|  |  |
| --- | --- |
|  | ***PACT 2013-2014*** |

***Rapport d’avancement du groupe “PACT 4.1”***

***PAN 2: Cahier des Charges***

***“The Bumpers”***



*Date de mise à jour: 19/01/2014*

*Damien Auffert Charles Deschard*

*Arturo Castellanos Salinas Marc Faddoul*

*Cédric Meston Félix Richart*

*Julien Roméro Alexandre Tadros*

|  |  |
| --- | --- |
| ***Nom des encadrants:*** | ***Fonction:*** |
| Jean Le Feuvre | Tuteur |
| Mathieu Cholet | Encadrant informatique/Expert JAVA |
| Guillaume Duc | Expert électronique/GPIO |
| Tarik Ghraba | Expert électronique/GPIO |
| Jean Claude Dufourd | Expert Androïd |
| Thomas Robert | Expert Communication |
| Rémi Maniak | Expert SES |

***Sommaire:***

1. **Résumé du sujet choisi** (page 3)
2. **Summary** (page 4)
3. **Étude de marché** (page 5)
   1. **Étude d'antériorité**
   2. **Analyse des pistes de développement**
   3. **Réseau de valeurs**
   4. **Recherche de partenaires**
   5. **Sondage**
   6. **Business Model**
   7. **Parcours d’usage**
4. **Description fonctionnelle du prototype allégé** (page 16 )
   1. **Description littérale** (page 16)
   2. **Description fonctionnelle du prototype** (page 16)
   3. **Liste des Modules** (page 9)
   4. **Liste du matériel demandé** (page 10)
5. **Architecture informatique du prototype allégé** (page 21)
6. **Planification temporelle** (page 24)
7. **Perspectives pour le prototype final** (page 25)
8. **Bibliographie** (page 24)
9. **Annexes** ( PAGE 25)

**Annexe 1 : fiche d’identité du groupe** (page 25)

**Annexe 2 : fiches modules** (page 26)

**Annexe 3 : liste des modifications** (page 33)

***1. Résumé du sujet choisi:***

*Le reproche principal que l’on peut formuler à l’égard des réseaux sociaux et de la prolifération des Smartphones est un qu’ils détruisent le contact humain de proximité. Un paradoxe quand on pense que ces outils ont réussi a rapprocher des inconnus écartés à des milliers de kilomètres…*

*L’idée du BumpBand est simple. Il s’agit d’exploiter cette capacité incroyable des réseaux sociaux à rapprocher les inconnus mais à l’échelle d’un évènement festif. Il s’agit, via un bracelet ludique, surfant sur la vague des objets connectés utiles et du quantified self, de transformer le contact humain en composante d’un réseau social local. Ajoutez à cela un coté éphémère et une architecture qui protège efficacement la vie privée des utilisateurs et vous obtenez un objet amusant, pratique et qui peut ajouter une vraie valeur ajoutée à une soirée.*

*L’utilisateur doit simplement à mettre son BumpBand autour poignet et le synchroniser avec une application Smartphone préalablement téléchargée : il est prêt à l’emploi. Doté d’un petit écran LCD et pouvant être illuminé de la couleur souhaitée par l’utilisateur, on pourra communiquer avec les autres bumpers via un code couleur établi ou simplement le personnaliser à son goût. Deux personnes deviennent des BumpFriends (BF) par simple contact entre leurs bracelets, le Bump. Une fois ce geste effectué, ils auront accès à un certain nombre d’interactions : un service de messagerie qui durera le temps de l’événement et proposer à un BF de le rechercher par clignotement lumineux avec des couleurs spécifiques. De plus, chacun possède un fil d’actualité qui s’affiche sur le Smartphone et qui recense les activités des BF en temps réel.*

*Le gérant de la soirée peut envoyer un message à tous les Bumpers ainsi que lancer un Flash MoB (Flash Meet other Bumpers). Ce jeu a pour but de rassembler des participants qui ne se connaissent pas forcément. Lorsque le Flash MOB commence, 5 bumpers choisis aléatoirement se voient attribuer une même couleur. Il faut parvenir à bumper les 4 autres personnes avant la fin du jeu afin de gagner un crédit boisson. De cette manière les 5 Bumpers feront connaissance.*

