

PROJET TUTEURE S2 : PRODUCTIV'HEAD



Daniel Blasko, Laurine Boulio, Lina Zouiri, Olivier Broutier.

IUT LYON 1 Département informatique

Table des matières

I.	Le projet <i>Productiv'head</i> : Une application Android pour optimiser votre productivité	2
1.	Présentation du projet	2
2.	Motivations & objectifs	2
3.	Demande, public cible et contexte	3
II.	L'application Android : ses fonctionnalités, son aspect et son implémentation	4
1.	Fonctionnalités de l'application	4
2.	Charte graphique	6
3.	Implémentation : architecture, diagrammes et maquette	8
a)	Architecture	8
b)	Diagrammes	8
c)	Maquette	15
III.	Le développement de l'application	20
1.	Organisation et planning	20
2.	Choix des outils utilisés	22
a)	Les outils de collaboration	22
b)	Les outils de modélisation	22
c)	Les outils de développement	23
3.	Les contraintes rencontrées et à venir	24

I. Le projet *Productiv'head* : Une application Android pour optimiser votre productivité

1. Présentation du projet

Productiv'head est un projet d'application native Android servant à rendre l'utilisateur plus productif, qu'il soit étudiant, professionnel ou ait tout simplement des tâches à accomplir et à organiser. *Productiv'head* n'est pas un assistant qui fait les choses à la place de l'utilisateur : il s'agit d'un outil qui permet à l'utilisateur de concentrer son temps et ses ressources sur le fait d'accomplir des tâches et d'avancer au lieu de perdre son temps à les organiser et à les mémoriser.

Pour ce-faire, l'application propose un ensemble de fonctionnalités servant à optimiser son temps, se rendre plus efficace, ne pas oublier des choses ou encore à instaurer de nouvelles habitudes en un mois.

Le choix du format d'application mobile n'est pas fait par hasard : *Productiv'head* accompagne l'utilisateur tout au long de la journée, où qu'il aille, en se trouvant en permanence à la portée de sa main, prêt à le notifier d'un rappel important ou à le laisser s'alléger d'une information à retenir.

2. Motivations & objectifs

L'idée de l'application est avant tout née d'un besoin ressenti par l'équipe. En effet, nous sommes quatre étudiants. Pour la plupart, nous devons gérer en plus des études divers événements, s'occuper de notre logement, voire travailler. La productivité est donc un élément clé de nos vies plus que jamais auparavant et nous cherchons tous à l'optimiser autant que possible. Et pour répondre à ce besoin, nous avons besoin d'un outil flexible, portable, que nous aurons toujours à disposition et qui saura s'adapter à nos besoins tout en ne consommant pas trop de notre temps.

C'est donc cette problématique qui a motivée le projet *Productiv'head*. Nous souhaitons créer un outil qui nous correspond, et qui correspond généralement à toute personne souhaitant accomplir des choses et avancer au quotidien. L'objectif est surtout de centraliser toutes sortes d'outils que nous utilisons déjà pour la plupart – tout en les améliorant et en en ajoutant des nouveaux – dans une unique application toujours disponible sur notre smartphone. En effet, nous utilisons déjà des timers de travail, des todo-lists et autres, mais nous voulons créer un espace de travail unique : un outil que l'utilisateur lance quand il veut travailler, et qui répond à tous ses besoins pour qu'il n'ait pas à le quitter pour ne pas perdre son temps. Un vrai *couteau suisse de productivité*, en somme.

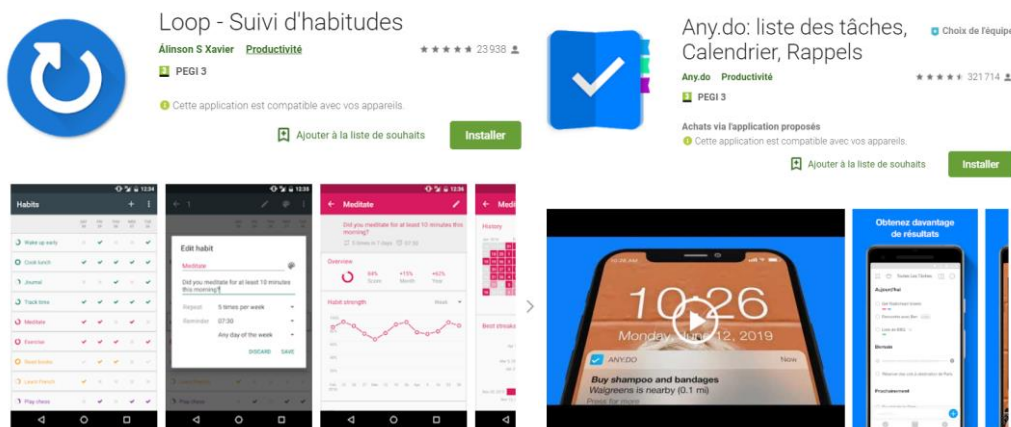
3. Demande, public cible et contexte

Notre application vise un public plutôt large : accessible à n'importe qui ayant un smartphone sous Android, elle pourra potentiellement être obtenue par beaucoup de personnes. Mais elle vise avant tout un public d'étudiants souhaitant s'organiser et travailler plus efficacement. Elle concerne aussi des personnes actives professionnellement ou même personnellement, en bref, souhaitant devenir plus productives et exploiter au maximum leur temps.

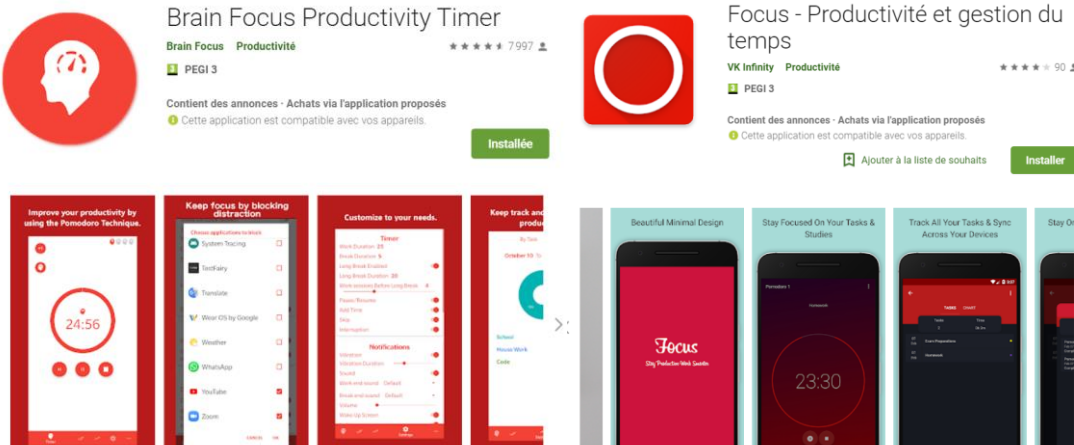
La demande est donc importante, ne serait-ce que du côté des étudiants. Ce genre de services sont très demandés, et surtout sous forme d'application mobile. Toutefois, cette importante demande est aussi synonyme d'une agressive concurrence : sur le Play Store, de nombreuses applications de productivité, aux fonctionnalités diverses, existent d'ores et déjà.

Toutefois, notre application se différencie par sa variété de fonctionnalités centralisées (tous les concurrents se focalisent sur une fonctionnalité en particulier) et surtout par un outil forgé en réponse aux manques ressentis et aux besoins non remplis par les applications déjà existantes. En effet, si nous avons décidé de créer cette application, c'est surtout car les solutions existantes ne nous satisfaisaient pas !

Voici quelques exemples d'applications concurrentes :



<https://play.google.com/store/apps/details?id=org.isoron.uhabits> <https://play.google.com/store/apps/details?id=com.anydo>



<https://play.google.com/store/apps/details?id=com.AT.PomodoroTimer>
<https://play.google.com/store/apps/details?id=infinity.vk.com.focus.your.mind>

II. L'application Android : ses fonctionnalités, son aspect et son implémentation

1. Fonctionnalités de l'application

Menu principal

Le menu principal sera l'écran d'accueil au lancement de l'application, et accessible à tout moment grâce à un bouton de la toolbar. Depuis ce menu, on pourra, via une liste d'aperçus, accéder à n'importe quelle fonctionnalité de l'application. On aura aussi accès aux fonctionnalités de la toolbar, évoquées ci-dessous.

Toolbar

La toolbar, propre à toute application Android native, est présente dans chaque activité (fenêtre) de l'application. Depuis, on pourra accéder aux paramètres, activer ou éteindre le mode travail (expliqué ci-dessous), revenir au menu principal et enfin accéder à n'importe quelle autre fonctionnalité de l'application via un drawer (tiroir) partant de la gauche.

Menu de paramètres

Le menu de paramètres sera une activité à part entière accessible à tout moment depuis la toolbar via une icône. Divisé en plusieurs sections, on pourra y paramétrer les notifications (horaires, fréquence...), le thème de l'application, observer ses statistiques ou encore régler le mode travail en détail. En fonction des retours d'utilisation sur l'application, le contenu des paramètres s'agrandira pour offrir un outil plus flexible et polyvalent.

