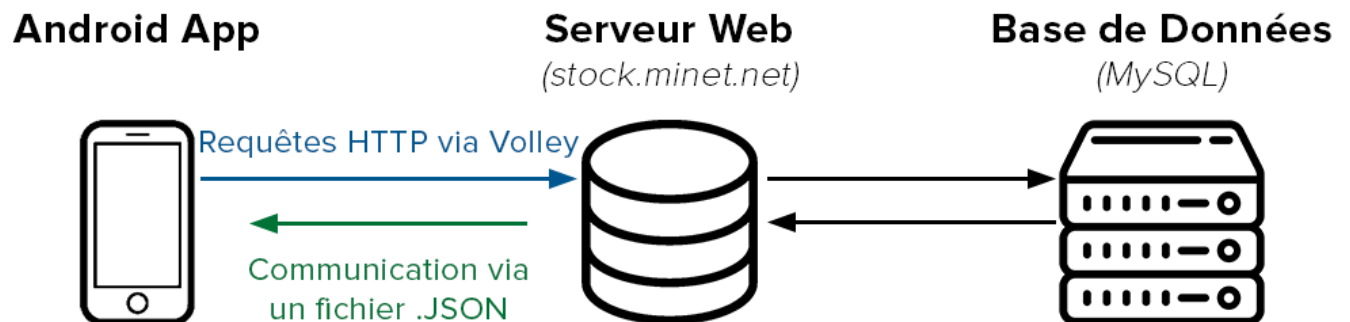
Cette application est disponible gratuitement sur Android et en abonnement sur ordinateur (la photo ci-dessous montre le prix assez élevé de ses différents abonnements qui découle de sa grande utilisation par les professionnels). Elle peut s'utiliser en ligne et hors ligne avec un compte, en même temps par plusieurs utilisateurs avec en plus la possibilité de gérer les droits d'accès à l'application (par exemple certains utilisateurs ne peuvent accéder qu'à certains entrepôts). Cependant, il faut un certain temps de pratique avant de bien prendre en main l'application.

	19.90	32.90	51.90
	€HT / 1 ^{er}	€HT / 1 ^{er}	€HT / 1 ^{er}
Abonnement mensuel	utilisateur	utilisateur	utilisateur
	-----	-----	-----
	+ 12.90 €HT	+ 25.90 €HT	+ 38.90 €HT
	/ utilisateur	/ utilisateur	/ utilisateur
Total annuel	178,80 €HT	298,80 €HT	478,80 €HT



III - Analyse fonctionnelle :

1. Principe de fonctionnement :



Nous utiliserons la librairie Android Volley (<https://developer.android.com/training/volley/index.html>) pour effectuer les requêtes au serveur.

2. Description des fonctions :

Fonction	Ajout de matériel dans la base de données.
Objectif	Permettre à l'utilisateur d'ajouter du contenu à la base de données de façon simple. L'ajout peut se faire de différentes façons. Soit par scan du

	code barre soit par ajout manuel de la référence.
Contrainte	Le scan et la reconnaissance du code barre doit être rapide.

Fonction	Retirer du matériel de la base de données.
Objectif	Permettre à l'utilisateur de retirer du contenu à la base de données de façon simple. La suppression d'objet peut se faire de différentes façons. Soit par scan du code barre soit manuellement via la référence de l'objet.
Contrainte	Le scan et la reconnaissance du code barre doit être rapide.

Fonction	Afficher la base de données.
Objectif	Permettre à l'utilisateur d'afficher le contenu de la base de données.
Contrainte	L'affichage doit se faire rapidement et de façon esthétique, notamment pour les bases de données larges.

Fonction	Recherche dans la base de données.
Objectif	Permettre à l'utilisateur de rechercher du contenu dans la base de données grâce aux différents champs.
Contrainte	La recherche doit être assez rapide.

3. Description des classes :

Nom de classe	abstract Item
Champs	Int <i>ref</i> (définit la référence de l'objet, celle-ci est unique et est un entier). String <i>date</i> (Date de création de l'objet). String <i>hour</i> (Heure d'ajout de l'objet). String <i>desc</i> (Description facultative de l'objet). String <i>position</i> (Position de l'objet sous forme de chaîne de caractère libre). String <i>borrowedTo</i> (Chaîne de caractère correspondant à la personne à qui l'objet a été prêté) Int <i>State</i> (état de l'objet : 0 si retiré du stock, 1 si présent dans le stock et 2 si prêté).
Méthodes	getType() et getText()
Description	Une classe fille de cette classe définit un type d'objet et une instance d'une classe fille de cette classe définit un objet de manière unique (notamment grâce à la référence unique).

Nom de classe	Cable extends Item
Champs	Double <i>length</i> String <i>connector</i>
Méthodes	Constructeur Cable prenant en paramètre la connectique, la longueur, la référence, l'heure/date d'ajout, l'utilisateur ayant ajouté l'objet, une description et une position.
Description	Cette classe définit la catégorie câble.

Nom de classe	Ecran extends Item
Champs	Double <i>taille</i>
Méthodes	Constructeur Ecran prenant en paramètre la résolution, la taille, la référence, l'heure/date d'ajout, l'utilisateur ayant ajouté l'objet, une description et une position.
Description	Cette classe définit la catégorie écran.

Nom de classe	Adaptateur extends Item
Champs	String <i>entryport</i> (Type de port connectable en entrée) String <i>outputport</i> (Type de port connectable en sortie) Int <i>taille</i> (Taille de l'adaptateur en cm).
Méthodes	Constructeur Adaptateur prenant en paramètre : <ul style="list-style-type: none"> • Le port d'entrée / de sortie de l'adaptateur et sa taille en cm. • La référence, l'heure/date d'ajout, l'utilisateur ayant ajouté l'objet, une description et une position (redéfinition des paramètres du constructeur mère).
Description	Cette classe définit la catégorie adaptateur.

Nom de classe	Projecteur extends Item
Champs	String <i>model</i> (Type de modèle/technologie du projecteur) String <i>connectique</i> (Type de connectique du projecteur) Int <i>resolution</i> (Caractérise le degré de netteté de l'image en pixel)
Méthodes	Constructeur Projecteur prenant en paramètre : <ul style="list-style-type: none"> • Le modèle du projecteur, la connectique et la résolution du projecteur. • La référence, l'heure/date d'ajout, l'utilisateur ayant ajouté l'objet, une description et une position (redéfinition des paramètres du constructeur mère).
Description	Cette classe définit la catégorie projecteur.

Nom de classe	Utils
----------------------	-------

Champs	
Méthodes	testNetwork filterNameCount toDate filtrerDate filtreText actualiserHistory actualiserSwipeHistory actualiserSwipe actualiserClique postRequeteUpdate postRequeteAddCable postRequeteAddEcran
Description	Cette classe regroupe certaines méthodes étant utiles au reste de l'application.

Nom de fonction	testNetwork
Entrée	Error <i>error</i> (l'erreur fournie par le module qui s'occupe de faire la liaison entre l'application et l'accès au serveur web)
Sortie	Int <i>errorResult</i> (en fonction de la valeur de l'erreur on peut fournir une explication claire à l'utilisateur de quel type d'erreur il s'agit)
Description	Permet de tester la connexion à la bdd et renvoi une valeur spécifique selon l'erreur ou 0 si tout s'est bien déroulé

Nom de fonction	filterNameCount
Entrée	Collection<Item> <i>array</i> (une collection d'Item)
Sortie	Collection<String, Int> <i>map</i> (une collection qui fait correspondre à chaque objet partageant les mêmes caractéristiques principales une quantité)
Description	Permet de grouper les objets par caractéristique principale (via la méthode getText() des classes héritant d'Item) et de les compter

Nom de fonction	toDate
Entrée	String <i>d</i>
Sortie	Date <i>date</i>
Description	Convertit une date de la forme "yyyy-MM-dd" en objet date

Nom de fonction	filtrerDate
Entrée	Date <i>d</i> , Collection<Item> <i>array</i> (une date au format “yyyy-MM-dd” et une collection d’Item)
Sortie	Collection<Item> <i>array</i> (une collection d’objet)
Description	Permet de filtrer une collection d’objet par date

Nom de fonction	filtrerText
Entrée	String <i>text</i> , Collection<String,Int> <i>array</i> (une chaîne de caractère selon laquelle on veut filtrer la collection)
Sortie	Collection<String,Int> <i>arrayFinal</i> (la collection correspondant au filtre donné en paramètre)
Description	Permet de filtrer une collection d’objets par la chaîne de caractère « <i>text</i> »

Nom de fonction	actualiserHistory
Entrée	String <i>date</i> , Collection<item> <i>mesItems</i> , String <i>urlserver</i>
Sortie	
Description	Permet de charger la page History depuis la base de donnée se situant sur « <i>urlserver</i> » et ce à partir de la date « <i>date</i> »

Nom de fonction	actualiserSwipeHistory
Entrée	String <i>date</i> , Collection<item> <i>mesItems</i> , SwipeObject <i>swipeContainer</i> , String <i>urlserver</i> (SwipeObject est un objet générique désignant l’interface gérant le swipe afin de rafraîchir la page)
Sortie	
Description	Permet d’actualiser la page History en utilisant le swipe du SwipeObject (en récupérant les objet dans la bdd et filtrant à partir de la date <i>date</i>)

Nom de fonction	actualiserClique
------------------------	------------------

Entrée	Collection<item> <i>mesItems</i> , String <i>urlserver</i>
Sortie	
Description	Permet de charger la page Stock en récupérant les données sur le serveur (à partir d' « urlserver »)

Nom de fonction	actualiserSwipe
Entrée	Collection<item> <i>mesItems</i> , SwipeRefreshLayout <i>swipeContainer</i> , String <i>urlserver</i> , String <i>text</i> (SwipeObject est un objet générique désignant l'interface gérant le swipe afin de rafraîchir la page)
Sortie	
Description	Permet d'actualiser la page Stock par le swipe. Permet également de filtrer les objets contenant <i>text</i>

Nom de fonction	postRequeteUpdate
Entrée	String <i>ref</i> , String <i>url</i> , String <i>typeRequest</i>
Sortie	
Description	Permet de mettre à jour la bdd en envoyant un type de requête correspondant à 0 pour supprimer l'objet, 1 pour le prêter, 2 pour retourner l'objet et ce pour l'objet correspondant à l'unique référence « ref »

Nom de fonction	postRequeteAddCable
Entrée	String <i>ref</i> , String <i>url</i> , String <i>length</i> , String <i>connector</i> , String <i>position</i>
Sortie	
Description	Permet d'ajouter un objet de type câble dans le BDD

Nom de fonction	postRequeteAddEcran
Entrée	String <i>url</i> , String <i>ref</i> , String <i>length</i> , String <i>resx</i> , String <i>resy</i> , String <i>position</i> , String <i>desc</i>
Sortie	
Description	Permet d'ajouter un objet de type écran

IV – Maquettes :

HomeAdapter

Maquette initiale :

Rechercher

Lettre A :

ADAPTEUR USB-RJ45

x20

Lettre C :

CABLE RJ45
LONGUEUR : 3M

x45

CABLE RJ45
LONGUEUR : 10M

x30

Lettre S :

SWITCH

x3

Maquette finale :