*En plus de ces fonctionnalités ludiques, le BumpBand simplifie réellement la vie de ses utilisateurs. Avant l’événement, il est possible d’acheter la place via internet et de l’enregistrer sur le bracelet. Le passage par une file prioritaire permet alors d’entrer dans la salle via un bump signalant sa présence à l’entrée. De même plus besoin de ticket pour le vestiaire : un simple bump suffit à enregistrer son numéro. On peut enfin créditer un compte au bar au début de l’événement. On peut ensuite le débiter en prenant des consommations en bumpant au bar. On peut choisir sa boisson sur le Smartphone qui sélectionne automatiquement la couleur correspondante pour le bracelet. Le barman peut ainsi plus facilement savoir quelles boissons préparer et en quelles quantités. Finalement l’utilisateur peut suivre sa consommation approximative d’alcool et utiliser ses nouvelles relations pour partager un taxi pour rentrer, histoire de rentrer en sécurité.*

***2. Summary:***

*One of the main forms of criticism that can be directed at the overwhelming presence of Smartphones and the generalized use of Social Networks is that they destroy authentic face-to-face social interactions. A paradoxical reality when one considers how social networks have helped give birth to contacts between human beings thousands of miles apart…*

*The idea behind the BumpBand is simple. It aims to take advantage of social networks’ incredible ability to bring two perfect strangers to interact, by applying it at the scale of a party. Via a recreational bracelet, surfing on the current trends of connected objects and of the idea of the quantified self, it aims to convert physical interactions into components of the social networks’ architecture. Add to that idea a construction that guarantees user privacy, the ephemeral life span of the network and a series of practical apps and you get a entertaining object which can really enrich the user’s experience of the event.*

*The user simply wears the bracelet around his wrist and syncs it with a smartphone app he previously downloaded and it is ready to use. The BumpBand has an embedded LCD display and can be lit up with virtually any colour. The user can therefore communicate via an existing color code with other users or simply customize his bracelet. Two people become BumpFriends (BF’s) by bumping each other’s bracelets. Once they are BF’s, they will have access to a certain amount of features, which include a closed messaging system, and the possibility to request the two bracelets flash the same color, allowing BF’s to regroup. Furthermore the user can access a live News Feed, which will keep him informed of his BF’s activities.*

*The party’s organizer can also message all users and start a Flash MoB (Flash Meet other Bumpers). The game, which strives to bring bumpers to meet each other, randomly selects five people who aren’t already Bf’s and sets their bracelet a specific color. These 5 users will have to Bump in the set time to win free drinks.*

*In addition to these features, the BumpBand offers several apps that aim to ease the user’s experience. You can purchase tickets to the event online and simply Bump the bouncer to access the party. The bracelet can also save your coat hanger number, preventing the easy loss of a paper ticket. Finally it can be used to pay for drinks by bumping the bar and can use the subsequent data to analyze your alcohol consumption. From there, why not use your new social network to safely share a cab back home?*

***3. Étude de marché***

**A- Etude d’anteriorité**

Le marché des bracelets intelligents est en plein essor. Voici quelques uns des produits existants, classés suivant les fonctionnalités qu’ils partagent avec notre projet.

**- Des bracelets permettant de notifier à l’utilisateur les activités du Smartphone:**

**Embrace+** est un bracelet lumineux permettant de notifier l’utilisateur de tout ce qui se passe sur son Smartphone (réception d’appel, de messages, notifications Facebook, Twitter) en s’allumant d’une certaine couleur suivant un code couleur paramétrable.

<http://www.bitrebels.com/technology/tech-bracelet-notification-device/>

**LinkMe** pousse le principe plus loin en affichant les messages reçus sur un écran en diodes, tout en restant un device très simple du type de notre Party Band (différent de la montre Samsung).