Todo-list

La todo-list permettra d'organiser ses idées, ses journées... Il s'agit d'un outil indispensable à la productivité, utilisé en permanence : c'est pourquoi notre todo-list doit être réalisée au mieux, car il s'agira d'une des fonctionnalités clés de l'application. On pourra y créer des listes qui auront un titre, consulter des listes, et ajouter ou supprimer des éléments à chaque liste. Les éléments peuvent être marqués comme faits ou urgents, ou encore être modifiés. Un élément urgent pourra envoyer une notification de rappel. La persistance des données se fera au travers de la base de données implémentée par l'application.

Système de notifications

Le système de notifications sera un élément clé de l'application, car il sera au cœur des différentes fonctionnalités. En effet, une todo-list pourra être marquée pour renvoyer un rappel, et la notification devra mener à cet élément de la todo-list. L'utilisateur pourra aussi régler des notifications personnalisées, tels que des rappels. De plus, dans les paramètres, on pourra activer une option qui permet d'envoyer une notification tous les jours à une heure donnée pour penser à venir travailler. Enfin, le suivi des habitudes enverra aussi une notification quotidienne (qui peut être désactivée) pour rappeler d'entrer son avancement du jour sur les habitudes. Toutes ces notifications doivent être paramétrables afin de créer une expérience qui ne dérange pas l'utilisateur mais qui lui permet aussi de ne pas se soucier de trop de choses en ayant des rappels réguliers.

Suivi des habitudes

La fonctionnalité de suivi des habitudes a pour objectif d'aider l'utilisateur à instaurer de nouvelles habitudes dans sa vie mensuellement. En effet, les habitudes positives sont un des meilleurs moyens durables de devenir plus productif. Pour cela, l'application permettra à l'utilisateur d'entrer une liste d'habitudes chaque mois. Par la suite, celui-ci aura une notification chaque soir afin de valider son avancement de la journée dans ces habitudes. Tout le long du mois, un graphe permettra d'observer son avancement, permettant de savoir où l'on en est mais surtout motivant l'utilisateur dans sa démarche. De plus, les données seront stockées dans la base de données de l'application, et l'utilisateur pourra ainsi observer son évolution via des graphes pour tous les mois passés, les graphes permettant de voir l'avancement de manière précise et concise.

Mode travail

Le mode travail pourra être activé ou désactivé à tout moment depuis la toolbar. Celui-ci permet à l'utilisateur de se concentrer pour travailler et accomplir les choses sans être dérangé : il coupe le wifi, le son et les données cellulaires. Toutefois, les paramètres désactivés peuvent être précisés dans les paramètres, pour quelqu'un qui aurait besoin du wifi en travaillant par exemple. Toutefois, cette partie de l'application peut être susceptible de changer du fait que les nouvelles versions d'Android bloquent l'accès à certains de ces paramètres aux applications. Les préférences de l'utilisateur seront stockées dans la base de données.

Timer de travail

Le timer de travail permet d'organiser ses sessions de travail afin d'être le plus productif possible. Il applique la méthode Pomodoro : des sessions de travail de 25 minutes entrecoupées de pauses de 5 minutes – toutes les quatre sessions de 25 minutes, la pause dure 20 minutes. Cette méthode a déjà fait ses preuves et augmente l'efficacité et l'endurance au travail. Toutefois, l'utilisateur pourra paramétrer la durée des sessions de travail ou des pauses selon ses besoins. Chaque session pourra être mise en pause voire remise à zéro. A chaque début et fin de pause, une sonnerie retentit pour notifier l'utilisateur.

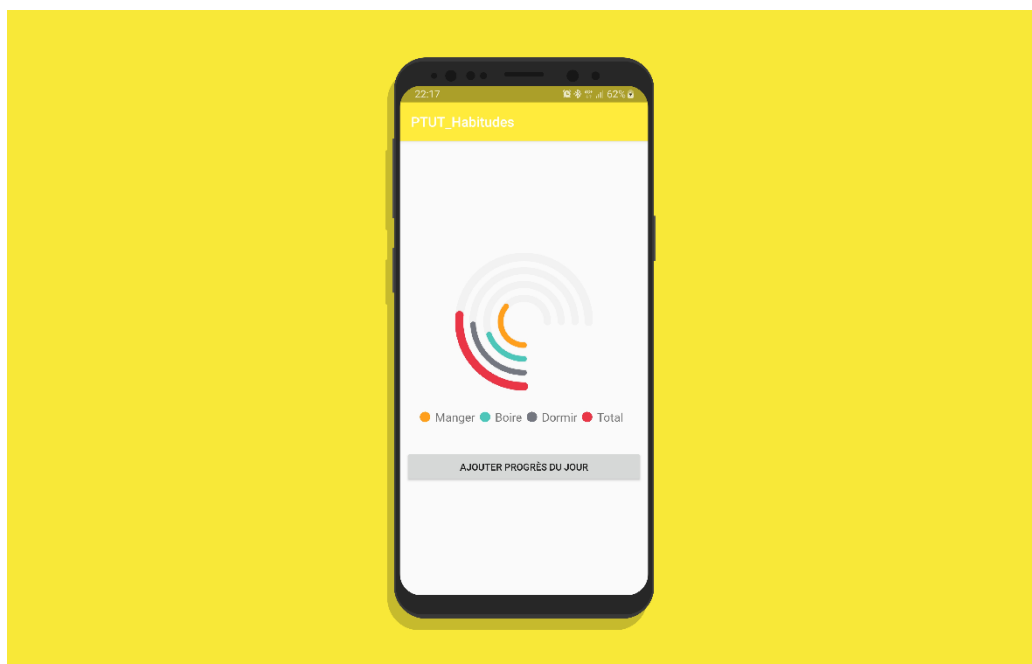
2. Charte graphique

Pour la charte graphique, nous avons décidé de suivre les guidelines de Google pour l'interface d'applications Android, et donc d'appliquer le Material Design.

La couleur dominante sera le jaune et le thème sera clair. Toutefois, ces préférences pourront être modifiées selon les préférences de l'utilisateur. Voici les couleurs choisies :

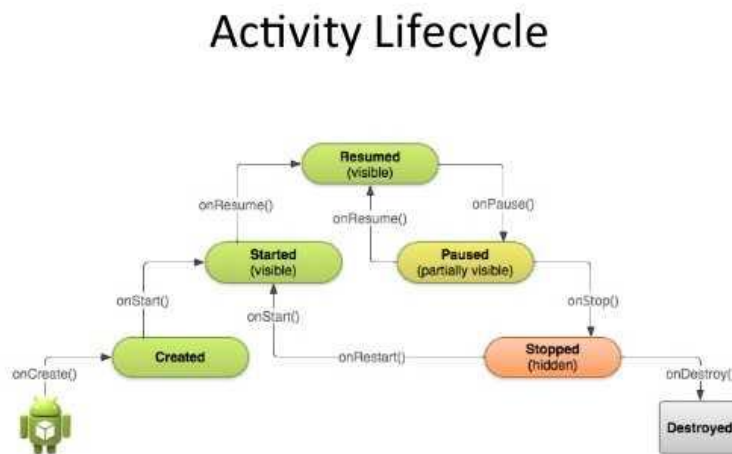
```
colors.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <color name="colorPrimary">#FFEB3B</color>
4      <color name="colorPrimaryDark">#AC9F29</color>
5      <color name="colorAccent">#D81B60</color>
6  </resources>
```

Le fichier colors.xml définit les valeurs hexadécimales des couleurs primaire, primaire sombre et d'accentuation de l'application. Elles sont centralisées dans un fichier .xml pour être communes à toute activité et facilement modifiables. Voici le rendu d'une activité avec ces valeurs :



Notion d'activité

L'activité est un élément clé de l'univers du développement Android. Concrètement, il s'agit d'une classe Java qui hérite de `AppCompatActivity` et correspond généralement à au moins une *View* (vue, fenêtre affichée sur l'écran de l'appareil mobile). On y programme la partie « métier » d'une vue, donc d'un écran. Une activité peut avoir plusieurs états :



*Différents états d'une activité Android.
Source : dev2qa.com*

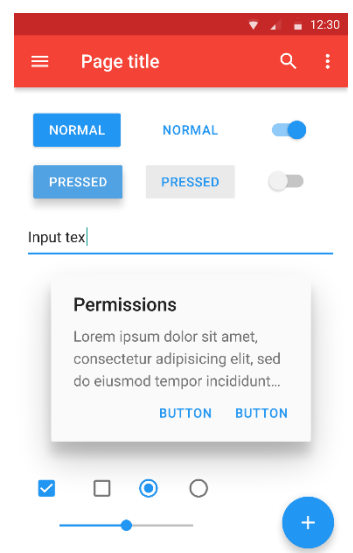
Une application Android est composée d'activités, qui fonctionnent de manière autonome, mais qui coexistent, se lancent les unes les autres et communiquent des données.

Qu'est-ce que le « material design » ?

Le material design est un ensemble de directives de design créées par Google. De base créé pour les mobiles, plus spécialement les applications de l'OS Android, il a été adopté par de nombreux acteurs majeurs du web et est aujourd'hui beaucoup appliqué dans le domaine du web aussi.

Il a été créé dans une période où le flat design (le contraire du skeuomorphisme, design où les formes sont réalistes) s'était imposé dans tous les domaines de l'informatique. Il s'en inspire donc grandement, en ajoutant un principe de « couches matérielles » - d'où son nom. Il repose donc sur des couches physiques, des gestes naturels et des formes simples, symétriques et en relief.