Stocks

QTE

OBJET

1

CABLE HDMI

1

ECRAN 55.0

1

ECRAN 558.0

1

ECRAN 224.0

2

ECRAN 5.0

1

ECRAN 245.0

2

ECRAN 12.0

1

ECRAN 1.0

2

PROJECTEUR HDMI

2

PROJECTEUR DD

1

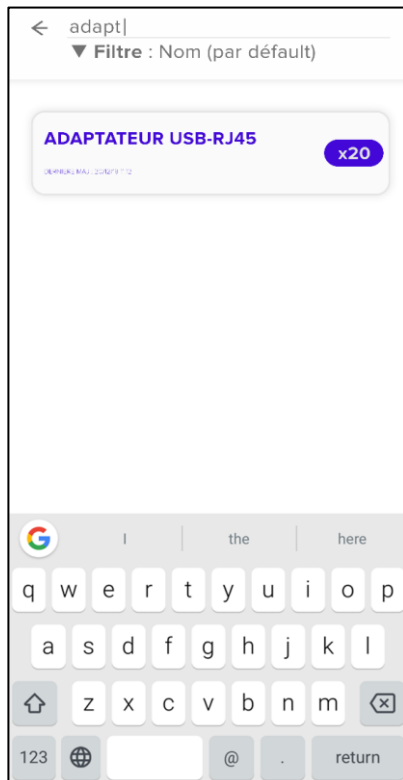
PROJECTEUR GSXS

Stocks

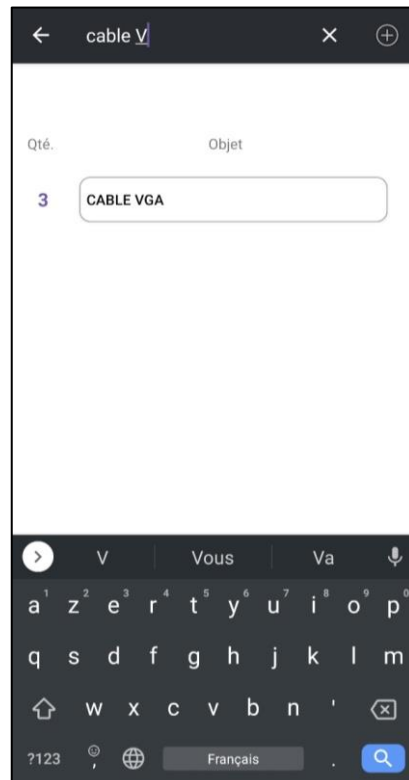
Historique

HomeFragment

Maquette initiale :



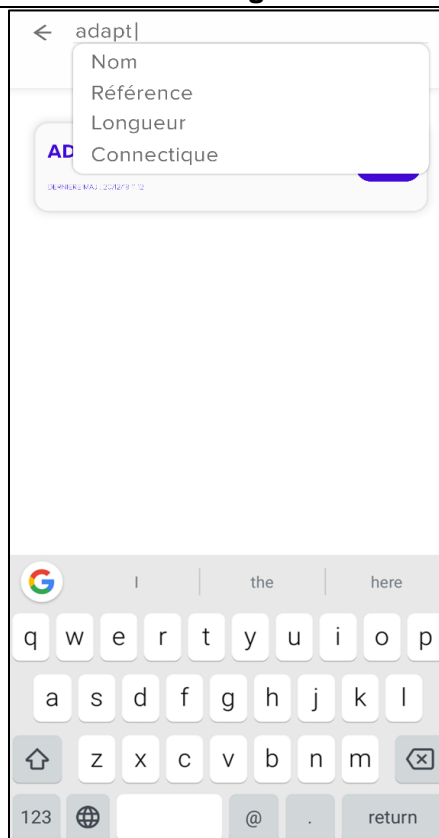
Maquette finale :



En appuyant sur le champ textuel de recherche situé en haut de la page précédente on arrive sur cette page qui correspond à la méthode HomeFragment (le filtre Nom activé par défaut n'a pas été implémenté finalement par manque de temps).

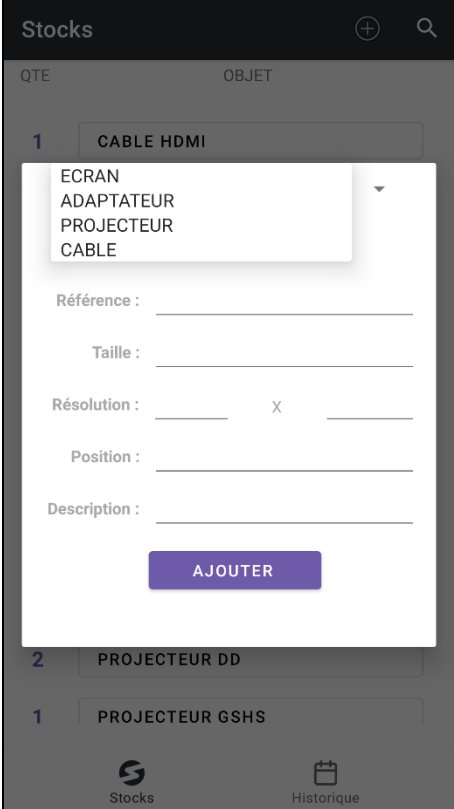
Il y a la possibilité d'en sélectionner un autre (non implémenté) :

HomeFragment



Ici on peut voir comment fonctionnerait la sélection de filtre parmi les caractéristiques disponibles.

AddDialogFragment



The screenshot shows a mobile application interface titled "Stocks". At the top, there is a header bar with a "+" icon and a search icon. Below the header, there is a table with two columns: "QTE" and "OBJET". The table contains three rows of data:

QTE	OBJET
1	CABLE HDMI
2	PROJECTEUR DD
1	PROJECTEUR GSHS

Below the table, there is a dialog box titled "AddDialogFragment". The dialog has a dropdown menu for selecting the object type, with options: ECRAN, ADAPTATEUR, PROJECTEUR, and CABLE. Below the dropdown, there are input fields for the following characteristics:

- Référence :
- Taille :
- Résolution : (with a unit "X")
- Position :
- Description :

At the bottom of the dialog, there is a purple button labeled "AJOUTER".

A partir de la page d'accueil "Stocks" ou "Historique", nous pouvons ajouter un élément à la base de données en cliquant sur l'icône + en haut à gauche dans la barre d'outils.

Il faut ensuite choisir le type de l'objet (ici, on a pour l'instant écran, adaptateur, projecteur et cable) et renseigner ses caractéristiques.

Enfin, on clique sur le bouton ajouter qui rend effectif (si tous les champs sont bien remplis) l'ajout de l'objet dans la base de données.

HistoryAdapter

Historique
+

Cable VGA date ajout: 2020-06-07
Date mise à jour: 2020-06-07

Cable HDMI date ajout: 2020-06-07
Date mise à jour: 2020-06-07

Ecran 55.0 date ajout: 2020-06-05
Date mise à jour: 2020-06-07

Ecran 55.0 date ajout: 2020-06-05
Date mise à jour: 2020-06-07

Ecran 558.0 date ajout: 2020-06-06
Date mise à jour: 2020-06-07

Ecran 224.0 date ajout: 2020-06-06
Date mise à jour: 2020-06-07


Ecran 5.0 date ajout: 2020-06-06
Date mise à jour: 2020-06-07


Ecran 5.0 date ajout: 2020-06-06
Date mise à jour: 2020-06-07

A partir de

2020-06-01

(i)


 Stocks


 Historique

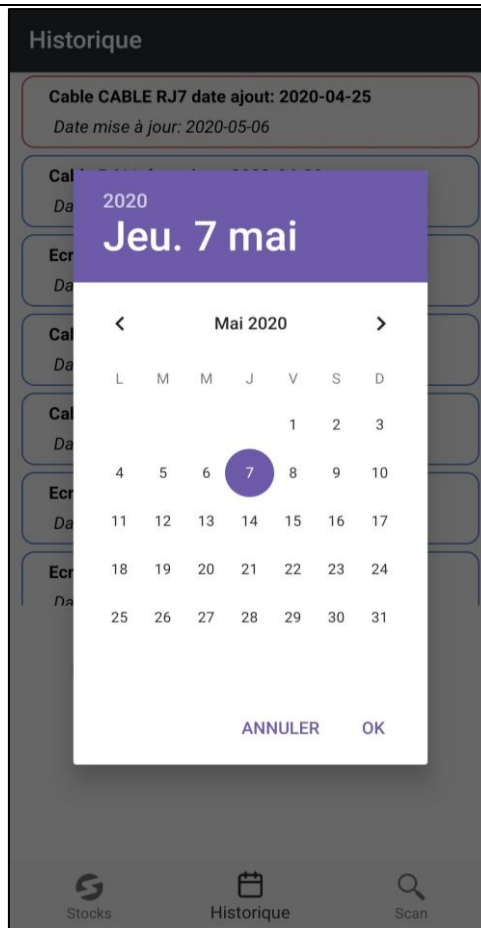
A partir de la page d'accueil, afin d'accéder à la page historique on appuie sur le bouton historique qui apparaît en bas de l'écran dans une barre d'outils.

De plus, l'écran ci-dessus est une représentation de la classe HistoryAdapter avec la date limite par défaut.

Enfin, la distinction entre objet présent, prêté et supprimé de la base de données est faite par un code couleur sur la couleur du contour du bouton :

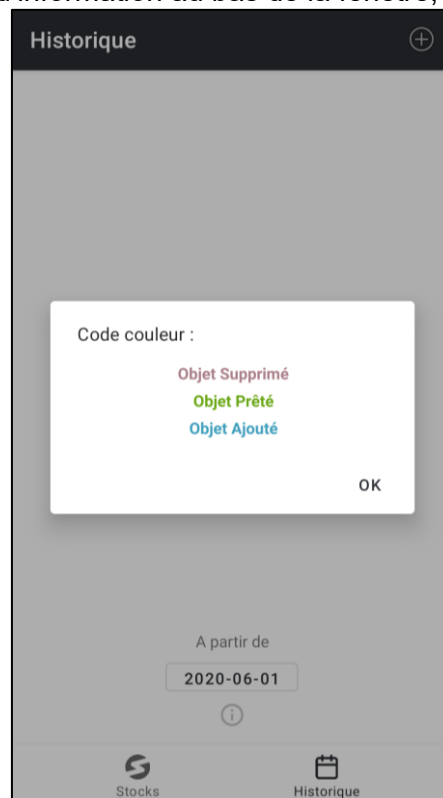
- Bleu correspond à un ajout d'objet
- Vert correspond à un prêt d'objet
- Rose correspond à la suppression d'un objet

HistoryFragment

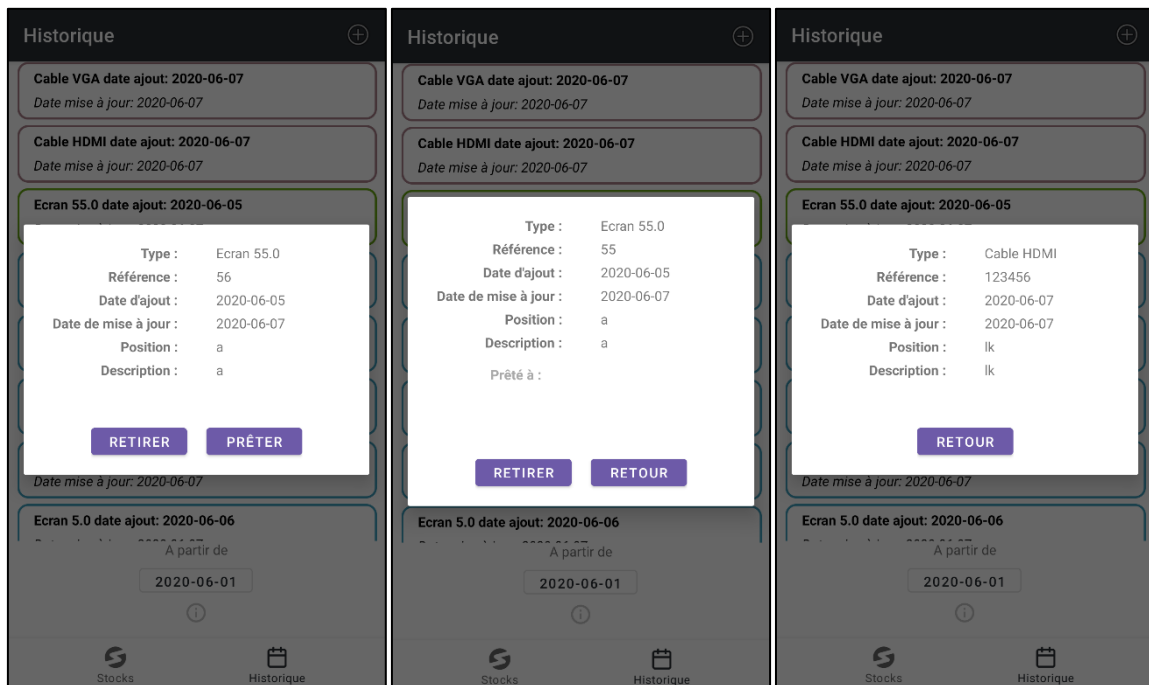


Cet écran représente la sélection de date de l'une des méthode de la classe HistoryFragment.