<http://blog.laptopmag.com/linkme-launches-worlds-first-smart-message-wristband>

**- Des bracelets permettant d’établir une connexion par un bump:**

**L’Ushuaia Ibiza Hotel wristband** est unbracelet destiné à **un usage localisé** (ici un hôtel, pour nous une soirée).

Équipé d’une puce RFID, le bracelet distribué aux clients permet de se connecter directement à son compte Facebook en bumpant des bornes “like la soirée” ou de poster une photo à partir d’une borne photomaton.

Un public jeune et noctambule est visé pour les deux produits. (l’hôtel est situé à Ibiza au coeur de la vie nocturne).

<http://www.tnooz.com/article/rfid-wristbands-are-so-last-year-social-media-loving-hotel-connects-fingerprints-to-facebook/>

Ce type de bracelets peut s’acheter sur internet, personnalisables et programmables pour de nombreuses applications du type de l’Ibiza Hotel.

**- Des bracelets lumineux créant une ambiance de soirée et permettant de retrouver ses amis:**

Le **Wave Wristband** est un bracelet de soirée pouvant s’illuminer de différentes couleurs (LED). Le DJ peut synchroniser les couleurs de toute la salle pour créer une ambiance générale sur le dancefloor. (idée que l’on a eu puis abandonée pour laisser le temps de developper d’autres fonctionnalitées). Il permet également clignoter en phase avec un ami.

On retrouve dans notre projet l’idée de synchronisation lumineuse avec un bump friend, pour permettre de le retrouver dans une salle en recherchant un motif lumineux identique au sien.

Le public visé est le même que le notre.

<http://www.bitrebels.com/technology/social-wristband-light-accessory/>

**- Des devices permettant d’afficher une information personnelle par la lumière:**

Le projet PACT 2012 **Lumi-Shirt** permettait d’afficher un message dans un contexte de soirée. De la même manière, les bumpers pourront décider d’afficher aux autres qu’ils veulent danser, fumer ou partirgrâce au code couleur associé.

<http://pact.enst.fr/projet-2012-22/>

**- Des bracelets sur le principe du quantified self:**

De nombreux bracelets intelligents développent actuellement cet aspect, dont l’incontournable **Nike FuelBand**. Notre projet ce place sur ce créneau à travers le suivi de la consommation d’alcool.

<http://www.nike.com/us/en_us/c/nikeplus-fuelband>

Enfin on peut trouver tout simplement **Facebook** dans la liste des produits antérieurs au Bump Band. En effet, le projet crée un réseau social, avec un **fil d’actualité**. Le géant Américain sera certainement une source d’inspiration, notamment lors de la recherche de sources de revenus pour notre business model. (intégration de pubs, suggestion de boisson par les amis…)

Snapchat fonde son succès sur la recherche de diminution de l’empreinte numérique des utilisateurs jeunes de réseaux sociaux. Notre réseau social éphémère s’inscrit dans cette lignée.

**Conclusion de l’étude d’antériorité :**

Le Bump Band se place sur le créneau en plein essor des bracelets.

Cette forme de device semble adapté à un public jeune et noctambule (voir Wave wristband, Ushuaia Hotel).

Il combine des fonctionnalités existantes dans d’autres produits (notification de réception de messages, recherche d’amis, bump, quantified self, afficher des informations) en les adaptant à un contexte festif. Ce panel applicatif ludique est recherché du public, en témoigne la liste précédente de produits analogues.

La spécificité du Bump Band est de combiner un bracelet festif à un réseau social. L’un des reproches principaux que l’on peut formuler aux gadgets numériques et aux réseaux sociaux “traditionnels” est qu’ils détruisent le contact humain direct, alors que ce contact fait partie intégrante de l’organisation de notre réseau social (Bump). Il permet également de communiquer sans laisser de trace ni transmettre de coordonnées, ce qui est appréciable dans un contexte festif. On peut donc caractériser le réseau Bump de local et éphémère.

**B- Analyse des pistes de dévelopemment:**

Application iPhone: Avant de penser à lancer notre Bump Band à grande échelle, il faut développer une application iPhone, qui conserve une part importante du marché des smartphones en particulier au sein de notre clientèle cible.

Bateau de croisière, resort hôtel: On conserve la notion de réseau social local (le milieu est fermé) et éphémère (les rencontres ne dureront pour la majorité pas au delà de la durée des vacances, surtout qu’on rencontre des gens habitant à priori loin).