Ces guidelines de Google ont engendré une grande cohérence dans les interfaces des applications de son OS, et une expérience utilisateur généralement accrue du fait d'une cohérence générale et donc des fonctionnements connus de l'utilisateur, mais aussi du fait d'une affordance fortement supérieure par rapport au flat design en général. Ainsi, les interfaces créées sont beaucoup plus intuitives et confortables à l'utilisation.



Éléments graphiques d'une application Android en Material Design.

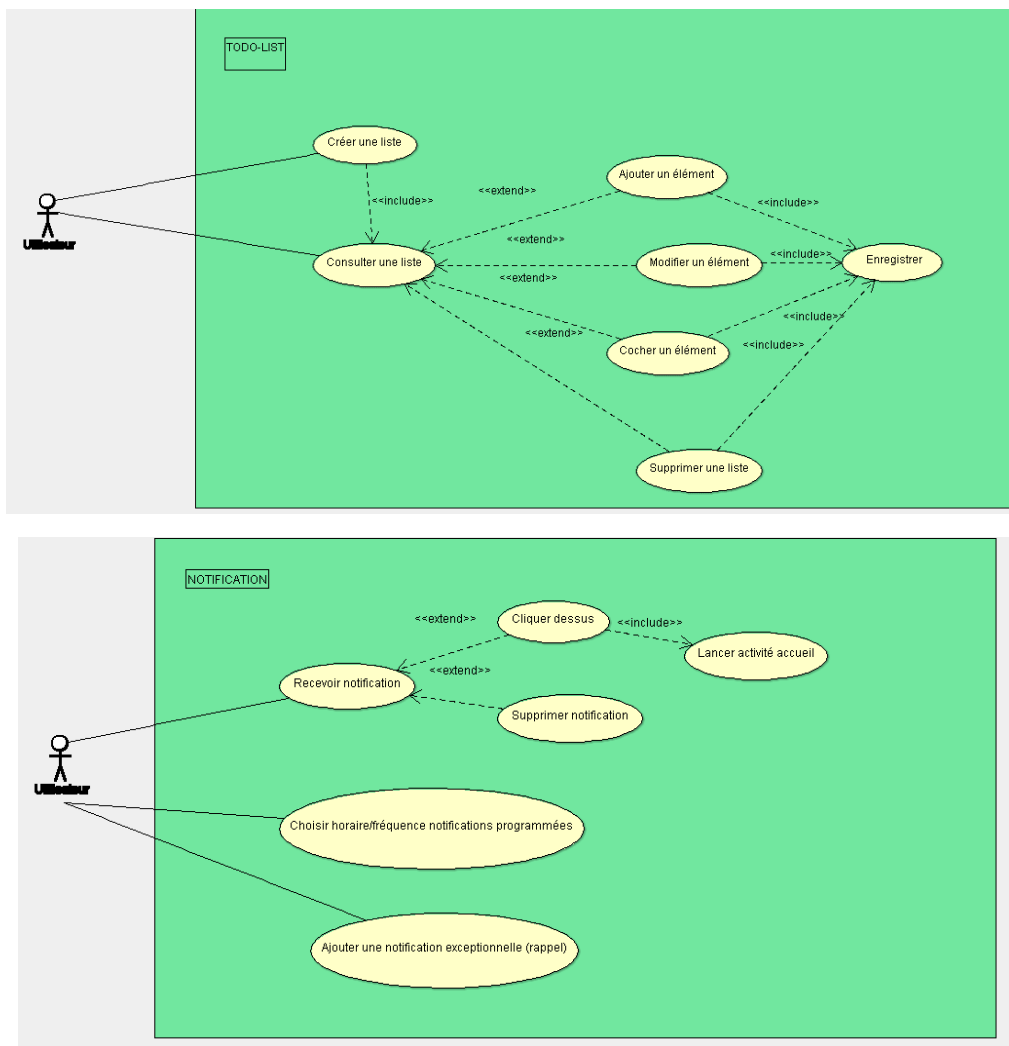
3. Implémentation : architecture, diagrammes et maquette

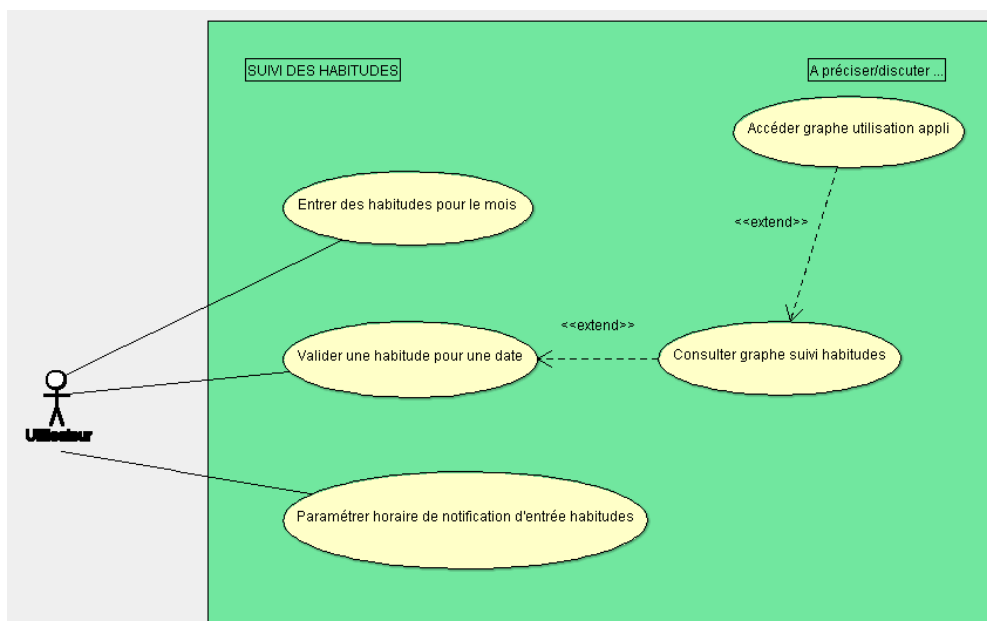
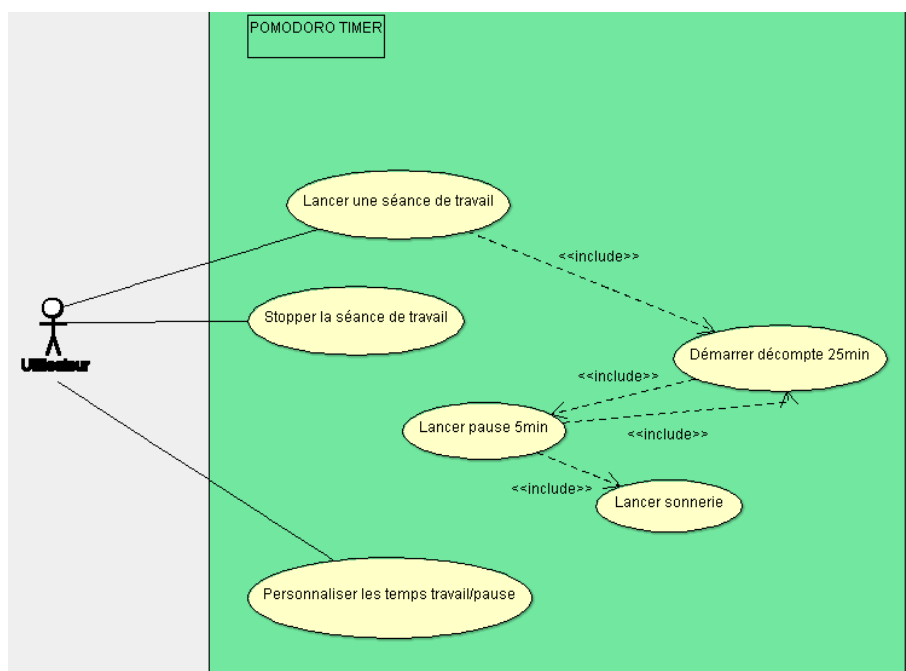
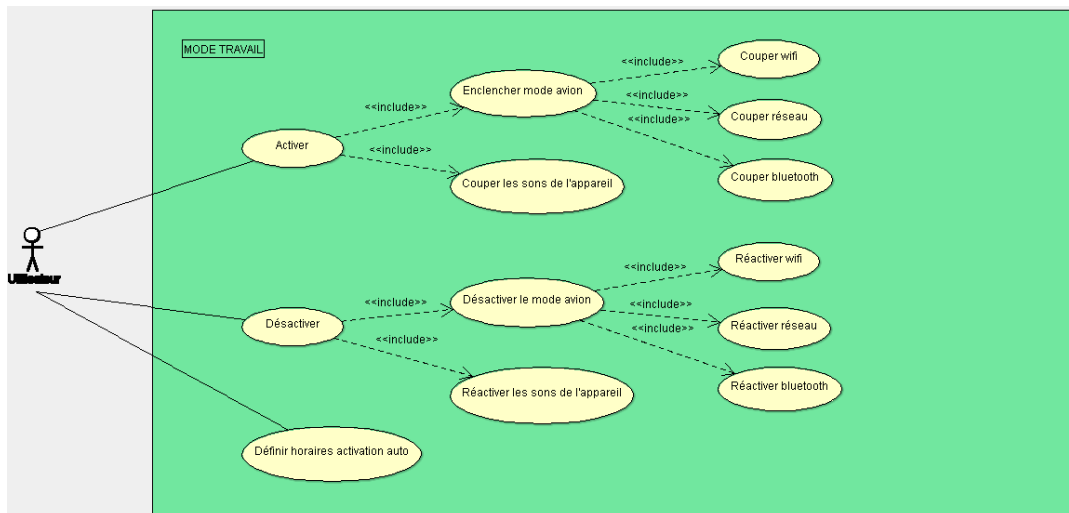
a) Architecture

Nous avons décidé de dédier une activité pour chaque application. La persistance des données se fera via une base de données stockée localement sur l'appareil. Les données étant stockées localement, nous avons fait le choix de ne pas implémenter de système d'utilisateur et donc de connexion à son compte.