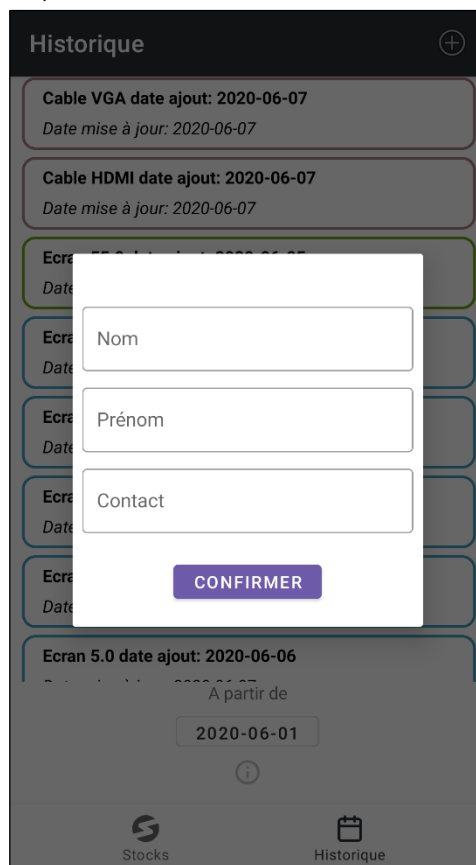
En appuyant sur le bouton d'information au bas de la fenêtre, le pop-up suivant apparaît :



UpdateDialogFragment



En cliquant sur un des boutons représentant un objet, il apparaît ce popup affichant les différentes caractéristiques de l'objet et offrant la possibilité de le prêter ou de le supprimer dans le cas où l'objet est présent dans le stock (gauche), de le retourner ou le retirer dans le cas où il a été prêté (milieu) ou de le retourner dans le cas où il a été supprimé (droite).



La fenêtre ci-dessus apparaît lorsque l'on souhaite prêter l'objet.

V – Backend

1. Mise en œuvre du serveur :

Ce texte a pour but de documenter la configuration du serveur dans le cadre du projet de développement informatique PRO3600_17.

Nous utilisons ici un serveur sous Debian.

- 1^{ère} étape : *Installer php et les dépendances requises.*
Dans le terminal, écrire : 'apt-get install php php-fpm php-mysql' en tant que 'root'.
- 2^{ème} étape : *Installation d'un serveur web*
Ici, nous utilisons le serveur web 'nginx' (<https://nginx.org/>). Pour installer ce serveur web, il faut entrer : 'apt install nginx' en tant que 'root' dans le terminal.

Il faut penser à ouvrir le port (les ports) sur lequel (lesquels) nginx écoute (en général le port 80 pour un serveur web ne possédant pas de certificat SSL et 443 pour un serveur en possédant un).

- 3^{ème} étape : *Installation du système de gestion de base de données MySQL*

Il faut installer le client et le serveur MySQL, pour cela taper 'apt install mysql-server mysql-client' dans le terminal.

Après avoir installé ces deux paquets, lancer 'mysql_secure_installation' et répondre aux questions posées :

- Set root password ? :
Permet de modifier le mot de passe root (de mysql)
- Remove anonymous users ? :
Retire les accès anonymes
- Disallow root login remotely ? :
Retire l'accès root distant
- Remove test database and access to it ? :
Retire la base *test* et ses accès
- Reload privilege tables now ? :
Recharge les privilèges suites aux modifications

On crée alors une base de données pour notre serveur Web :

```
CREATE DATABASE Stock;
```

Il faut ensuite créer un utilisateur dans MySQL pour utiliser cette base de données :

```
CREATE USER 'StockUser'@'localhost' IDENTIFIED BY 'mot2passe';
```

Et lui donner tous les droits sur la base de données créée précédemment :

```
GRANT ALL PRIVILEGES ON Stock.* TO 'StockUser'@'localhost';
```

La configuration du serveur est terminée.

- 4^{ème} étape : *Installation de l'application web Laravel*

Après avoir récupéré l'application web dans le dépôt Git, la placer dans `/var/www/html/`. Il faut ensuite aller configurer le serveur Nginx. Pour cela rendez-vous dans `/etc/nginx/sites-available`. Et créez un nouveau fichier portant le nom de votre site web.