Le bracelet peut potentiellement communiquer avec le portable sur une distance de l’ordre de 100m. On peut donc laisser son smartphone dans la chambre et ajouter de nouveaux Bump Friend, ou payer ses consommations en se baladant en maillot de bain sur le bateau avec son bracelet waterproof au poignet.

Les atouts du bracelet, à savoir ses aspects ludique et convivial sont particulièrement pertinents sans un contexte de vacances.

La fonctionnalité de paiement est particulièrement intéressante (consommations au bord de la piscine) , avec une meilleure sécurité que de dire le numéro de chambre.

L'algorithme du flashMob peut être complexifié pour faire se rencontrer des gens d'âge proches (la population est bien plus hétérogène que dans une soirée).

Week-end d'intégration de grosses entreprises ou d’écoles:

la différence principale avec le bateau de croisière est que les relations ont pour objectif de durer. Il faut donc proposer un échange de numéro de téléphone en plus.

Forum: Nous avions au début pensé à vendre notre bracelet à des évènements tels que des forums d’entreprise. Cela n’est pas possible pour deux raisons:

- l’idée était d’adapter le principe d’un « match de proximité » : on fait correspondre les offres d’emploi aux CV des porteurs de bracelets lorsqu’ils passent à proximité d’un stand. Nous n’avons pas trouvé de solution technique pour réaliser une communication restreinte à un périmètre limité à une dizaine de mètres.

- l’intérêt de notre bracelet comparé à une simple application smartphone est son aspect ludique et festif : le bracelet est lumineux, le bump convivial. Ces atouts ne sont pas très pertinents dans un cadre de forum d’entreprise.

Application à une montre Samsung:

Potentiellement, on peut imaginer une version du Bump Band utilisable sur une montre Galaxy Gear.

L’écran couleur relativement large peut jouer le rôle des LED sur notre bracelet et offre une possibilité d’affichage supérieure au notre.

La Galaxy Gear est en effet équipée d’un gyroscope et d’un accéléromètre. On pourrait donc bumper en enregistrant le mouvement associé. Mais ce choix technique permet de plus d’enregistrer d’autres mouvements qui pourraient être associés à d’autres commandes. (on conserve l’aspect ludique, on ne sort pas son téléphone)

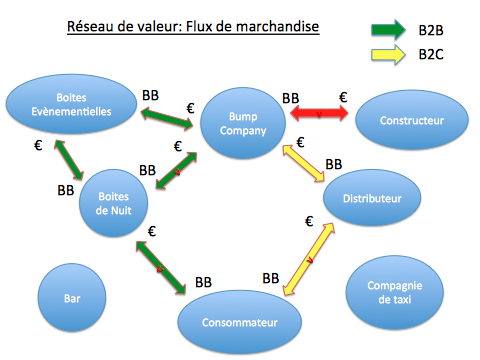
On donne alors une dimension supplémentaire au BumpBand pour pouvoir étendre son contenu applicatif, qui sera exploitée pas les consommateurs eux mêmes du fait du caractère Open Source.

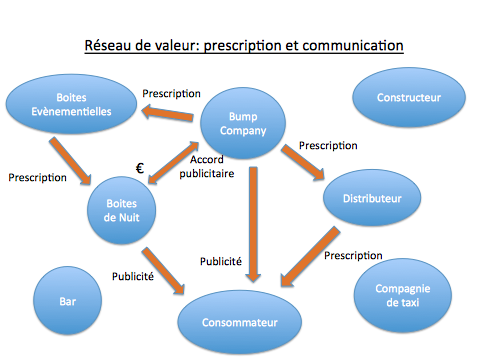
Enfin, le choix de JAVA comme langage de programmation rend le logiciel très portable, et donc facilement adaptable à la Galaxy Gear, mais également à d’autres montres de ce type qui pourraient gagner brutalement une grosse part de ce marché encore naissant.