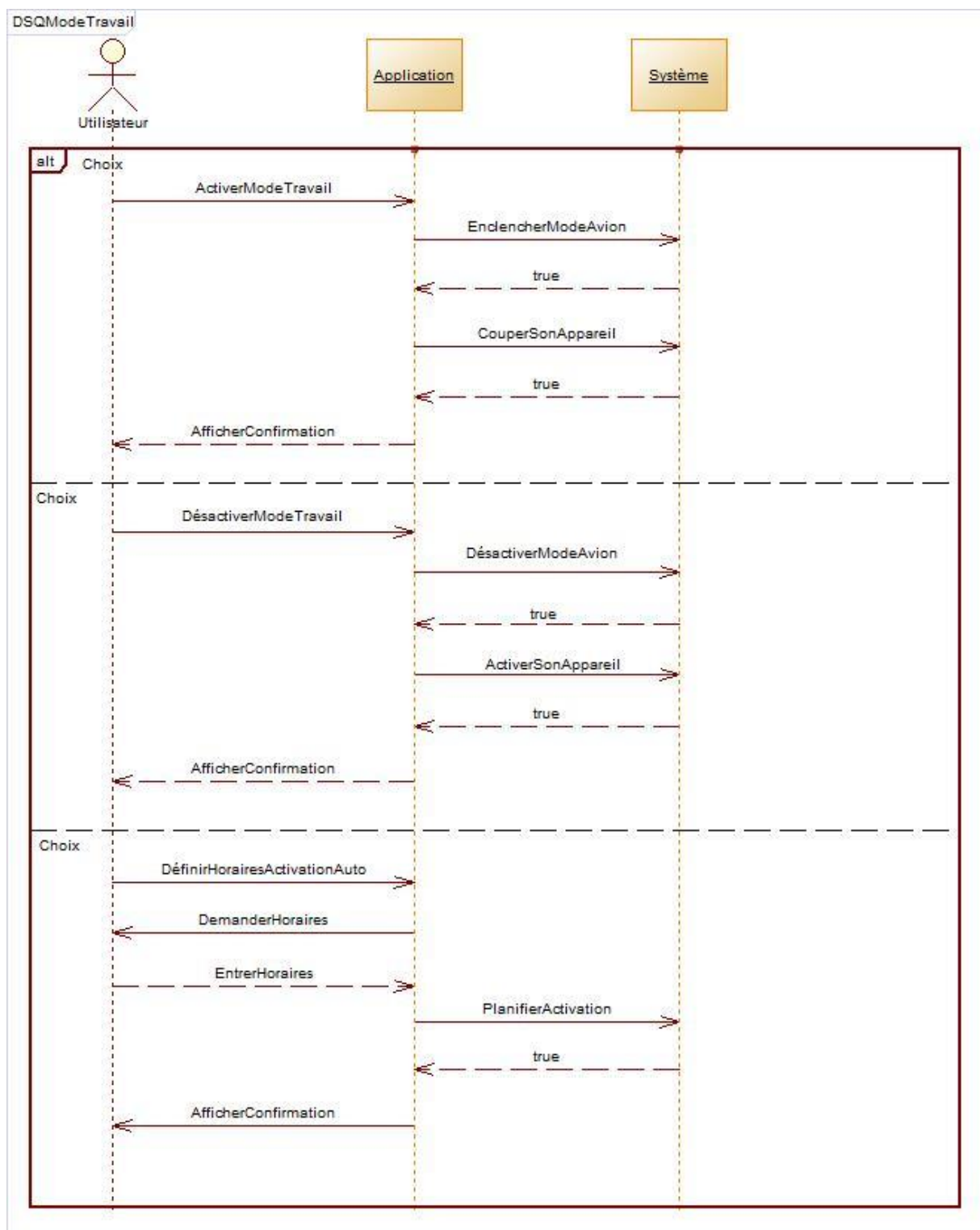
b) Diagrammes

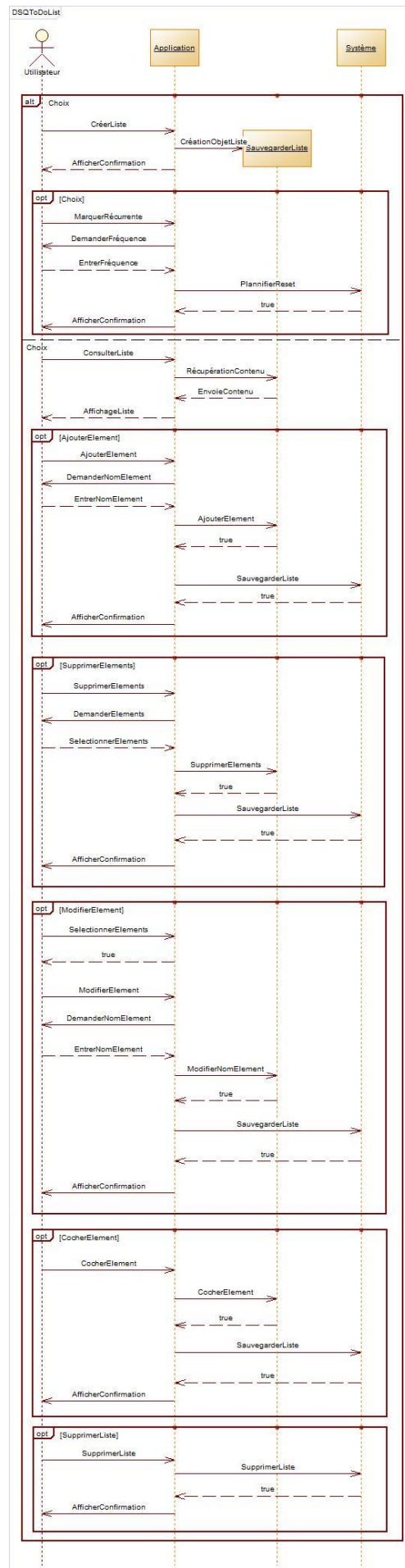
Voici d'abord les diagrammes de cas d'utilisation, décrivant les diverses fonctionnalités de l'application :