Dans ce fichier coller cela :

```
server {
    listen 80;    //Le port sur lequel vous voulez que votre serveur
web écoute, ne pas oublier de /
//l'ouvrir dans le par feu

    server_name 192.168.xxx.xxx //votre adresse ip;
    root /var/www/html/nomDeVotreAppLaravel/public;

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-XSS-Protection "1; mode=block";
    add_header X-Content-Type-Options "nosniff";

    index index.html index.htm index.php;

    charset utf-8;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location = /favicon.ico { access_log off; log_not_found off; }
    location = /robots.txt  { access_log off; log_not_found off; }

    error_page 404 /index.php;

    location ~ \.php$ {
        fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME
$realpath_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\. (?!well-known).* {
        deny all;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

2. Description de la base de données :

La Base de Données (BD) est pour le moment constituée de 6 tables : cables, ecrans, adaptateurs, projecteurs et migrations mais seules les quatres premières vont importer dans cette description.

Le schéma relationnel (en ne tenant compte que de ces deux tables) de cette BD est :

cables (id, state, type, connector, borrowed_to, dateAdd, dateUpdate, length, position, description)
ecrans (id, state, type, width, resolution, borrowed_to, dateAdd, dateUpdate, position, description)
adaptateurs (id, state, type, entryport, outputport, length, borrowed_to, dateAdd, dateUpdate, position, description)
projecteurs (id, state, type, resolution, borrowed_to, dateAdd, dateUpdate, position, description)

Les clés primaires étant soulignées.

La table *cables* possède ainsi comme clé primaire l'attribut *id* (choix classique) qui est un entier strictement positif permettant d'identifier chaque élément de cette table de manière unique.

Ensuite l'attribut *state* est un entier qui prend soit la valeur 0, 1 ou 2 signifiant respectivement que l'objet est retiré, prêté ou stocké.

Il suit juste après les attributs *type* et *connector* qui sont des chaînes de caractères désignant respectivement le type de l'objet (cependant, étant donné que le type de l'objet dans cette table sera toujours un câble, on pourra raisonnablement penser à retirer cette attribut) et la catégorie dans laquelle il entre (HDMI, VGA ...).

Après, l'attribut *borrowed_to* est une chaîne de caractères indiquant à qui est prêté l'objet (bien sûr seulement si l'objet est prêté).

Il vient ensuite les attributs *dateAdd* et *dateUpdate* qui sont tous les deux de type string indiquant respectivement la date d'ajout et la date de mise à jour de l'objet dans la table.

Puis l'attribut *length* est un entier positif explicitant la longueur en cm du câble.

Après celui-ci, l'attribut *position* est une chaîne de caractères spécifiant le lieu où est entreposé l'objet.

Enfin, l'attribut *description* est une chaîne de caractères explicitant la condition de l'objet (s'il est en bon état, passable, très abîmé ...).

La table *ecrans* comporte les attributs *id*, *state*, *type*, *dateAdd*, *dateUpdate*, *position* et *description* qui jouent le même rôle que dans la table *cables*.

Elle possède de plus l'attribut *width* de type entier positif déterminant la largeur de l'écran en cm (cependant, étant donné que la taille des écrans est caractérisée par la longueur de sa diagonale en pouce, on pourrait modifier le nom de l'attribut par "*Screen size*" et explicitant des pouces afin qu'il désigne cette caractérisation).

La table *adaptateurs* comporte aussi les attributs *id*, *state*, *type*, *dateAdd*, *dateUpdate*, *position* et *description* qui jouent le même rôle que dans la table *cables*.

Elle possède de plus les attributs *entryport*, *outputport* et *length* qui correspondent respectivement à une chaîne de caractères désignant le type de port connectable en entrée, le type de port connectable en sortie et un entier indiquant la taille de l'adaptateur en cm.

La table *projecteurs* comporte aussi les attributs *id*, *state*, *type*, *dateAdd*, *dateUpdate*, *position* et *description* qui jouent le même rôle que dans la table *cables*.

Elle possède de plus l'attribut *resolution* qui est un double indiquant le nombre de pixels par unité de longueur du projecteur.

VI – Conclusion :

1. Perspectives

Le projet étant riche en fonctionnalités prévues, nous n'avons pas eu le temps de tout implémenter. Voici donc quelques fonctionnalités qu'il serait intéressant d'intégrer.

Premièrement, l'application, en l'état actuel, est très limitée étant donné que seule certaines catégories d'objets sont prises en charge. Il serait donc intéressant de créer un système pour avoir une application plus flexible et adaptable à toute sorte de stock. Il serait également judicieux d'ajouter la possibilité de modifier des objets déjà ajoutés.

D'autre part, il faudrait rendre l'application utilisable par n'importe quel serveur. En effet, pour l'instant, on ne peut l'utiliser qu'avec stock.minet.net. De plus, le site n'est pas adapté au design de l'application pour le moment, il faudrait redesigner le site avec la charte graphique de l'application. En plus de cela, la création d'un installateur côté serveur semble une bonne idée étant donné que la configuration de celui-ci peut être ardue pour des personnes qui n'ont jamais touchées aux systèmes Unix.

Finalement, l'ajout des fonctionnalités secondaires (et que nous n'avons pas eu le temps d'implémenter) semble nécessaire. A savoir, un système de login et plus de filtres pour les objets par exemple.

2. Conclusion générale

Ce projet informatique avait pour but de développer une application fonctionnelle et ergonomique de gestion de stock. Après avoir fait l'état de l'art des applications disponibles en ligne, nous avons fait les maquettes de notre application. Nous avons choisi de coder cette application en Kotlin, un langage dérivé du Java et adapté au développement mobile. Le développement de cette application a été réalisé sur Android studio. Le projet a été mené à bien, l'application fonctionne, toutefois nous aurions voulu développer plusieurs fonctionnalités secondaires dont un scan par QR Code que nous n'avons pas pu implémenter dû à la limite de temps qui nous était impartie.

Au niveau du travail de groupe, le projet s'est bien déroulé avec une bonne entente des 4 membres. La répartition des tâches a été efficace et le temps total du travail a été respecté par les 4 membres. De plus le niveau élevé de 2 élèves du groupe ont permis aux 2 autres de se former et d'en apprendre plus grâce à leurs connaissances. Il était intéressant de travailler avec des élèves de niveau différents car cela permet un vrai partage de connaissance qui est très important dans la gestion d'un projet.

3. Conclusions personnelles

a. Vincent LABARRE :

J'avais choisi de participer à ce projet car il allait servir concrètement pour les associations telles que MiNET ou BPM, ce qui me donnait envie de m'investir dedans. En plus, nous allions apprendre le langage Kotlin et la programmation sur Android Studio.

Ce projet m'a permis de prendre conscience de la difficulté de créer une application : cela demande un travail très long, laborieux et très rigoureux pour mettre en place une application. J'ai alors pu me rendre compte que cela ne me plairait pas spécialement de faire cela plus tard dans mon métier. Cependant, j'ai aimé le fait d'avoir pu apprendre des choses dans la

programmation, que ce soit avec Kotlin ou avec le langage xml. Un de mes regrets pour ce projet est de ne pas avoir travaillé sur la partie web.

Je pense que nous nous en sommes pas mal sortis, mais que nous aurions pu travailler plus ensemble, surtout en début de projet pour établir le cahier des charges par exemple (après cela aurait été compliqué avec le confinement). Un des enseignements que je tire aussi de ce projet est l'importance de la communication, qui nous a fait défaut quelque fois.

Je tiens à remercier particulièrement Mohamed pour m'avoir parlé de ce projet et m'avoir accepté dans le groupe, ainsi que notre tuteur Monsieur Belaid pour sa pédagogie et sa bienveillance.

Finalement, je pense que ce serait une bonne idée de continuer à développer cette application avec un futur groupe de Développement Informatique.

b. Valentin BAILLEUL :

Ce projet a été très intéressant pour ma part car je n'avais jamais conçu une application de A à Z, j'ai pu y découvrir un nouveau langage et les méthodes de développement d'une application. Le travail en groupe a été très intéressant car les autres membres du groupe non pas hésité une seule seconde à partager leurs connaissances dans ce domaine ce qui m'a permis d'en apprendre plus sur le développement d'applications et sur l'informatique, les réseaux... en général.

De plus grâce à leurs aides j'ai réussi à résoudre quelques difficultés que je possédais en début de projet. Mon travail était plus axé sur le design de l'application et son ergonomie. J'ai apprécié de réfléchir sur l'ergonomie de l'application. Au niveau de la gestion du projet l'entente a été très bonne durant tout le projet.

De plus les réunions avec notre tuteur M. Belaid nous ont permis de garder une bonne ligne de conduite sur le développement de l'application, je le remercie pour tous ses conseils. J'espère que le projet sera continué pour finir de développer le scan de QR code et finaliser totalement l'application.

c. Mohamed CHAMROUK :

En conclusion de ce projet de développement informatique, j'exprime le sentiment que ce projet a su se montrer très instructif.

Tout d'abord pour son aspect de projet de groupe et le contexte qui lui y est associé. Plus qu'un simple projet de développement d'application Android, ce projet m'a montré les différents aspects du travail de groupe et qu'est-ce qu'avoir un supérieur signifie dans un projet comme celui-ci. Cela m'a permis de mieux organiser mon travail, d'être plus productif et aussi l'importance de la communication entre les membres du groupe et avec le supérieur, ici le tuteur Mr. BELAID.

Bien sûr, l'aspect projet de groupe n'existerait pas si ce n'est pour le projet qui s'est développé derrière. Le développement d'application Android était une compétence que je voulais acquérir depuis longtemps mais pour laquelle je n'avais jamais eu l'occasion sérieuse de m'y mettre jusqu'à ce projet. J'estime avoir atteint l'objectif que je m'étais fixé voire plus grâce à la communication de notre application avec un serveur web et toutes les librairies que cela nécessite.

En conclusion, ce projet était l'un des plus instructif dans lequel j'ai pris part. Que ce soit d'un point de vue de gestion de projet ou de développement informatique, ce projet a su se montrer utile sous tous les aspects.

d. Dzenan CINDRAK :

Le projet m'a beaucoup appris, tant au niveau technique qu'au niveau gestion de projet.

En effet, je n'avais jamais fait de projet concret sous Android studio. Cela m'a permis d'atteindre un niveau intermédiaire en Kotlin et d'apprendre de nouveaux concepts comme le modèle REST. D'autres part, j'ai pu approfondir mes connaissances dans le framework Laravel utilisé pour les développements web. J'ai également pris connaissance du fait que certains outils (comme Stackoverflow, Git ou Kdoc) sont quasiment indispensables pour réaliser un projet informatique.

En plus de cela, j'ai pu prendre part à une 'vraie' situation de gestion de projet informatique. En effet, cela m'a appris beaucoup de choses, notamment que la gestion du temps est fondamentale, spécialement quand on travaille sur deux projets en même temps étant donné que je participais aussi au développement du site web pour mon projet GATE. De plus, la communication est indispensable à une bonne cohésion et efficacité au sein du groupe projet. Je pense d'ailleurs qu'une de notre plus grosse erreur était la faible communication au sein du groupe.