On doit en effet garder à l’esprit que pour l’instant, la Galaxy Gear ne rencontre pas un franc succès. ( voir [cet article](http://obsession.nouvelobs.com/high-tech/20131120.OBS6148/galaxy-gear-la-montre-connectee-de-samsung-fait-elle-un-flop.html) du nouvelObs)

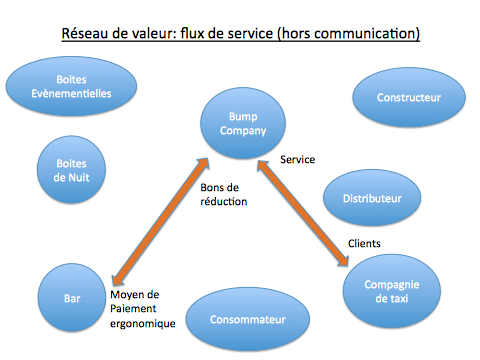
**C- Réseau de valeurs:**

La partie qui suit constitue une première version du réseau de valeurs de notre business plan, avec pour hypothèse que nous lançons le Bump Band pour le marché des boites de nuit.





Même si la prescription et la publicité font partie des services, ils sont traités dans un réseau à part par souci de lisibilité.



**D- Recherche de partenaires:**

Afin de vendre notre bracelet, deux possibilités s’offrent à nous:

Nous pouvons soit le vendre en B2B, soit en B2B:

**B2C:** La première possibilité serait de vendre notre bracelet aux particuliers. La difficulté principale est qu’il faut que la majorité des individus qui participent à un évènement possède le bracelet. Il faut donc une vaste campagne de publicité avant le lancement, en passant par les boites de nuit ou d'événementiel par exemple.

On peut imaginer concentrer ses efforts sur une seule ville au départ, puisqu’il faut que la concentration de bracelets soit importante pour que le BumpBand ait un intérêt.

En plus du coût important de la publicité, le risque que le bracelet ne soit pas acheté en nombre suffisant pour créer l’engouement est relativement important.

Par conséquent la prise de risque est forte et il semble préférable de vendre notre bracelet en B2B.

**B2B:** La seconde possibilité est de vendre le bracelet en grand nombre aux organisateurs d’évènements (type boite de nuit) qui ensuite mettrons à dispositions leurs bracelets aux participants (sous réserve de payement?) . L’avantage principal est que la majorité des gens auront le bracelet. Il faut alors réaliser des partenariats avec les organisateurs d’évènements. C’est pourquoi j’ai contacté par mail douze boites de nuit dans toute la France afin d’avoir leur avis et de leur proposer un partenariat si le projet est commercialisé. J'attends toujours leurs réponses (mails envoyés ce mercredi). D’un point de vue de publicité, les dépenses sont nettement réduites: il suffit de vendre les bracelets aux organisateurs d’évènements qui se chargeront eux même de faire connaître notre produit auprès des consommateurs.

Développement à l’international: Si le BumpBand est une réussite en France, il faut ensuite le commercialiser à l’étranger. L'absence de taxes douanières au sein de l’espace Schengen nous incitent alors à tout d’abord vendre notre produit dans les pays proches de nous et très développés (le public visé est riche), c’est à dire dans un premier temps l’Angleterre, l’Espagne, l’Allemagne, l’Italie, la Suisse, la Belgique, etc… Il faudra alors traduire notre application, qui pourra être téléchargée partout dans le monde. C’est également à ce moment qu’il faudra contacter d’autres services (tels les bateaux de croisières) afin de s’étendre à d’autres marchés.

**E- Réalisation d’un sondage:**

Afin de pouvoir finir notre étude de marché, j’ai réalisé un sondage durant 4 jours sur internet grâce à SurveyMonkey. Le partage du sondage s’est fait “d’amis à amis” sur Facebook et comptabilise au total 230 participants. Les questions étaient variées, allant de l’utilité et des applications du bracelet lui même au prix maximal que le participant acceptent de payer et au choix entre B2B et B2C. Voici quelques résultats intéressants:

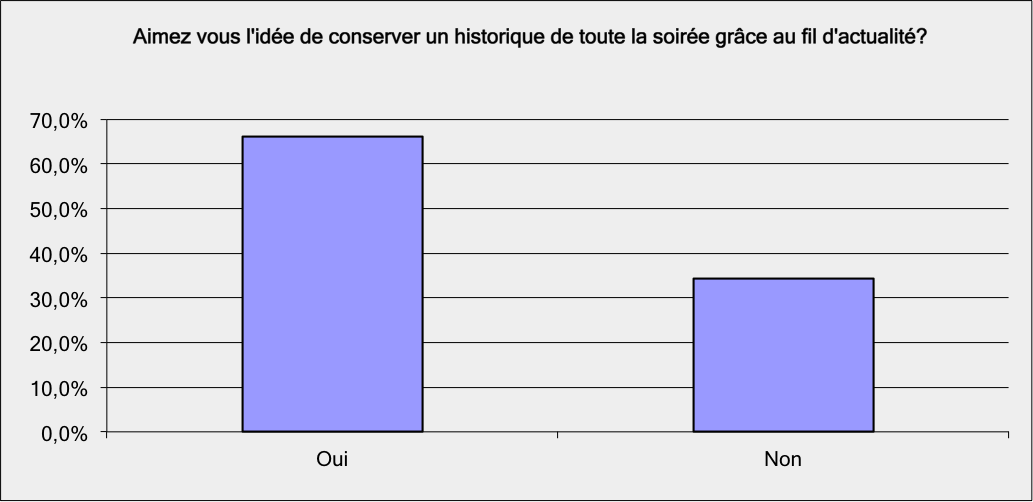
*Remarque sur les statistiques du sondage:*