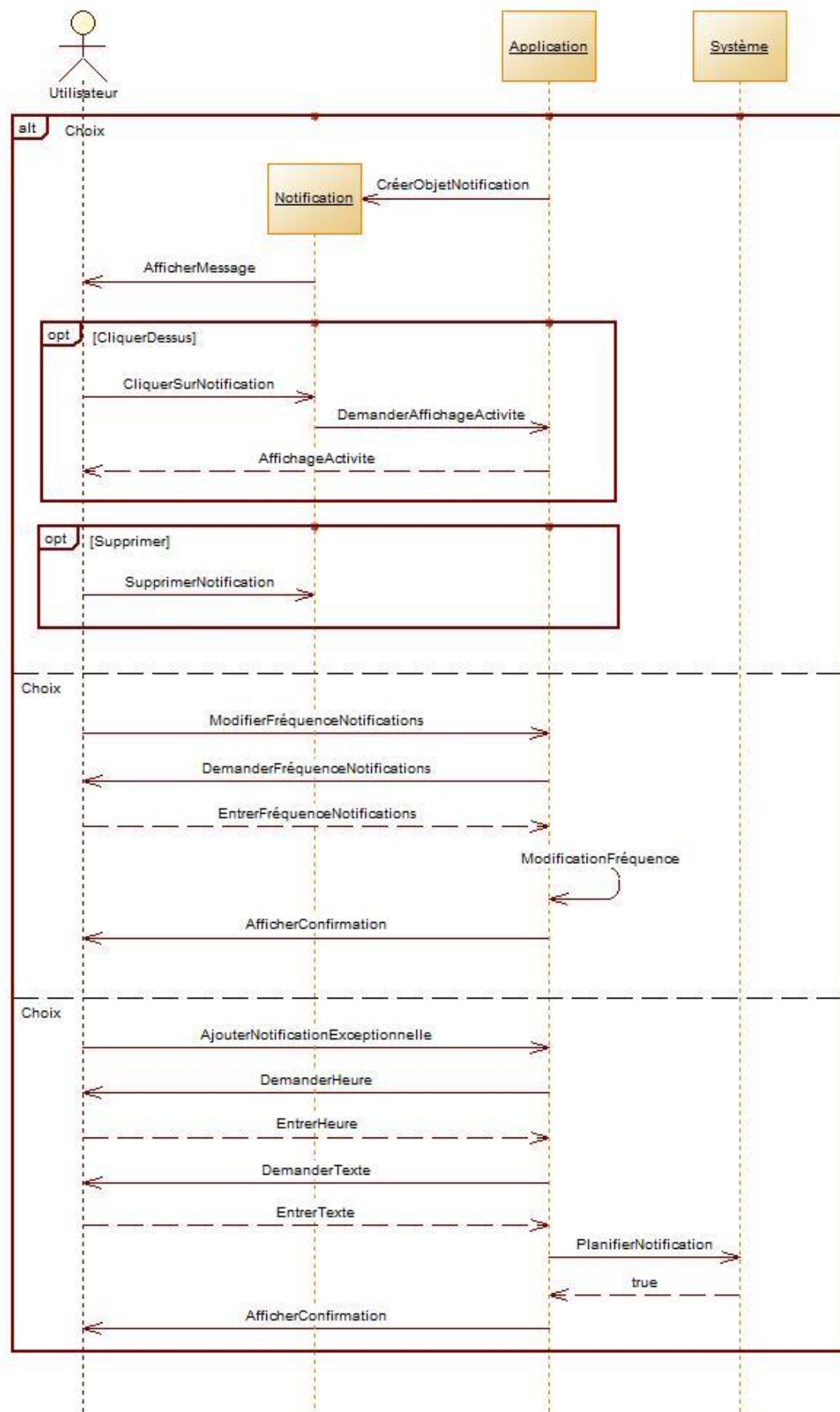


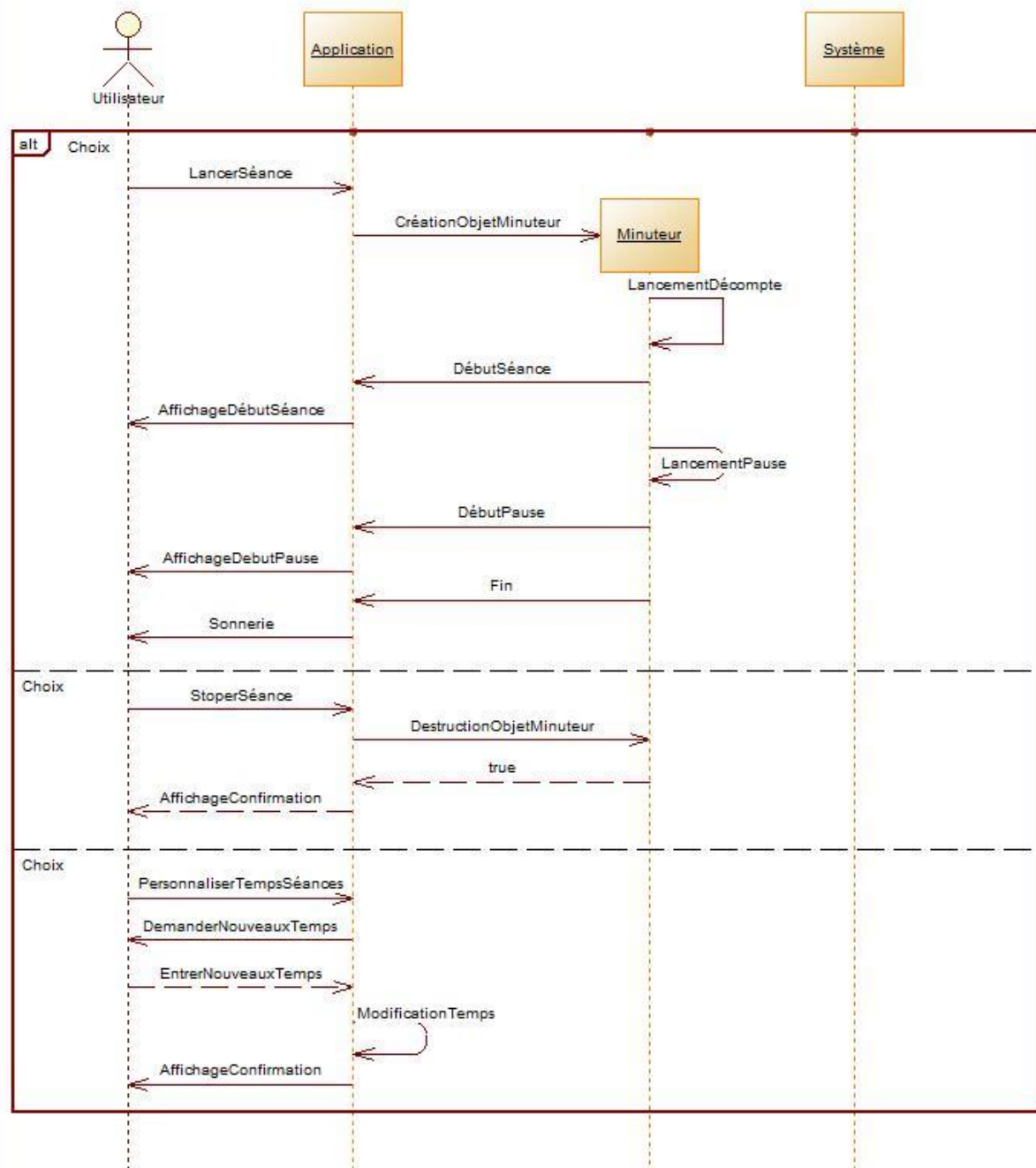


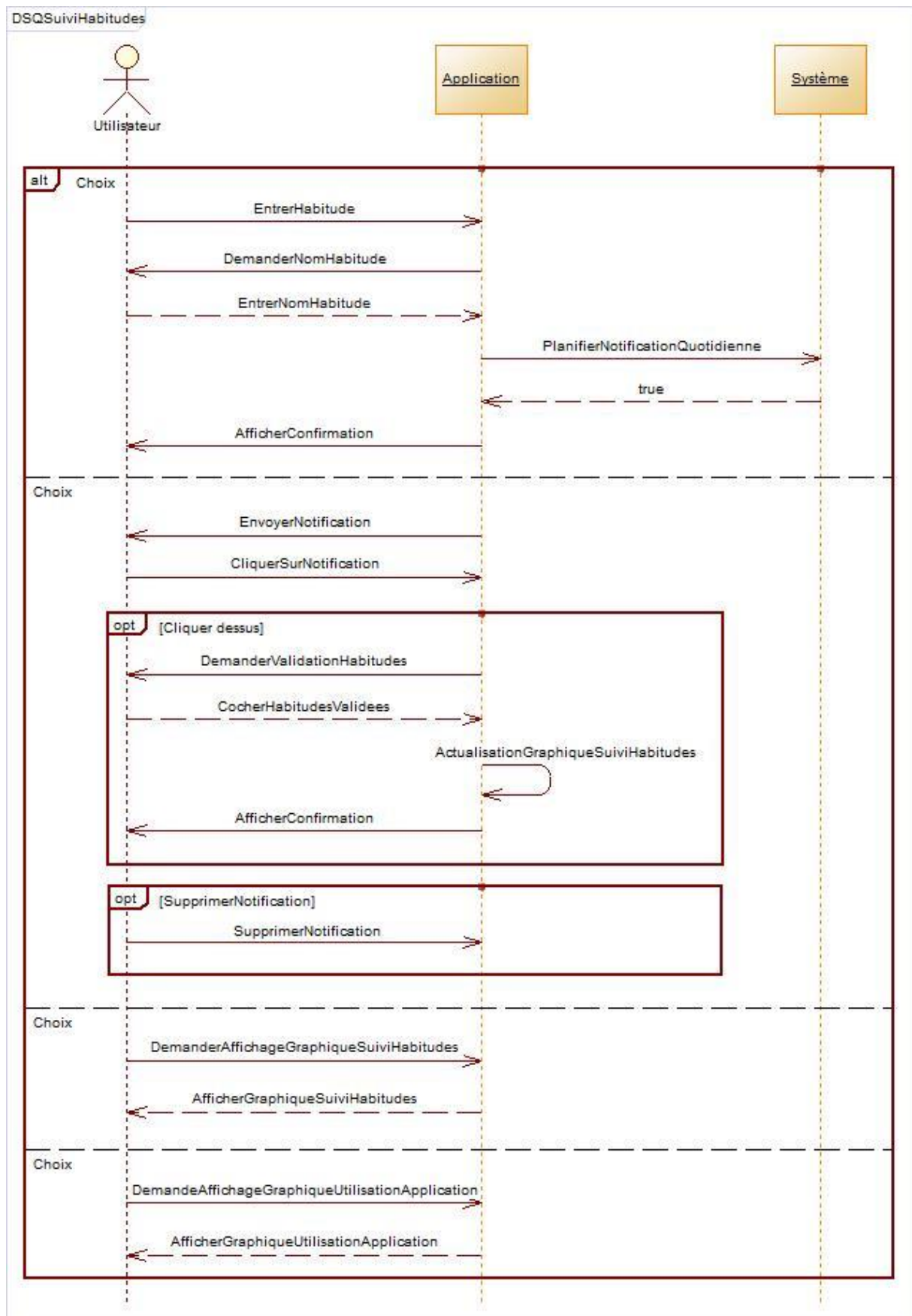
Voici les diagrammes de séquence :