Finalement, je trouve que c'était une expérience très enrichissante dans toutes ses dimensions et j'espère que le projet sera approfondi par les prochaines générations grâce à notre rapport qui est, selon moi, assez détaillé pour que n'importe qui puisse comprendre le projet et le continuer.

VII - Annexes :

Annexe 1 : Compte-rendu des réunions :

1. Compte Rendu du 28/01 :

Motif/type de réunion : Suivi Mensuel	Lieu : Télécom SudParis
Présent(s) : tous les membres	Date et durée : 28/01 à 17h, 40mins

Ordre du jour : Préciser la définition du projet et établir les premiers objectifs.

Informations échangées :

Premier échange :

- Tour de table :
 - Application : Base de données
 - Description de l'inventaire (on parle plutôt ici de gestion de stock)

- avec une utilisation simple, pratique, intuitive => utile pour les associations du campus
- exemple : Minet.
- Tuteur :
 - Objectifs du Développement Informatique :
 - Consolider les acquis/non-acquis du cours
 - Progresser ensemble
 - Aspects importants :
 - aspect développement
 - aspect gestion de projet
 - 45 à 60 heures de travail par personne
 - Planning prévisionnel :
 - 4 colonnes pour chaque membre
 - date et durée de travail
 - lignes communes : nombreuses au début et fin
 - joindre le planning prévisionnel avec le suivi d'activité
 - Cahier des charges :
 - Préciser le périmètre primaire et secondaire de que l'on veut développer
 - liste des fonctionnalités
 - limiter à Android
 - Maquettes :
 - voir "l'état de l'art" : que fait la concurrence ?
 - demander aux futurs utilisateurs concernés leurs retours
 - Aspect fonctionnel :
 - Analyse et développement avec des jalons intermédiaires
 - Distinction Inventaire et gestion de stock :
 - L'inventaire concerne la différence du calcul des produit attendus avec ceux présents
 - Précisions sur certaines fonctionnalités de l'application :
 - La contrainte ne doit pas être technique, il peut cependant y avoir des contraintes fonctionnelles
 - Base de donnée sur un serveur
 - Formation :
 - Se mettre au même niveau entre tous les membres de l'équipe
 - Utilisation du GIT

Questions à se poser :

- Est-ce que l'interface graphique établie est adaptée à la demande du client ?
- Quelles sont les caractéristiques qu'auront l'application ?
- Que veut-on précisément faire avec l'application ?
- Comment allons-nous construire ces caractéristiques ?
- Pourquoi nécessairement une petite structure ?

Décisions :

- Engagement complet dans le projet et donc avec un travail régulier
- On considère que l'on aura toujours une connexion internet
- Le langage de programmation sera défini ultérieurement

Actions à suivre / Todo list

Description	Délai	Livrable
Formation Android	2 Mois	Compétences de

		développement
Planning prévisionnel des tâches de chacun et du temps de travail	1 Semaine	Tableau résumé du planning
Premier dossier complet de description du projet	1 à 2 Semaines	Cahier des charges
Images décrivant le design de l'application	1 à 2 Semaines	Maquettes

2. Compte Rendu du 04/02 :

Motif/type de réunion : Suivi	Lieu : Télécom SudParis
Présent(s) : CHAMROUK Mohamed, LABARRE Vincent, BAILLEUL Valentin, CINDRAK Dzenan	Date et durée : 28/01 à 17h45, 50 min

Ordre du jour :

- Revue du livrable précédent
- Fixer les objectifs pour les 2 prochaines semaines.

Informations échangées :

Livrables rendus :

- Planning :
 - Modification à faire directement sur "Partage" en attendant le cours sur GIT
 - Améliorations possibles :
 - Ajouter une colonne : "la description de l'activité"
 - Ajouter deux colonnes : "Ce qu'on prévoit de faire" et "Ce qu'on a fait"
 - Un fichier par membre du groupe.
Nom du fichier = Planning-Nom_Prenom.xlsx
- Cahier des Charges (CdC) :
 - ~~Type de l'objet~~ Catégorie
 - Caractérisation des objets :
 - Nom
 - Description
 - Liste de valeurs
 - ~~Référence~~ Identifiant unique :
 - Chaîne de caractères libres
 - Emplacement de l'objet :
 - Chaîne de caractères libres
 - Définir ce qu'est un utilisateur
 - Fonctionnalité doit :
 - Avoir un nom

- Donner des entrées
- Ajouter peut signifier :
 - Obtenir un nouvel objet
 - Récupérer un objet emprunté
- Même problème pour le terme “retirer” :
 - Nécessité d'utiliser des termes précis
- Rapidité n'est pas une contrainte spécifique mais générale !
- Critère d'affichage de la base de données :
 - Par catégorie ?
 - Par nombre d'objets ?
- Présentation à étoffer
- Remarques générales :
 - Faire très attention aux fautes d'orthographe !
 - Le CdC est le premier chapitre du rapport
 - Cahier des charges clair permet de rapidement développer le code

Actions à suivre / Todo list

Description	Délai	Livrable
État de l'art	Avant le 10 février	Rapport (version courante datée)
Maquette et description de chaque fonctionnalité de l'application	Avant le 16 février 20h00	Nouvelle version du rapport datée
Suivre le MOOC : “Développez une application pour Android” et faire des recherches sur les langages de programmation possibles à employer pour éventuellement faire un seul développement pour android et iOS	1 semaine (pendant la semaine du 17 février)	Ajouter au rapport en annexe un CR succinct d'auto-formation

3. Compte Rendu du 25/02 :

Motif/type de réunion : Suivi	Lieu : Télécom SudParis
Présent(s) : CHAMROUK Mohamed, LABARRE Vincent, BAILLEUL Valentin, CINDRAK Dzenan	Date et durée : 25/02 à 18h05, 1h10

Ordre du jour :

- Revue du livrable précédent.
- Préciser les objectifs les plus immédiats pour le début du développement.

Informations échangées :

- Remarques sur la gestion de projet :

- Créer le compte GIT pour le projet (sur le GitLab).
- Mettre à jour les plannings en ajoutant la colonne “activité réalisée” et en corrigeant les coquilles présentes dans les dates.
- Remarques sur la forme du rapport :
 - Ne pas oublier de valider les CR (vérifier les modifications faites par le tuteur et les valider) et de les ajouter en annexe du rapport.
 - Faire le sommaire automatiquement sur Google Doc ou Word.
 - Inverser l'ordre de l'état de l'art et du cahier des charges dans le rapport.
- Remarques sur le fond du rapport :
 - État de l'art : description des fonctionnalités offertes par les applications mentionnées qui nous intéressent : les descriptions doivent être indépendantes entre elles.
 - Analyse fonctionnelle = analyse des fonctions : on part du plus haut niveau de description (processus) vers le plus bas (l'architecture matérielle).
 - Faire attention au nommage de la classe générique de l'application (Object déjà utilisée).
 - Pour définir une nouvelle catégorie, on peut :
 - Restreindre la création d'une catégorie à l'administrateur (et donc l'utilisateur ne peut pas sortir de ce cadre).
 - Créer deux ou trois catégories de base, et d'autres qui en découleront.
 - Ajouter une fonctionnalité permettant de créer une catégorie.
 - Mettre au clair l'utilisation de User :
 - addUser doit être la fonction d'ajout d'un utilisateur.
 - Une fonction supplémentaire faisant le lien entre l'utilisateur et la base de données accessible par l'administrateur doit être mise en place.
 - En algorithmique, on utilise “vrai” ou “faux” pour savoir si une fonction effectue bien ce qu'elle est supposée faire, et ensuite on traduit ces termes dans le langage de programmation utilisé.
 - Le nom des maquettes doit correspondre au nom de la fonctionnalité associée.
 - Écran initial enregistré à l'utilisateur et un autre associé à ajouter un utilisateur.
 - Maquettes et fonctionnalités doivent être faites ensemble.
 - Définir clairement la fonctionnalité Historique :
 - Définir les fonctions et fonctionnalités.
 - Faire le lien entre les maquettes et les fonctionnalités.
 - Fonctionnalité stock = affichage de tout le stock.
 - Les maquettes peuvent être faites au crayon.
 - Conseils très importants :
 - Il doit y avoir une cohérence entre fonctionnalités et maquettes !
 - Avoir des écrans simplifiés pour éviter la complexité.
 - Dans le menu, on doit pouvoir accéder à **toutes** les fonctionnalités.
 - Fonctionnalité “Ajouter un produit à modifier” : avoir un écran de début de la fonctionnalité, puis on peut avoir derrière des écrans secondaires pour qu'au bout de ses écrans, la fonctionnalité puisse être assurée.
 - Stock : Affichage par défaut dans l'ordre alphabétique : il faut donc compléter la description de la fonction displayDataBase : “Par défaut, rangé par ordre alphabétique. Puis autre filtre : par date,”
 - Login et password ne sont pas utiles dans l'aspect fonctionnel, on les néglige pour l'instant (les conflits aussi).
 - Conseil pour la page d'accueil :

- En haut le logo et une barre de recherche (saisie de texte ou bouton permettant une recherche avancée), puis en bas une liste (par ordre alphabétique). Donc il y aura une fonction Search.

Actions à suivre / Todo list

Description	Délai	Livrable
Formation	Pour le 2 mars	Compte rendu (en annexe du rapport)
Modifications du rapport (état de l'art, cohérence entre description des fonctionnalités et maquettes)	Pour le 2 mars	Rapport mis à jour

4. Compte Rendu du 03/02 :

Motif/type de réunion : Suivi	Lieu : Télécom SudParis
Présent(s) : CHAMROUK Mohamed, LABARRE Vincent, BAILLEUL Valentin, CINDRAK Dzenan	Date et durée : 02/03 à 18h, 30min

Ordre du jour : Revue des modifications faites dans le rapport et planification des prochains livrables.

Remarques sur la gestion de projet :

- Cette réunion ne sera pas très productive car le groupe n'a pas informé le tuteur de la mise à jour du rapport pour discuter lors de cette réunion les aspects fonctionnelles, les maquettes et la cohérence d'ensemble.
- Ne pas mettre V2, V3 pour les nouvelles versions des documents mais plutôt les datés (comme pour les CR).
- Mettre le lien explicite du Git : https://gitlabens.imtbs-tsp.eu/dzenan.cindrak/pro3600_17-gestiondestocks
- Ajouter une bibliographie.

Remarques sur l'analyse fonctionnelle et les maquettes :

- Ne pas mettre les lettres en clair, trier uniquement par ordre alphabétique.
- Ce n'est pas un bouton mais un champ textuel pour la barre de recherche.
- Longueur et connectique : quand on tape "ada", on ajoute un menu (mais pour l'instant trop compliqué). Dans un premier temps, on affiche juste une liste des caractéristiques communes à tous les objets.
- Il faut distinguer l'interface graphique et l'objet java en mémoire :
 - Si la date est un string, alors il va falloir faire des tests pour vérifier que c'est une date (ce qui rend son utilisation complexe).

- Il faut plutôt créer un objet date ou utiliser un format déjà présent.
- Fonction History :
 - Affichage de l'historique avec une date par défaut.
 - Si objet (ici la date) = null, il faut faire un affichage par défaut.
- Mettre les écrans du plus récent au plus ancien sans mettre "aujourd'hui", "hier" ...
- Fonction Scan :
 - Le scan se base seulement sur le code-barre d'un objet déjà rentré dans le stock :
 - Chaque référence sera stockée dans une base de données.
 - Il serait plus simple de rentrer des codes QR.
 - Améliorer la présentation du scan.
- Mettre la vue du menu latéral gauche (oublié dans la version actuelle du rapport).
- Mettre **tout** les écrans par lequel doit passer l'utilisateur et l'écran final pour chaque méthode.

Actions à suivre / Todo list

Description	Délai	Livrable
Réalisation de l'interface graphique de la page d'accueil (avec le code)	pour le 23 mars	Code de l'aspect graphique de l'application
Réalisation de la base de données	pour le 23 mars	Code du "backend" de l'application

5. Compte Rendu du 23/03/2020 :

Motif/type de réunion : Suivi Mensuel	Lieu : Visioconférence
Présent(s) : CHAMROUK Mohamed, LABARRE Vincent, BAILLEUL Valentin, CINDRAK Dzenan	Date et durée : 23/03 à 18h, 40 min

Ordre du jour : Présentation du premier livrable : première page de l'application et backend

Informations échangées :

Aspect fonctionnel :

- Page boutons cliquable non définies dans les vues
- Ajouter la maquette et le texte qui va bien dans le rapport
- Documenter la partie WEB (bibliothèques utilisée ...) Plus mettre à disposition le code qui est sur le serveur minet.
- Backend : redéfinir les caractéristiques des données (1 table par catégorie objet)
- Spécifier le processus d'installation :