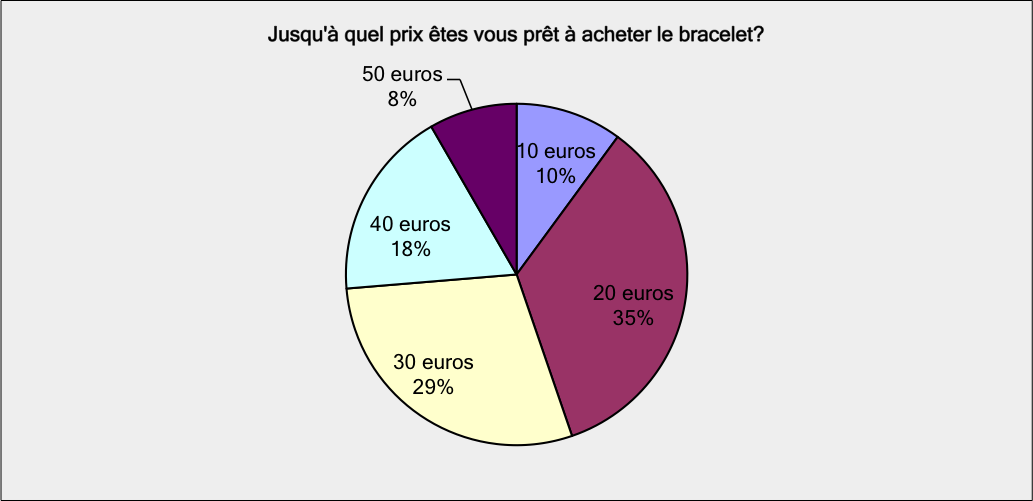
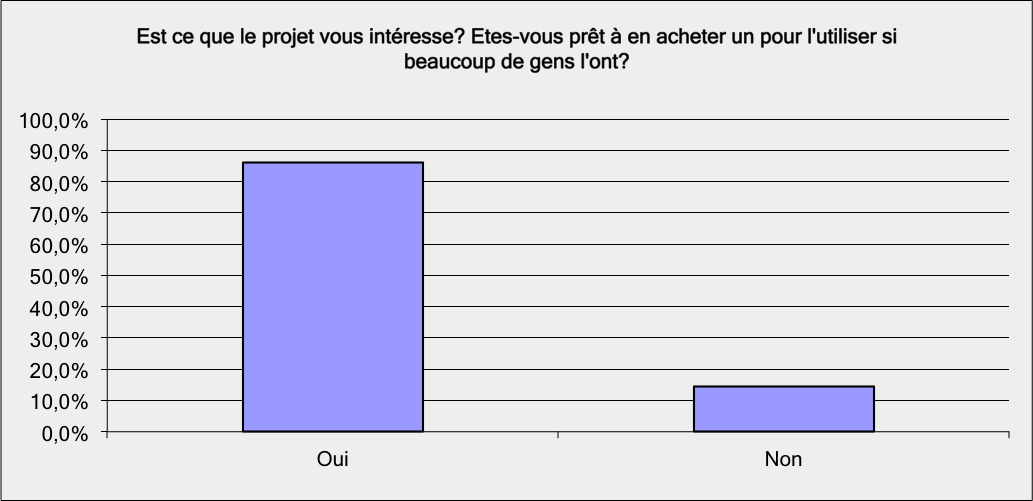
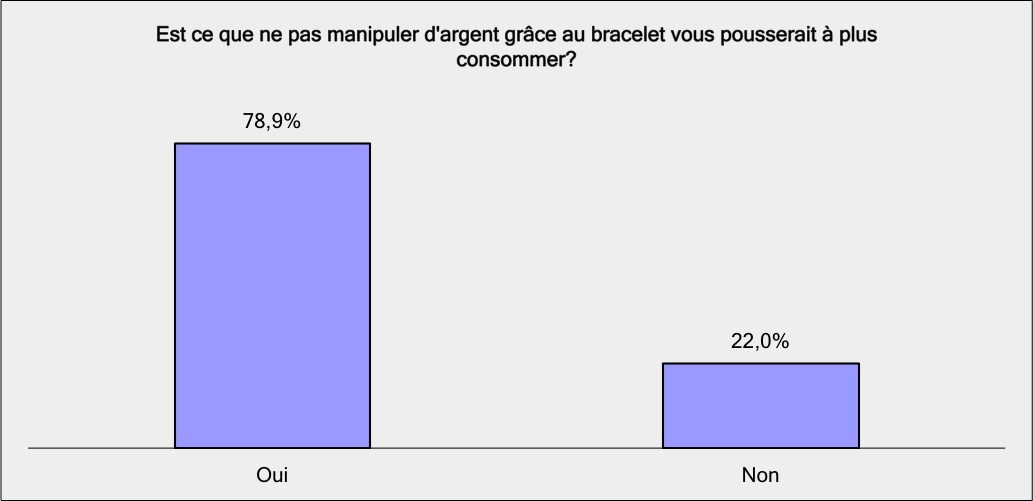
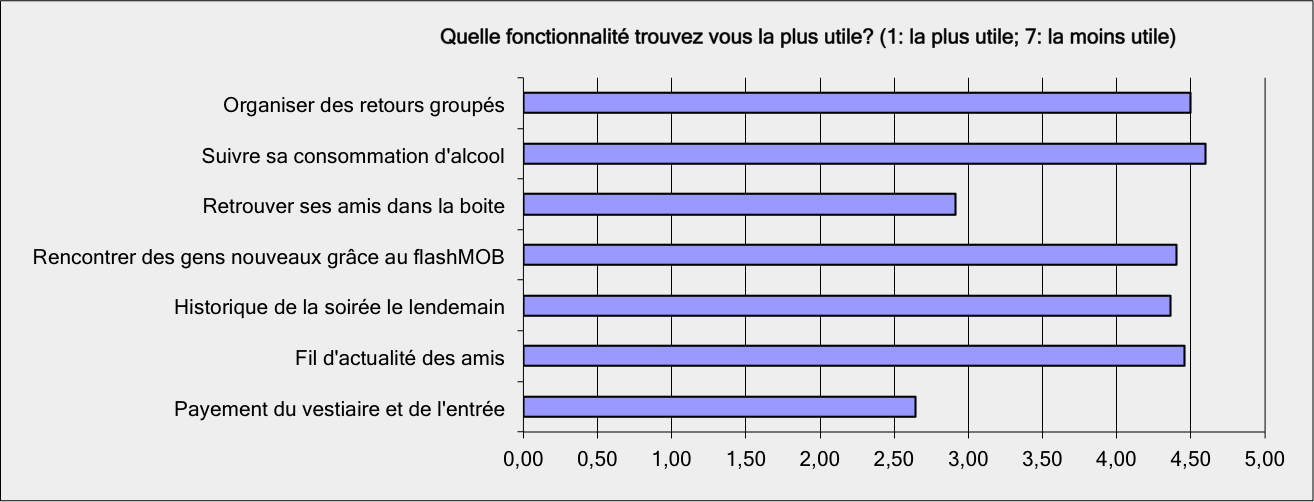
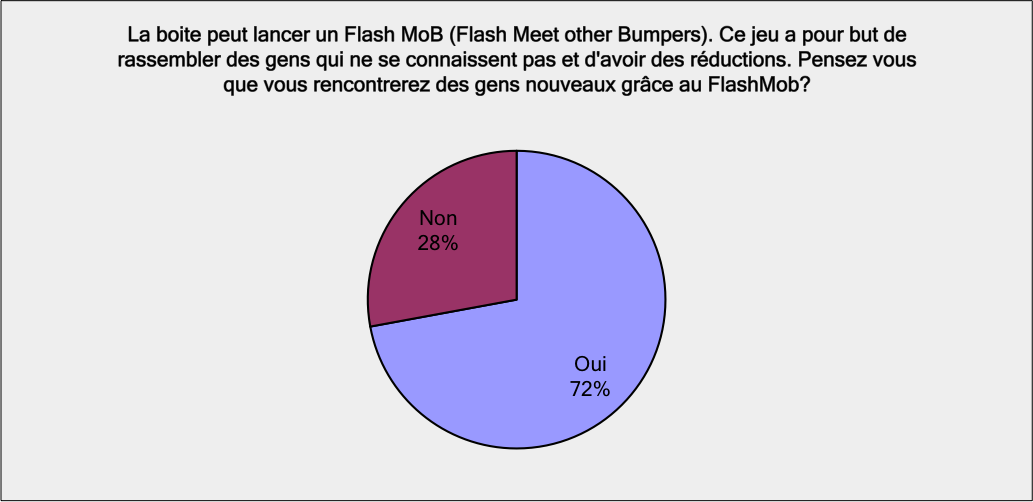
*229 participants, 41,52% de filles (93).*