c) Maquette

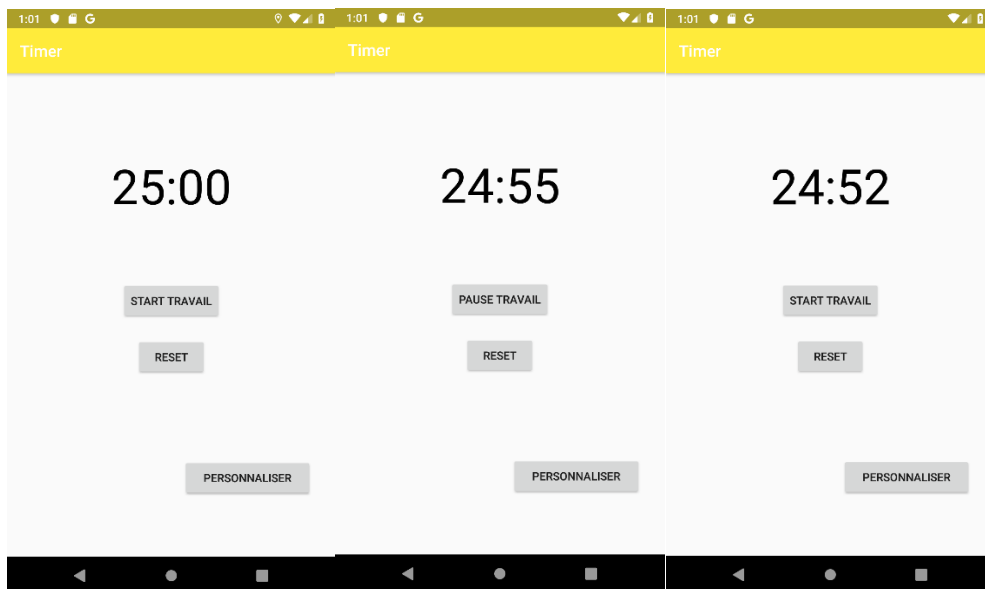
Pour l'élaboration d'une première maquette, nous avons fait le choix de nous répartir certaines des fonctionnalités envisagées et d'en prototyper une chacun, tout en mettant en commun ce que l'on a appris.

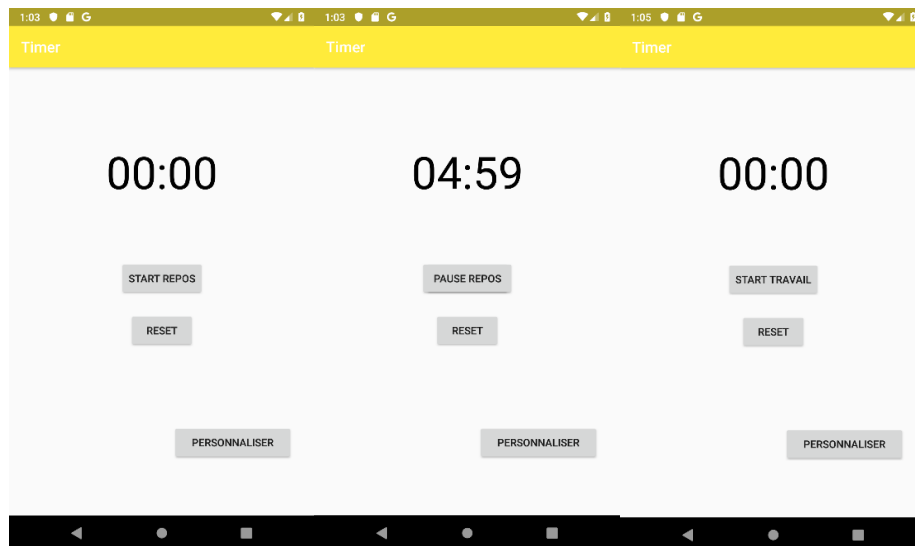
En effet, nous avons estimé que pour que chacun découvre et prenne en main le SDK Android, il serait préférable que chacun puisse développer un premier test à son rythme et de la manière dont il préfère. En faisant ainsi, nous nous sommes assurés du fait que chaque membre du groupe a déjà développé une application Android et comprend le fonctionnement de base du développement d'une telle application.

Le bilan a été très positif : malgré les appréhensions de certains membres, le développement s'est avéré faisable et surtout très agréable. En effet, l'avantage est qu'il y a très vite un résultat « tangible », ce qui motive et donne le sentiment d'avancer. Finalement, même si les maquettes ne reflèteront peut-être pas forcément le rendu final, cette approche nous a permis de rendre le développement moins intimidant et nous a rendu généralement non seulement plus confiants, mais aussi plus réalistes, car nous avons vite compris que certaines choses prévues seraient plus difficiles ou plus simples à réaliser que nous le pensions.

Vous trouverez par la suite des captures d'écran des différentes maquettes réalisées, fonctionnant donc chacune indépendamment.

Timer





Le timer a été élaboré de manière à que nous soyons le plus productif possible. Son fonctionnement est basé sur la méthode Pomodoro, ses effets étant expliqués dans l'étude suivante : [https://www.cell.com/neuron/fulltext/S0896-6273\(10\)00006-1](https://www.cell.com/neuron/fulltext/S0896-6273(10)00006-1)

Le temps est initialisé à 25 min ce qui correspond au temps de travail. L'utilisateur appuie sur le bouton "start travail" pour le mettre en route. Le timer décrémente jusqu'à arriver à 0. Une alarme sonne pour avertir l'utilisateur de la fin de sa session de travail.

L'utilisateur presse ensuite sur le bouton "start pause" pour commencer sa pause de 5 min.

De même, l'alarme s'active à la fin des 5 min. Ce processus est répété en boucle jusqu'à que l'utilisateur arrête.

L'utilisateur peut aussi appuyer sur le bouton "pause" qui permet d'arrêter le timer. Il peut ensuite activer le bouton "start travail" ou "start pause" suivant la session dans laquelle il se trouvait, pour qu'il reprenne sa décrémentation.

Un bouton personnalisé permet à l'utilisateur de choisir son temps de travail et son temps de pause.

À tout moment, l'utilisateur peut appuyer sur le bouton "reset" pour que le timer se réinitialise à la position fixe de départ, c'est à dire 25min.

To-do list

The image displays three sequential screenshots of a To-do list application interface.

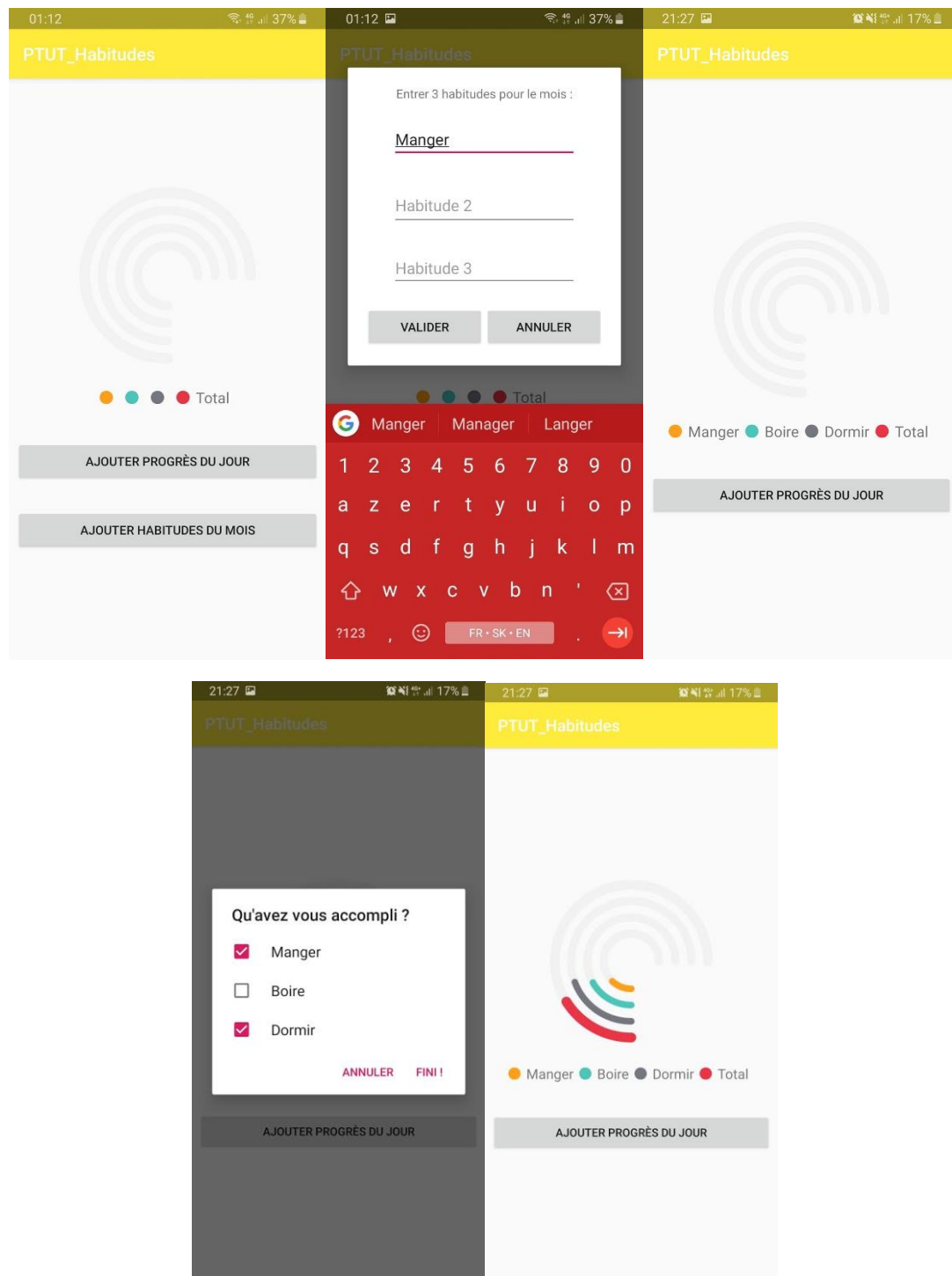
Screenshot 1 (Left): Shows the 'Liste des tâches' screen. It has a yellow header with the title and a '+' icon. The list contains four items: 'Tâche n°1', 'Tâche n°2', 'Tâche n°3', and 'Tâche n°4'.