 - Frontend (Android Studio)
 - Backend (Laravel, mysql, phpmyadmin)
- Fenêtre ou pop up pour afficher les caractéristiques
- Suite du projet :

- Boutons cliquables qui amènent vers une description détaillée d'un objet
- Barre de recherche qui fonctionne
- Navigation entre plusieurs pages et l'ajout de la page Historique
- Description de tout le backend

Gestion de projet :

- Lien de visioconférence : <https://webconf.imtbs-tsp.eu/frontend/dja-6nh-ahu>

Actions à suivre / Todo list

Description	Délai	Livrable
Pop-up caractéristiques d'un objet	1 Semaine	Code Android Studio
Barre de recherche fonctionnelle	1 Semaine	
Navigation entre plusieurs pages	1 Semaine	

6. Compte Rendu du 02/04/2020 :

Motif/type de réunion : Suivi	Lieu : Visioconférence
Présent(s) : CHAMROUK Mohamed, LABARRE Vincent, CINDRAK Dzenan	Date et durée : 02/04 à 18h, 40 min

Ordre du jour : Revue des livrables en cours et planification des prochains.

Informations échangées :

Remarques sur la gestion de projet :

- Adresse de la base de données : <http://stock.minet.net/>
- Documentation backend_serveur :
 - Nommer ce document "Mise en œuvre du serveur".
 - Détailler ce que sont les tables failed_jobs et migrations.
 - Le mettre dans le rapport.
- Mettre à jour les plannings : bien remettre les dates à jour et détailler l'activité réelle (ne pas dire simplement "programmation design" par exemple mais détailler ce qu'on a fait dans cette programmation design)).

Remarques sur le développement de l'application :

- Pourquoi plusieurs branches ? Pour pouvoir mettre en place les fragments (permet d'avoir une navigation plus fluide entre les différentes "fenêtres" de l'application et un code plus clair, simple et optimisé quant à la navigation).
 - Fragment : implémentable depuis la version Android 3.x .
- Dans le pop- up, faire des retours à la ligne (pour plus de clarté, en attendant peut- être de mettre les chaînes de caractère sous forme d'un tableau).
- "Historique depuis : " à la place de "Choix de la date" (pour avoir une utilisation plus intuitive du bouton) dans le bouton permettant de choisir cette date.
- Mettre une légende dans History pour expliquer le code couleur des boutons (ou autre chose pour avoir une utilisation intuitive).
- Rajouter les noms des contributeurs de chaque partie (classes, méthodes ...) du code.
- Ajouter des commentaires dans les codes !
 - Ajouter les descriptions des classes faites dans le rapport
 - Permet d'examiner la cohérence entre ce qu'on a écrit dans le rapport et ce qui a été codé.
 - Avoir un code "auto-suffisant" : pas besoin d'aller voir le rapport pour le comprendre !
- S'assurer qu'à aucun moment l'utilisateur rentre une date mais qu'il choisisse (en cliquant sur un bouton) une date.
- Dans une classe constante, définir des attributs constants (dans Android Studio, rentrer ces attributs dans le fichier xml "strings").
 - Pas besoin de retaper à chaque fois la chaîne de caractères.
 - Pas besoin de modifier à chaque fois la classe, seulement l'attribut dans la classe constante.

Actions à suivre / Todo list

Description	Délai	Livrable
Mise à jour de la documentation du serveur/	pour 6 avril	Rapport
Ajouter les commentaires dans le code, qui l'a rédigé, mettre tous les string dans le fichier xml strings	pour 6 avril	Code Android Studio.
Améliorer le pop-up (retour à la ligne ou si possible ajout d'un tableau).	pour 6 avril	
Terminer le développement de la barre de recherche	pour 6 avril	
Améliorer la partie web	pour 6 avril	Site Web
Mise à jour des plannings	pour 6 avril	Chaque planning à jour.
Si on a le temps, mettre en place le bouton ajouter ou supprimer des objets.	pour 6 avril	Code Android Studio.

7. Compte Rendu du 09/04/2020 :

Motif/type de réunion : Suivi	Lieu : Visioconférence
Présent(s) : CHAMROUK Mohamed, LABARRE Vincent, CINDRAK Dzenan, BAILLEUL Valentin	Date et durée : 09/04 à 18h, 40 min

Ordre du jour : Revue des livrables en cours et planification des prochains.

Informations échangées :

Remarques sur la gestion de projet :

- Importance du planning :
 - Bien le mettre à jour car il sert d'indicateurs (suivi des heures d'activité, il ne faut pas faire moins, mais non plus beaucoup plus que ce qui est attendu).
 - Problème avec OnlyOffice dans l'affichage des dates (faire les modifications nécessaires).
- Mettre le code en fichier zip sur le partage.

Remarques sur le développement de l'application :

- Importance de préciser auteur et contributeur.
 - Permet la différenciation entre propriété d'usage (tous les participants du projet) et propriété morale (créateur du code).
- "Historique depuis : " à la place de "Affichage entre" et supprimer la deuxième date.
- La recherche se fait par rapport au nom de l'objet pour l'instant, il faut rajouter la fonctionnalité pour choisir sur quelle caractéristique la recherche doit se faire.

Actions à suivre / Todo list

Description	Délai	Livrable
Correction des différents bugs	pour le 30 avril	Code Android Studio.
Mettre en place le bouton ajouter ou supprimer des objets.	pour le 30 avril	

Rajouter la possibilité de rechercher avec différents filtres	pour le 30 avril	
---	------------------	--

8. Compte Rendu du 06/05/2020 :

Motif/type de réunion : Suivi	Lieu : Visioconférence
Présent(s) : CHAMROUK Mohamed, LABARRE Vincent, CINDRAK Dzenan	Date et durée : 06/05 à 18h, 50 min

Ordre du jour : Revue des livrables en cours et planification des prochains.

Informations échangées :

Remarques sur la gestion de projet :

- Problème encore sur le planning.
 - Ceci occasionne une perte de points dans la gestion de projet (en entreprise, ce serait une perte d'argent).
 - Bien modifier les dates.
 - Détailler l'activité effective.

Remarques sur le développement de l'application :

- Spécifier auteurs et contributeurs.
- Barre de recherche sur toutes les pages de l'appli
- Cohérence entre le code et le rapport :
 - Changement dans la liste des objets à notifier dans le rapport.
- Remplacer "Retirer" par "Supprimer" (plus explicite) dans "Historique".
- Ajouter une vue détaillée des objets sur la page d'accueil
 - menu déroulant pour chaque objet
 - ou remplacer la vue actuelle avec le détail de tous les objets
- Avoir la même "philosophie" de fonctionnement entre l'affichage sur l'application et l'affichage sur le site stock.minet.net (optionnel)
- Pour une version future, on pourrait penser à mettre en place un système de droits entre 2 types d'utilisateur : utilisateur lambda et administrateur.
- Soutenance : le 3 juin à 14H00.

Actions à suivre / Todo list

Description	Délai	Livrable
Modification des plannings (bien mettre les dates dans le bon format + détailler la description de l'activité réelle).	pour le 14 mai	Plannings
Modification du rapport.	pour le 14 mai	Rapport

-Vue détaillée des objets sur la page d'accueil -Bouton de rechercher et d'ajout sur toutes les pages	pour le 14 mai	Code Android Studio.
--	----------------	----------------------

9. Compte Rendu du 14/05/2020 :

Motif/type de réunion : Suivi	Lieu : Visioconférence
Présent(s) : CHAMROUK Mohamed, LABARRE Vincent, CINDRAK Dzenan, BAILLEUL Valentin	Date et durée : 14/05 à 18h05, 30 min

Ordre du jour : Revue des livrables en cours et planification des prochains.

Informations échangées :

Remarques sur la gestion de projet :

- Dans le rapport, on rentre les méthodes fonctionnelles :
 - Spécifier leurs paramètres.
 - Documenter chaque paramètre.
 - Documenter chaque méthode.
- Améliorer la forme :
 - Tout écrire en même police d'écriture, de même taille.
 - Bien mettre tous les CR en annexe, la bonne pagination, le bon sommaire
- ...
- Documenter les classes.
- Merge toutes les branches en documentant le plus proprement possible.
- La description fonctionnelle doit être indépendante du langage/protocole utilisé.
- Description des fonctionnalités.
- Il faut décrire les mots-clés d'Android Studio.
 - Par exemple, Context correspond à quelle utilité fonctionnelle ?
- Avant la conclusion, il faut un paragraphe qui récapitule et analyse ce qui a été / n'a pas été fait.
- Un chapitre du rapport doit porter sur la partie serveur.
- La conclusion du rapport :
 - Aspect apprentissage technique.
 - Aspect travail en équipe.

Remarques sur le développement de l'application :

-

- Nom, prénom, email de la personne à qui on a prêté l'objet.
- Modifier le design du popup des items dans l'historique.
 - Affichage des boutons en bas du popup.
- Ajouter des catégories d'objet (en plus de câble et écran) qui seront utiles pour MINET.
- 26 mai 20h00 : remise des livrables.
- Pour la soutenance :
 - "Scénariser" la présentation :
 - Pour la démonstration :
 - Prévoir ce que l'on veut montrer pour le jour de la soutenance.

- En faire une version propre sous forme vidéo pour pallier un éventuel problème le jour j.
- Nombre de slides :
 - Slides utiles (ie exceptées celles obligatoires comme la première par exemples) :
 - Entre 2 et 3 minutes.

Actions à suivre / Todo list

Description	Délai	Livrable
Rapport à compléter (Description + maquettes).	20 mai	Rapport
Améliorer les parties design évoquées plus haut, finaliser la documentation et merge le code.	20 mai	Code Android Studio

10. Compte Rendu du 20/05/2020 :

Motif/type de réunion : Suivi	Lieu : Visioconférence
Présent(s) : CHAMROUK Mohamed,, CINDRAK Dzenan, BAILLEUL Valentin, LABARRE Vincent	Date et durée : 20/05 à 17h35, 30 min

Ordre du jour : Revue des livrables en cours et planification des prochains.

Informations échangées :

Remarques sur la gestion de projet :

- Il n'y a pas besoin de mettre le livrable sur la plateforme (voir le mail de Mr Ranc)
- Pour la soutenance, il faut convenir d'une date pour avoir 2 membres dans le jury

Remarques sur le développement de l'application :

- Tout le programme est sur une seule branche mais pas sur la branche master du fait de quelques problèmes.
- Il faut modifier le design des boutons pop-up pour une meilleur ergonomie.
- Mettre les nouvelles classes dans le git et dans le rapport.
- Dates disponibles pour la soutenance:
 - mercredi 3 14h
 - jeudi 4 après midi
 - vendredi 5 après midi
 - jeudi 11 après midi
 - vendredi 12
 - mardi 9 10h
 - Lundi 15 après midi

mercredi matin 10h

Actions à suivre / Todo list

Description	Délai	Livrable
Finaliser le programme Android	5 juin	Code android
Finaliser le rapport	5 juin	Rapport

11. Compte Rendu du 05/06/2020 :

Motif/type de réunion : Suivi	Lieu : Visioconférence
Présent(s) : CHAMROUK Mohamed, LABARRE Vincent, CINDRAK Dzenan, BAILLEUL Valentin	Date et durée : 05/06 à 18h00, 45 min

Ordre du jour : Revue des livrables pour une dernière mise à jour.

Informations échangées :

Remarques sur la gestion de projet :

- Soutenance lundi 15 juin à 14h00.
 - Démonstration de notre réalisation :
 - Copie d'écran des maquettes.
 - Slides backup (en fin de présentation, à montrer si jamais problème ou question).
 - Avoir un film montrant l'application en train de fonctionner (servant de back up, de 1 à 3 minutes).
 - Peut servir dans un CV d'ailleurs (pour présenter ce qu'on a fait visuellement plutôt que simplement avec deux lignes de texte dans le CV).
 - Avant jeudi soir (11/06), si on met les slides sur Partage, on peut avoir un retour dessus le vendredi soir.
- Mettre le rapport en police 11.
- Réduire les marges en haut, en bas et à droite (laisser à gauche) et les espaces entre les parties.
- Fonction de la section 2 sont majoritairement dans la classe Utils :
 - Elles rentrent dans la description de la classe Utils.

- Juste après la classe Utils, mettre la description de ses différentes méthodes !
- On commence par les fonctionnalités (page 4, correspondant à une analyse fonctionnelle), non par les fonctions ni les classes !
- Dans la description fonctionnelle des besoins, on ne doit pas rentrer dans les détails :
 - Les détails sont dans l'analyse fonctionnelle (chapitre 3).
- Il manque les parties perspective et conclusion.
- En annexe :
 - Annexe 1 : Compte-rendus de réunion.
 - Contenant les sous sections décrivant chaque réunion
 - Annexe 2 : Compte-rendu d'auto-formation.
 - Annexe 3 : Plannings de gestion de projet.

Remarques sur le code :

- Manque le code de la partie web.
 - Remarque sur les commentaires (du code de la partie web) :
 - Javadoc : génère automatiquement la documentation du code html.
 - Vérifier que javadoc est pris en charge par Android studio :
 - Si oui, il faut reformater toutes les classes avec les commentaires javadoc.
- Supprimer un des stocks (dans le nommage).
- Convention de formatage :
 - Nom de domaine inversé (exemple : eu.tsp.stocks.ui.notifications).
 - Bien utiliser les conventions de nommage pour les fichiers/packages et les commentaires.

Actions à suivre / Todo list

Description	Délai	Livrable
Compléter le rapport avec les remarques ci-dessus.	8 juin	Rapport
Formatage commentaire javadoc (si possible sur Android studio) + nom fichier.	8 juin	Code Android Studio

Annexe 2 : Compte-rendus de l'autoformation :

1. Vincent LABARRE :

Au cours de la semaine du 17 février, j'ai commencé à suivre le MOOC « Créez des applications pour Android ». Après avoir découvert les bases d'Android Studio avec (Création d'une activité, découverte des layouts ..), j'ai pris conscience de la nécessité d'étudier le XML (étant donné qu'il est très souvent évoqué dans ce cours). Par conséquent, j'ai suivi en parallèle la première partie du MOOC (Structurez vos données avec XML) de manière à en maîtriser les bases (et d'ainsi pouvoir comprendre son utilisation dans Android Studio).

Par la suite, trouvant que le premier MOOC sur Android Studio passait un peu trop vite sur les notions de base et qu'il n'exploitait quasiment pas l'interface graphique d'Android Studio, j'ai décidé de suivre un autre MOOC intitulé « Developing Android Apps with Kotlin » qui était plus pédagogique. En même temps, j'ai effectué les différents tutoriels mis à disposition directement sur le site developer.android.com.

En outre, j'ai pris le temps de suivre un autre MOOC sur Kotlin se nommant : « Initiez-vous à Kotlin » permettant de saisir les caractéristiques de ce langage (qui se veut être un langage « simplifié » de Java, donc ils ont beaucoup de points en commun mais présentent aussi certaines grosses différences, comme par exemple l'inférence de type, la déclaration d'une classe ou encore le fait que les conditions sont des expressions en Kotlin alors qu'en Java ce sont des instructions). Ainsi, j'ai pu réviser en même temps certains principes de Java (comme la déclaration d'erreur) et en découvrir de nouveau (le mot-clé final, l'énumération ...).

Finalement, j'ai étudié beaucoup de choses cette semaine, ce qui m'a permis de découvrir de nombreuses nouvelles notions qui seront utiles pour ce projet de Développement Informatique. Néanmoins, n'ayant pas eu le temps de tout assimiler en profondeur, je vais devoir fournir un travail conséquent durant les prochaines semaines en vue de consolider et d'approfondir ces nouvelles connaissances.

2. Valentin BAILLEUL :

J'ai commencé le Mooc « Créez des applications pour Android » lors de la semaine du 24 février. Je peux, grâce à ce Mooc apprendre le fonctionnement d'Android studio. La programmation d'une application m'était inconnue auparavant et je commence de plus en plus à apprécier et à voir les possibilités que m'offre le fait de suivre cette formation pour mes projets futurs.

J'ai continué à me former pendant la programmation de l'application. J'ai accentué ma formation sur le design. J'ai appris à créer des boutons "design" car l'on pourrait croire que le design est un peu superflu mais il est très important car il donne envie à l'utilisateur d'utiliser notre application et pas une application concurrente. De plus le design c'est le premier avis que ce fait l'utilisateur sur l'application car lorsque l'on télécharge une application sur google play par exemple, on ne peut que voir des photos et l'on fait son choix en partie avec les photos. On peut alors se projeter dans l'application.

Je me suis ensuite intéressé à l'ergonomie d'une application. (Autrement dit "comment améliorer l'UX d'une application"). J'ai pu y apprendre plusieurs points intéressants que j'ignorais comme par exemple la nécessité de pouvoir naviguer dans l'application à l'aide d'une seule main.

3. Mohamed CHAMROUK :

J'ai débuté la formation pour Android et le langage Kotlin il y a 2 mois et je continue à la suivre. La formation est proposée par des employés de Google et orienté pratique plutôt que théorie. Les différentes étapes de la formation se font simplement et sont expliquées en détails. Pour ce qui est de l'architecture, elle est expliquée hiérarchiquement de manière à traiter en premier les éléments de surface (Image, textes, formes, boutons...) puis les éléments de bas niveau tel que la fragmentation des pages, les liens entre les différentes pages etc...

Je n'ai, pour l'instant, pas complété entièrement la formation mais pour la programmation des premiers de l'application il ne m'est pas nécessaire de poursuivre entièrement la formation étant donné que cette dernière traite tous les détails de la programmation sur Android.

4. Dzenan CINDRAK :

J'ai commencé ma formation au cours de la semaine de 24 février. Premièrement, j'ai suivi le MOOC « Créez des applications pour Android » qui m'a permis d'approfondir les bases d'Android Studio. D'autre part, j'ai regardé de nombreuses vidéos et lu la documentation Android Studio sur le site developer.android.com. Je pense être capable de commencer à coder l'application avec les compétences que j'ai acquises. Le développement de l'application va me permettre de consolider ces connaissances.

Par la suite, je me suis renseigné sur la façon dont communique une application Android et une base de données. Il faut passer par un serveur web pour envoyer et recevoir les données. Il existe de nombreux exemples sur internet et je pense que cela ne sera pas difficile à concevoir. D'autre part, il existe aussi des librairies permettant d'envoyer des requêtes à des serveurs web depuis un téléphone Android, il faut donc en choisir une et la maîtriser.

Annexe 3 : Plannings :

1. Vincent LABARRE :

Date de l'activité	Description de l'activité	LABARRE Vincent			Activité réelle
		Heure début	Heure fin	Durée	
1/28/2020	Réunion de lancement	17:00	17:40	0:40	Réunion de lancement
1/30/2020	Etablissement des plannings	16:00	17:15	1:15	Rédaction du CdC
2/4/2020	Réunion de suivi	18:00	18:40	0:40	Réunion de suivi
2/7/2020	Etat de l'art	16:00	17:30	1:30	Etat de l'art (Découverte d'application de stockage) + mise en place des plannings
2/8/2020	Etat de l'art	10:00	11:30	1:30	Etat de l'art (Rédaction)
2/13/2020	Formation	15:45	17:15	1:30	Installation + prise en main d'Android Studio
2/16/2020	Formation	14:45	16:30	1:45	Exploration des layouts (MOOC)
2/17/2020	Formation	13:00	14:30	1:30	Découverte du XML (MOOC)
2/18/2020	Formation	10:00	12:00	2:00	Découverte des écrans, intent sur Android Studio (MOOC)
2/19/2020	Formation	16:30	18:00	1:30	Exploration du site developer.android.com et accomplissement de certains tuto de base
2/20/2020	Formation	16:00	18:00	2:00	Découverte de Kotlin (MOOC)
2/21/2020	Formation	14:00	16:00	2:00	Création guidée d'une application : Dice Roller (MOOC)
2/22/2020	Formation	16:00	18:00	2:00	Découverte des styles, contraintes pour les layouts (MOOC)
2/23/2020	Formation	14:00	16:00	2:00	Approfondissement de Kotlin (MOOC + recherches internet)
2/25/2020	Réunion de suivi	18:05	19:15	1:10	Réunion de suivi
2/27/2020	Gestion de Projet (GdP)	22:00	23:15	1:15	Modification rapport (insertion sommaire automatique + CR en Annexe + réorganisation des sections)
3/2/2020	Réunion de suivi	18:00	18:30	0:30	Réunion de suivi
3/8/2020	Formation	10:00	12:00	2:00	Cycle de vie d'une activité + fragments (MOOC)
3/10/2020	GdP + programmation	13:30	15:30	2:00	Ajout bibliographie et CR + programmation de la maquette "historique" : ajout d'une Toolbar
3/14/2020	Programmation	15:00	17:00	2:00	Ajout ScrollView + boutons éléments à l'accueil
3/18/2020	Programmation	15:00	17:00	2:00	Ajout d'éléments stylistiques pour les boutons de l'accueil
3/23/2020	Réunion de suivi	18:00	18:40	0:40	Réunion de suivi
3/24/2020	Programmation	8:00	10:00	2:00	Création template pop item (Layout + Classe)
3/27/2020	Formation	20:00	21:30	1:30	Comment créer un popup ? un fragment ? (recherches internet)
3/28/2020	Programmation	10:00	12:00	2:00	Création pop up pour chaque bouton de la page d'accueil
3/30/2020	Programmation	22:30	23:30	1:00	Modifications de forme sur la page History
4/1/2020	GdP	21:30	22:00	0:30	Rédaction de la description de la base de données
4/2/2020	Réunion de suivi	18:00	18:40	0:40	Réunion de suivi
4/3/2020	Programmation	18:00	19:00	1:00	Correction des erreurs, réarrangement du code (dans tout le code kotlin), ajout des strings dans le fichier xml
4/4/2020	Programmation	15:00	17:00	2:00	Ajout des commentaires (dans tout le code Kotlin)
4/9/2020	Réunion de suivi	18:00	18:40	0:40	Réunion de suivi
4/23/2020	Programmation	19:00	20:00	1:00	Recherche du problème d'affichage de la page d'accueil sur certains téléphones
4/25/2020	Programmation	14:00	16:00	2:00	Création du template pour le fragment_add_item (ie amélioration de l'ancien + ajout de deux boutons)
4/27/2020	Programmation	15:30	16:45	1:15	Création branche fragmentLabarre pour poursuivre l'implémentation du template fragment_add_item (pb pour le faire car pas adapté au début à la liste d'objets mise en place)
4/29/2020	Programmation	11:00	11:30	0:30	Correction des bugs d'affichage du pop up permettant l'ajout d'un objet dans la page d'accueil (Stocks).
5/6/2020	Réunion de suivi	18:00	18:50	0:50	Réunion de suivi
5/7/2020	GdP	17:30	18:30	1:00	Modification des maquettes dans le rapport : ajout de maquettes manquantes +
5/8/2020	Gdp	16:15	17:00	0:45	Mise en cohérence entre la description des classes dans le rapport et les classes codées dans Android Studio
5/11/2020	Réunion de suivi	18:05	18:35	0:30	Réunion de suivi
5/17/2020	Programmation	10:00	11:15	1:15	Conception des classes adaptateur et projecteur pour enrichir le nombre de catégorie d'objets, il reste à l'implémenter sur Android Studio
5/23/2020	Réunion de suivi	17:35	18:05	0:30	Réunion de suivi
5/24/2020	Programmation	16:45	17:45	1:00	Implémentation de Adaptateur et Projecteur dans les codes XML
6/2/2020	Gdp	9:30	10:30	1:00	Uniformisation du type et de la taille de la police d'écriture dans le rapport, ajout de la description des classes Adaptateur et Projecteur, ajout de la description des tables adaptateurs et projecteurs
6/5/2020	Réunion de suivi	18:00	18:45	0:45	Réunion de suivi
6/6/2020	GdP	10:00	11:00	1:00	Modification du rapport : Ajout des annexes et déplacement des sections
8/6/2020	GdP	14:00	15:30	1:30	Modification du rapport : paufinement (taille de la police, marges) et ajouts des perspectives et de la conclusion
Total :		59:35:00			

2. Valentin BAILLEUL :

AN DE CHARGES PRÉVISIONNELLES					
Date de l'activité	Description de l'activité	BAILLEUL Valentin			Activité réelle
		Heure de début	Heure fin	Durée	
1/28/2020	Réunion de lancement	17:00	17:40	0:40	Réunion de lancement
1/31/2020	Rédaction du CdC	18:00	19:00	1:00	Rédaction du CdC
2/4/2020	Réunion de suivi	18:00	18:40	0:40	Réunion de suivi
2/8/2020	Formation	18:00	19:00	1:00	Maquettes page historique
2/17/2020	Formation et maquettes	19:00	21:00	2:00	Maquettes uniquement
2/19/2020	Formation	19:00	21:00	2:00	Maquettes design global (couleur et forme)
2/22/2020	Formation	19:00	21:00	2:00	Maquettes + formation vidéo sur l'utilisation de android studio
25/02/2020	Réunion de suivi	18:00	18:40	0:40	Réunion de suivi
26/02/2020	Début codage	20:00	22:00	2:00	Formation vidéo
3/2/2020	Réunion de suivi	18:00	18:40	0:40	Programmation design page "stocks"
3/7/2020	Programmation	20:00	22:00	2:00	Programmation design page "stocks" Formation sur les text_views + différents essais
3/8/2020	Programmation	10:00	12:00	2:00	Programmation design page "stocks" Codage des text_views dans stocks_items.xml
3/12/2020	Programmation	20:00	22:00	2:00	Programmation design page "stocks" Codage sur stocks_item.xml deux boutons après une formation