Screenshot 2 (Middle): Shows the 'Ajouter une tâche' form. It has a yellow header with a close icon, the title, and a checkmark icon. The form includes a text input for 'Nom de la tâche', a date input for 'Date de fin de tâche', an unchecked checkbox for 'Me notifier ?', and a 'VALIDER' button.

Screenshot 3 (Right): Shows the 'Ajouter une tâche' form with the date selection calendar open. The 'Me notifier ?' checkbox is now checked. The calendar is for 'Janvier 1970' and shows the days of the week (S, M, T, W, T, F, S) and the dates (1 through 31). The date '5' is highlighted with a red circle. The 'VALIDER' button is at the bottom.

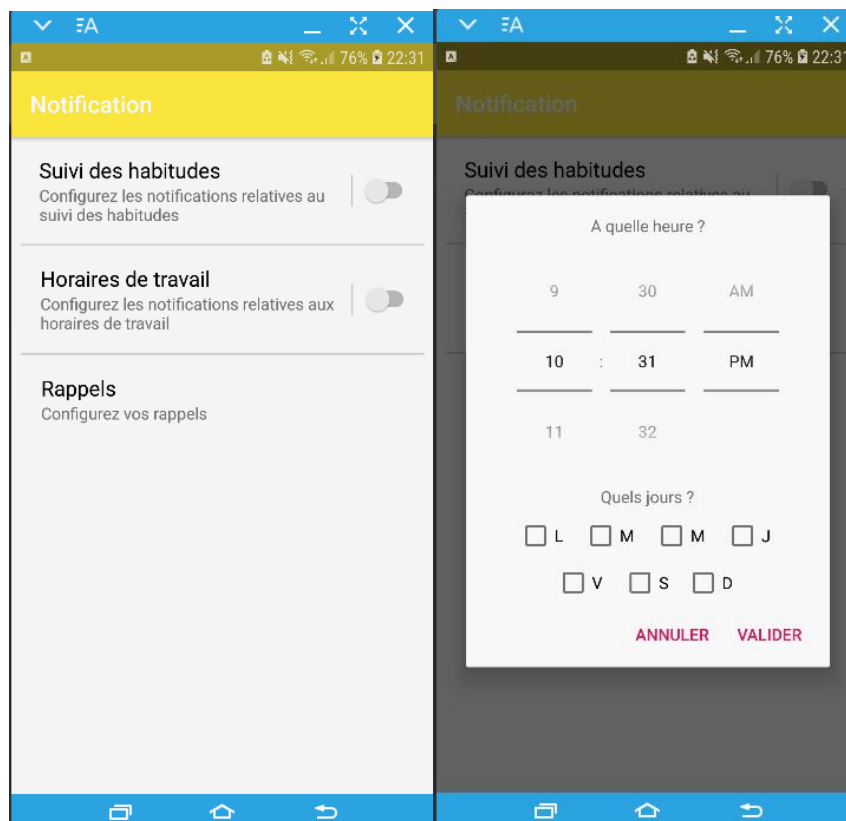
L'utilisateur peut, en créant une tâche, choisir une date où il sera notifié. Cela reste cependant optionnel, comme représenté sur les captures d'écran.

Graphe de suivi des habitudes



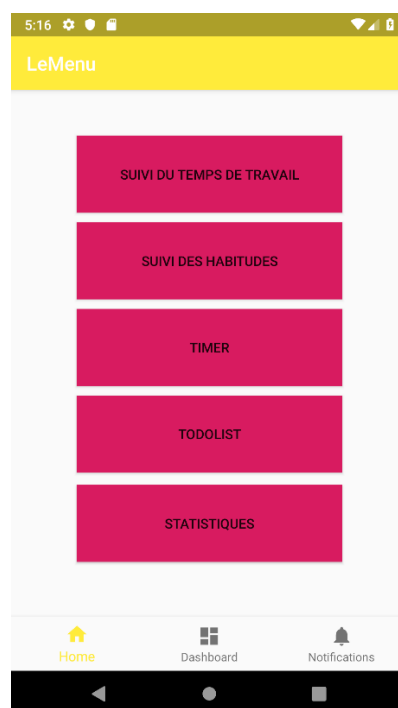
Le module de suivi des habitudes demandera les habitudes à suivre tous les mois. Ensuite, tous les jours, il sera rappelé à l'utilisateur de rentrer son avancement du jour via un Dialog. Puis, un graphe, accessible à tout moment, résumera l'avancement. Il y aura éventuellement une possibilité d'avoir des graphes différents.

Système de notifications



Le premier écran représente l'écran de paramétrage des notifications. Le deuxième permet de configurer en détail chacune des options proposées.

Menu



III. Le développement de l'application

1. Organisation et planning

Pour les réunions, nous avons en général fait une réunion toutes les trois semaines avec notre tuteur, lui présentant notre travail, en discutant pour l'améliorer et nous mettant d'accord avec son aide sur ce qui est à effectuer par la suite. Ces réunions régulières ont pu assurer un travail continu et ont fait qu'il n'y a eu aucune période de vide où le projet serait totalement délaissé.

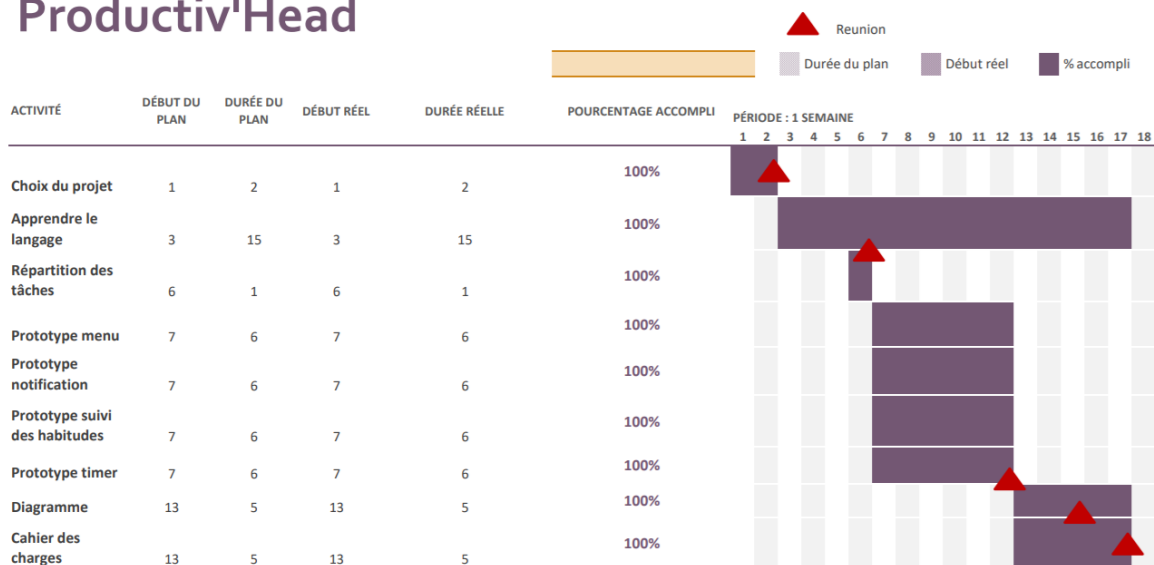
Toutefois, la collaboration a dû être assurée dans les périodes entre les réunions afin de garantir l'avancement du travail à présenter. Pour cela, nous avons mis en place un groupe de discussion qui nous a permis de nous organiser, de discuter, de mettre en place des rappels et des deadlines...

Toutefois, en plus de cela, pour « formaliser » l'organisation, nous avons mis en place un *Trello*, sorte de whiteboard avec des post-its en ligne. Nous nous en sommes servis pour clairement organiser les tâches à faire, réalisées, et qui s'occuperait de laquelle. Après chaque réunion, ce Trello a été mis à jour avec les différents fichiers, les nouvelles tâches que nous avons pu ainsi redistribuer sans rien oublier tout en restant flexibles et ouverts au changement. Cet outil a été réellement important pour notre organisation car nous ressentions le besoin d'avoir un endroit qui centraliserait tout ce qui est à faire de manière commune, accessible à tous et facilement adaptable.

Enfin, voici un diagramme de Gantt qui résume notre organisation et notre avancement au long de ce semestre, suivi d'un diagramme de Gantt de prévision pour le semestre à venir.

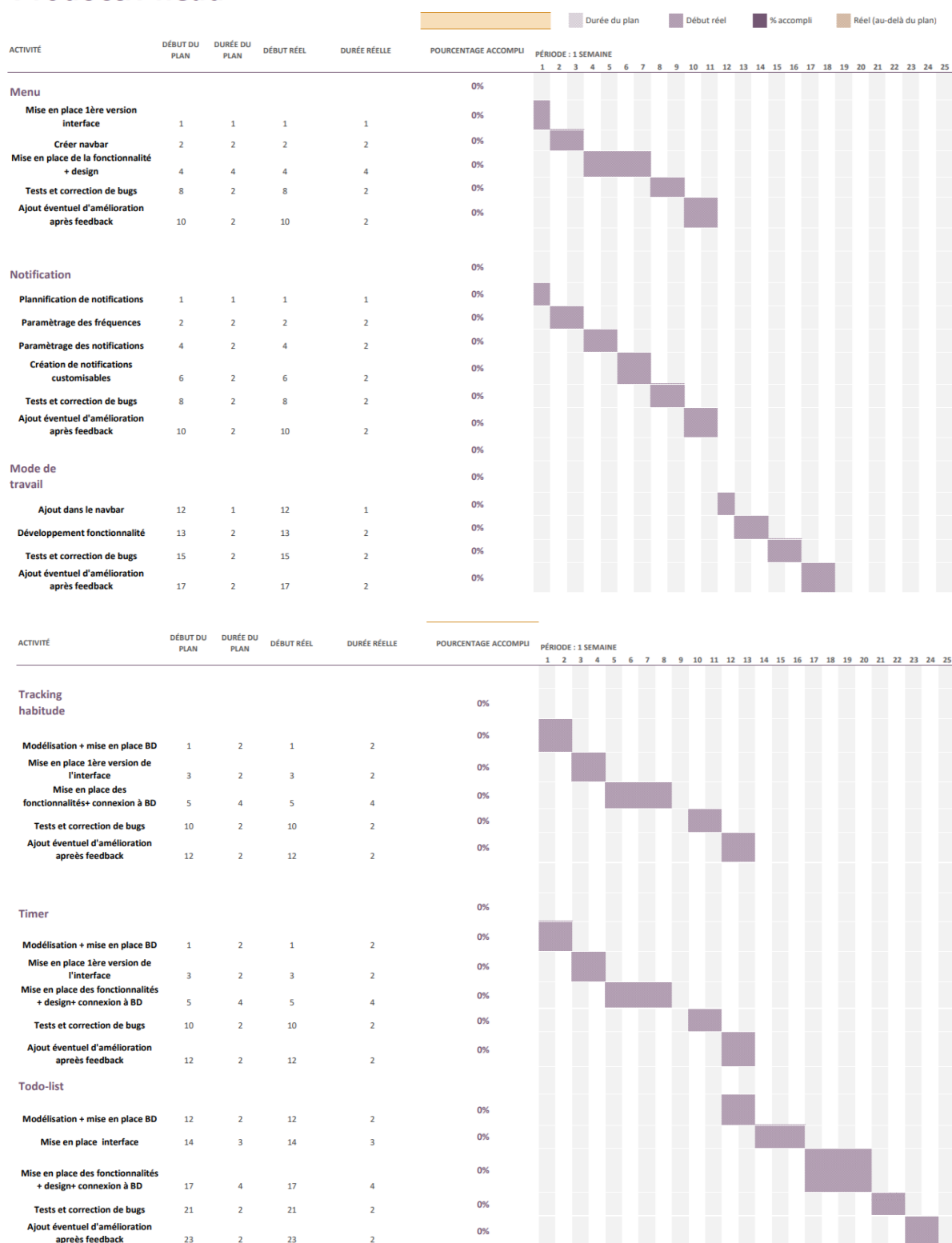
Semestre 2

Productiv'Head



Semestres 3 & 4

Productiv'head



2. Choix des outils utilisés

a) Les outils de collaboration

Plusieurs outils ont été mis en place afin de travailler ensemble de manière optimale. Comme évoqué auparavant, nous avons utilisé *Trello* à des fins d'organisation, tout en ayant un groupe *Messenger* permettant de mettre des rappels/deadlines communs et surtout de discuter vocalement et à l'écrit.

De plus, nous avons choisi *Git* comme outil de versionning et *Github* afin d'héberger notre dépôt distant.

Nous avons tout d'abord fait le choix de *Gitlab*, avec le serveur de l'université. Toutefois, avec du recul, nous avons estimé qu'il serait utile d'avoir le dépôt avec son historique même après notre DUT. Or, nous risquons de perdre notre compte de l'université par la suite, et c'est donc afin de conserver notre projet et des traces de ce qu'il nous a appris que nous avons décidé d'utiliser un dépôt public sur *Github*.



Logo de Github.

b) Les outils de modélisation

Pour modéliser nos diagrammes de cas d'utilisation, nous avons utilisé *ArgoUML*. Léger, open-source et multi-plateforme, celui-ci a été idéal pour permettre à tout le monde de travailler sur le même logiciel. Toutefois, certains membres ayant des habitudes et des connaissances sur Power AMC, les diagrammes de classe et de séquence ont été réalisés sur celui-ci grâce à la licence fournie par l'IUT.

Logo d'ArgoUML.

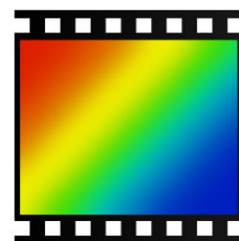


Les maquettes ont été développées sous *Android Studio* et mises en forme avec *Photofiltre7*.



Logo de Power AMC.

Enfin, les diagrammes de Gantt ont été réalisés sous *Microsoft Excel*.



Logo de photofiltre 7.

c) Les outils de développement

Pour le développement de l'application, nous avons fait face à plusieurs problématiques pour en arriver à nos choix.

Tout d'abord, nous devons nous décider sur les plateformes que nous viserons avec l'application : *Android*? *iOS*? *Les deux*?

Nous avons décidé de nous concentrer sur *Android*. En effet, faire une première application multi-plateforme mobiles aurait très probablement entraîné une interface peu adaptée et donc une mauvaise expérience utilisateur, devant nous adapter aux guidelines des deux univers très différents.

Or, le but étant de rendre l'expérience la meilleure possible pour améliorer la productivité, nous avons décidé de nous concentrer sur une application native *Android*, nous pliant au *material design*, nous donnant une UI et une UX cohérentes, intuitives pour l'utilisateur car il connaîtrait ainsi déjà les design templates *Android*. Ce choix étant fait, nous avons ensuite dû décider quels outils nous utiliserions.

Pour développer une application *Android* native, nous avons été contraints à utiliser le *SDK Android*. Plusieurs IDE sont alors envisageables : *Eclipse* avec un plugin, *IntelliJ* avec un plugin, ou *Android Studio*. Il s'agit d'une version d'*IntelliJ* de *Jetbrains* développée en collaboration avec *Google* servant spécifiquement au développement *Android*. Elle propose de nombreux outils de débogage, de design d'interface et autres extrêmement utiles et faisant que l'extrême majorité du développement *Android* se fait par cet environnement. C'est donc en sachant cela que nous avons décidé d'utiliser *Android Studio*.



Logo d'Android Studio

Enfin, nous avons eu à choisir le langage utilisé. Du côté du design de l'interface, pas le choix : tout se fait en *XML*. Toutefois, pour le reste, deux langages sont officiellement supportés par *Google* : *Java*, et depuis quelques années, *Kotlin*, un langage créé par *Jetbrains* tournant lui aussi sur la *JVM*. L'avantage est surtout la concision de sa syntaxe et son interopérabilité avec le *Java* : Il est possible d'écrire une partie en *Kotlin* et une partie en *Java* et tout marcherait parfaitement, et il est même possible, sous *Android Studio*, de convertir du code *Java* en code *Kotlin*.

Toutefois, ayant appris les grands concepts du *Java* au semestre 2, nous avons pensé qu'il s'agirait pour nous d'une bonne occasion de mettre en pratique concrètement ces notions de *Java* tout en les approfondissant et les comprenant réellement. En somme, choisir le *Java* semblait pour nous être une occasion d'ancrer ce que nous avons appris et faire de nos connaissances des compétences. C'est pour cela que nous avons choisi d'utiliser le *Java*, tout en souhaitant par la suite découvrir le *Kotlin* lorsque nous serons à l'aise avec le développement *Android* en *Java*.

Enfin, le choix du *Java* a été confirmé par le fait que la grande majorité de la documentation sur internet est écrite en *Java*. Bien que la documentation officielle de *Google* supporte le *Kotlin* aussi, la relativement récente officialisation du support du *Kotlin* fait que l'apprentissage du *SDK Android* sera plus aisé en *Java*, nous rendant plus de ressources accessibles.

3. Les contraintes rencontrées et à venir

Un premier problème que nous avons rencontré est le fait que, avec l'avancement des versions d'Android, la possibilité d'accéder aux paramètres de l'appareil a été beaucoup limitée pour des raisons de sécurité. Ainsi, les ambitions du « mode travail » ont dû être réduites et son développement risque d'être un peu plus technique que prévu.

De plus, la découverte des mécanismes du SDK Android n'a pas forcément été simple. Il s'agit de beaucoup de nouveaux concepts à comprendre et à prendre en main. Toutefois, de nombreuses notions d'IHM ont été retrouvées et certaines parties se sont aussi avérées plus abordables que prévues.

Enfin, nous avons au début fait l'erreur de ne pas mettre en place un dépôt git commun, ce qui nous a coûté de légères complications lors d'une réunion. Heureusement, cela est arrivé au début et n'a pas eu de conséquences, et nous aura bien montré l'importance de la rigueur à ce niveau-là.

Nous risquons bien évidemment de rencontrer de nombreuses nouvelles contraintes dans la phase de développement qui arrive, mais cela risque d'être extrêmement enrichissant et nous nous sommes organisés afin de toujours les mettre en commun pour tous en apprendre.