3/23/2020	Programmation	18:00	18:40	0:40	Programmation barre de recherche. Formation sur la programmation et design d'une barre de
3/28/2020	Programmation	20:00	21:00	1:00	Programmation barre de recherche. Amélioration de l'ergonomie et du design de la barre de
3/30/2020	Réunion de suivi	18:00	18:40	0:40	réunion reporté au jeudi
4/4/2020	Programmation	20:00	22:00	2:00	Programmation barre de recherche (formation sur la barre de recherche)
4/8/2020	Programmation barre de recherche			0:30	Amélioration de la barre de recherche (design)
4/9/2020	Réunion de suivi	18:00	18:40	0:40	Réunion de suivie
4/14/2020	Programmation	0:00	0:00	0:40	Formation sur la programmation spécifique aux boutons et fenêtres (niveau design et ergonomie)
4/15/2020	Programmation	12:00	14:00	2:00	Programmation dialog fragment_manage_item_dialog.xml. Création des
4/18/2020	Programmation	20:00	22:00	2:00	Programmation dialog fragment_manage_item_dialog.xml. Amélioration
4/20/2020	Réunion de suivi	18:00	18:40	0:40	réunion reporté au jeudi
4/23/2020	Programmation	0:00	0:00:00	2:00	Programmation dialog fragment alert
4/25/2020	Programmation	20:00	22:00	2:00	Programmation dialog fragment_manage_item_dialog.xml. Changement
4/27/2020	Programmation	20:00:00	23:00	3:00	Création de nouveaux code en xml pour "centraliser" les designs. (travail sur color.xml styles.xml)
4/30/2020	Réunion de suivi	17:00	17:40	0:40	
5/4/2020	Réunion de suivi	18:00	18:40	0:40	Réunion reporté au 6 mai
6/4/2020	Réunion de suivie	18:00	18:40	0:40	Réunion de suivie sur les différents livrables
5/9/2020	Programmation	20:00	22:00	3:00	Formation sur l'amélioration de L'UX d'une application. Comment la rendre agréable et facile d'utilisation.
5/10/2020	Programmation	11:00	12:00	1:00	Modification Sur la partie personnel du rapport
5/11/2020	Réunion de suivi	18:00	18:40	0:40	Réunion reporté au 13 mai
5/13/2020	Réunion de suivi	18:00	18:40	0:40	
5/16/2020	Programmation	20:00	22:00	2:00	Création des nouvelles classes "disques dur" "switch"
5/17/2020	Programmation	10:00	12:00	2:00	Résolution du bug du spinner dans la page d'accueil
6/4/2020	Programmation	18:00	20:00	2:00	Créations des slides pour la soutenance
6/5/2020	Programmation	20:00	22:00	2:00	Création des slides pour la soutenance
6/5/2020	Réunion de suivi	18:00	18:40	0:40	
Total :				52:30:00	

3. Mohamed CHAMROUK :

Date de l'activité	Description de l'activité	CHAMROUK Mohamed			Activité réelle
		Heure début	Heure fin	Durée	
28-Jan-20	Réunion de lancement	17:00	17:40	0:40	Réunion de lancement
30-Jan-20	Etablissement des plannings	13:00	13:40	0:40	Etablissement des plannings
4-Feb-20	Réunion de suivi	18:00	18:40	0:40	Réunion de suivi
8-Feb-20	Formation	12:00	14:00	2:00	Formation
9-Feb-20	Formation et CdC	19:00	21:00	2:00	Formation (Kotlin) et Cahier des Charges (définition des classes et fonction nécessaires à l'application)
15-Feb-20	Formation	17:00	20:00	3:00	Formation (Kotlin)
17-Feb-20	Formation et maquettes	16:00	18:00	2:00	Maquettes
25-Feb-20	Réunion suivi	18:00	18:40	0:40	Réunion suivi
26-Feb-20	Début codage	20:00	22:00	2:00	Formation Android
1-Mar-20	Programmation	12:00	14:00	2:00	Correction des maquettes et réalisation des nouvelles
2-Mar-20	Réunion de suivi	18:00	18:40	0:40	Revue des modifications faites dans le rapport et planification des prochains livrables.
7-Mar-20	Programmation Formation	10:00	12:00	2:00	Formation sur les serveurs web et la communication avec les applications Android
12-Mar-20	Programmation	16:00	19:00	3:00	Essai de construction d'une application avec une navigation en bas.
22-Mar-20		14:00	17:00		
23-Mar-20	Réunion de suivi	18:00	18:40	0:40	Revue des modifications faites dans le rapport et planification des prochains livrables.
28-Mar-20	Programmation	12:00	15:00	3:00	Elaboration de l'architecture en fragments et adaptation du code déjà existant (classes DataAdapter et HistoryAdapter)
30-Mar-20	Réunion de suivi	18:00	18:40		
30-Mar-20	Maquettes	17:00	18:00	1:00	Redesign de l'application avec une barre de navigation en bas
1-Apr-20		20:00	22:00	2:00	Mise en place de la fenêtre pop-up au sein des fragments et re-design de la page historique
2-Apr-20	Réunion de suivi	18:00	18:40	0:40	Revue des modifications faites dans le rapport et planification des prochains livrables.
4-Apr-20	Formation	10:00	12:00	2:00	Recherche sur le fonctionnement de la barre de recherche
6-Apr-20	Réunion de suivi	18:00	18:40		
8-Apr-20	Programmation Formation	16:00	21:00	5:00	Implémentation de la barre de recherche (modifications apportées à layout/menu et au fragmentHome)
9-Apr-20	Réunion de suivi	18:00	18:40	0:40	Revue des livrables en cours et planification des prochains.
12-Apr-20		10:00	14:00		
16-Apr-20		16:00	19:00		
19-Apr-20		10:00	14:00		
20-Apr-20	Réunion de suivi	18:00	18:40		
25-Apr-20		14:00	17:00		
27-Apr-20	Réunion de suivi	18:00	18:40		
30-Apr-20	Réunion de suivi	17:00			
3-May-20		10:00	13:00		
6-May-20	Réunion de suivi	18:00	18:50	0:50	Revue des livrables en cours et planification des prochains.
11-May-20	Programmation	20:00	0:00	4:00	Modification de la fenêtre pop-up au moment de la sélection d'un objet sur l'écran d'accueil (classe HomeAdapter)
13-May-20	Programmation	20:00	0:00	4:00	Rajout d'une deuxième fenêtre pop-up à la suite de la sélection d'un objet précis sur la fenêtre précédente. Rajout des boutons afin de prêter et retirer (classe HomeAdapter)
14-May-20	Réunion de suivi	18:05	18:35	0:30	Revue des livrables en cours et planification des prochains.
19-May-20	Réunion de suivi	20:00	0:00	4:00	Mise au propre du rapport et redéfinition des descriptions fonctionnelles (description des classes et des fonctions, fidélité des maquettes)
20-May-20	Réunion de suivi	18:00	18:40	0:40	Revue des livrables en cours et planification des prochains.
3-Jun-20	Programmation	17:00	20:00	3:00	Résolution des problèmes d'affichage pour le pop-up d'ajout de la page d'accueil et ajout du bouton d'information dans la page historique (classe HomeFragment).
4-Jun-20	Programmation	21:00	23:00	2:00	Ajustement du positionnement des éléments dans la page d'accueil et dans la page historique et unification de la mise en page des différents éléments graphique.
5-Jun-20	Réunion de suivi	18:00	18:45	0:45	Revue des derniers livrables pour vérification.
7-Jun-20	Programmation	21:00	0:00	3:00	Ajout de la confirmation de l'emprunteur dans la page d'accueil (classe HomeAdapter), finalisation de la documentation javadoc.
Total :				59:05:00	

4. Dzenan CINDRAK :

Date de l'activité	Description de l'activité	CINDRAK Dzenan			Activité réelle
		Heure déb	Heure fin	Durée	
1/28/2020	Réunion de lancement	17:00	17:40	0:40	Réunion de lancement
1/31/2020	Rédaction du CdC	18:00	19:30	1:30	Rédaction du CdC
2/4/2020	Réunion de suivi	18:00	18:40	0:40	Réunion de suivi
2/08/2020	Formation	12:00	14:00	2:00	Formation MOOC
2/17/2020	Formation	20:00	22:00		Malade (reporté)
2/20/2020	Formation	19:00	22:00		Malade (reporté)
2/25/2020	Réunion de suivi	18:00	18:40	0:40	Réunion de suivi
2/26/2020	Formation(report)	20:00	22:30	2:30	Report du 17/2/2020: Formation MOOC
2/28/2020	Formation(report)	20:00	22:00	2:00	Report du 18/2/2020: Recherche d'un moyen pour faire
3/1/2020	Modification de l'analyse fonctionnelle	15:00	17:00	2:00	Modification de l'analyse fonctionnelle
3/2/2020	Réunion de suivi	18:00	18:40	0:40	Réunion de suivi
3/8/2020	Programmation	22:30	1:30	3:00	Programmation: Mise en place de la base de donnée et du site web + commit initial (création des classes MainActivity, Item, Cable, Ecran et du fichier Utils.kt qui contient les fonctions utiles à l'application)
3/15/2020	Programmation	20:00	23:00	3:00	Programmation: Modification de la gestion des informations de la BDD par l'application (les informations de la BDD sont maintenant envoyées en format JSON)
3/16/2020	Programmation	23:00	1:00	2:00	Changement de la méthode d'affichage: On passe d'une scrollView à une recyclerView (création de la classe HomeAdapter)
3/17/2020	Programmation	21:45	23:45	2:00	Programmation: Ajout des commentaires pour rendre le code plus lisible + push backend final pour le premier rendu
3/20/2020	Programmation	19:00	19:40	0:40	Programmation: Merge sur la branch master + gestion des problèmes de compatibilité
3/23/2020	Réunion de suivi	18:00	18:40		Reportée
3/28/2020	Réunion de suivi (report)	18:00	18:40	0:40	Report du 23/3/2020: Réunion de suivi
3/30/2020	Réunion de suivi	19:00	23:45	4:45	Réunion de suivi reportée + Programmation: Mise en place de la page historique (création de 2 classes HistoryAdapter et HistoryFragment) et mise en place du filtre en fonction de la date
3/31/2020	Programmation	19:00	19:20	0:20	Programmation: Approfondissement de la page historique (les objets sont maintenant triés par date décroissante)
4/1/2020	Programmation	20:00	0:00	4:00	Programmation: Ajout de la prise en compte du fait qu'un objet est prêt, enlevé ou en stock et création des contours rouges, bleues ou jaunes pour savoir s'ils sont supprimés, présents ou prêts.
4/2/2020	Réunion de suivi (report)	18:00	18:40	0:40	Report du 30/3/2020: Réunion de suivi
4/6/2020	Réunion de suivi	18:00	18:50		Reportée
4/7/2020	Programmation	17:00	19:30	2:30	Programmation: Côté site web, création d'un script JS pour ajouter, prêter ou retirer des objets et modifications d'ordre stylistique
4/8/2020	Programmation	23:00	23:30	0:30	Programmation: Correction de quelques bugs graphiques
4/9/2020	Réunion de suivi (report)	18:00	18:40	0:40	Report du 6/04/2020: Réunion de suivi
4/10/2020	Programmation	18:00	22:00	4:00	Programmation: Modification du pop up pour afficher l'état d'un objet (création de la classe UpdateDialogFragment) et ajout de la fonctionnalité pour prêter, retirer ou retourner un objet sur la page historique
4/13/2020	Programmation	18:00	22:30	4:30	Programmation: Ajout d'un pop up pour ajouter un objet sur la page Stock (création de AddDialogFragment) et création de la fonctionnalité pour ajouter un objet dans la BDD
4/20/2020	Réunion de suivi	18:00	18:40		Reportée
4/27/2020	Réunion de suivi	18:00	18:40		Reportée
4/23/2020	Programmation	15:00	18:00	3:00	Programmation: Réctification d'un bug sur la page Stock qui empêchait l'affichage correct du nombre d'objets dans la BDD, ajout du rafraichissement 2 fois par seconde sur le pop up d'état d'un objet de la page historique
5/6/2020	Réunion de suivi (report)	18:00	18:50	0:50	Report du 20/4/2020: Réunion de suivi
5/11/2020	Réunion de suivi	18:00	18:40		Reportée
5/12/2020	Programmation	19:00	22:00	3:00	Gestion de projet: modification de la partie 3 du rapport (modification de la partie fonctionnelle et création d'une sous partie nommée principe de fonctionnement)
5/14/2020	Réunion de suivi (report)	18:00	18:40	0:40	Report du 11/05/2020: Réunion de suivi
5/16/2020	Programmation	18:00	20:10	2:10	Programmation: Ajout des tables Adaptateur et Projecteur dans la base de donnée côté serveur, ajout de la gestion de la requête "prêter à" et finalisation de la fonction "Prêter" dans l'application
5/18/2020	Réunion de suivi	18:00	18:40		Reportée
5/20/2020	Réunion de suivi (report)	18:00	18:40	0:40	Report du 18/05/2020: Réunion de suivi
6/1/2020	Programmation	18:00	19:30	1:30	Programmation: Ajout des classes et fonctionnalités liée aux classes Adaptateur et Projecteur
6/5/2020	Réunion de suivi	18:00	18:45	0:45	Réunion de suivi
6/7/2020	GdP	22:00	23:20	1:20	GdP: Ajout des commentaires suivant la norme KDoc
Total :		59:50:00			

VIII - Bibliographie :

- MOOC « Créez des applications pour Android » :
<https://openclassrooms.com/fr/courses/2023346/next-page-to-do>
- MOOC « Structurez vos données avec XML » :
<https://openclassrooms.com/fr/courses/1766341/next-page-to-do>
- MOOC « Developing Android Apps with Kotlin »
<https://classroom.udacity.com/courses/ud9012>
- MOOC « Initiez-vous à Kotlin » :
<https://openclassrooms.com/fr/courses/5353106/next-page-to-do>
- MOOC « Développez une application pour Android » :
<https://openclassrooms.com/fr/courses/3499366/next-page-to-do>
- Site de documentation Android:
<https://developer.android.com/>
- Site de la documentaton Kotlin:
<https://kotlinlang.org/>