*209 participants avaient entre 18 et 23 ans (91%).*

*8% avaient plus de 24 ans.*

*1% avaient moins de 18 ans.*





Conséquences du sondage sur notre projet:

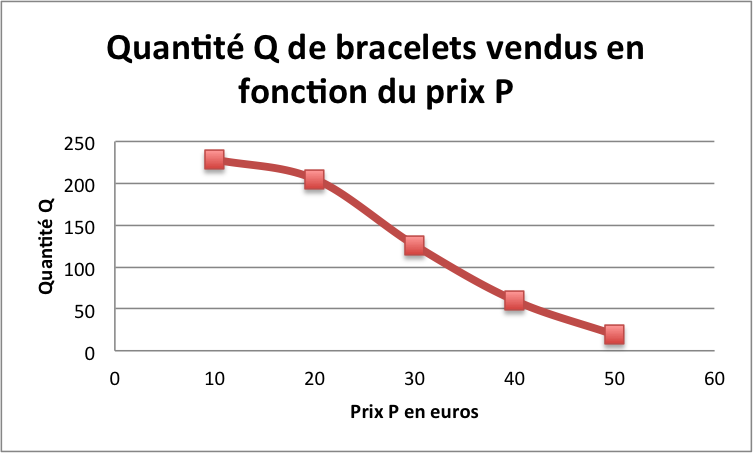
* Globalement notre projet semble très apprécié par les participants. Près de 85% seraient prêt à en acheter un. Nous avons même reçu des mails de personnes souhaitant nous aider à réaliser notre projet. (Nous avons du décliner leurs propositions dans le cadre du PACT).
* La conservation de l’historique de l’évènement semble être très appréciée. Il faudra donc la réaliser. (On s’en occupera pour le PAN4)
* L’idée du flashMOB est très appréciée, même si l’application la plus populaire semble être le payement de l’entrée et du vestiaire.
* Les participants préféraient posséder le bracelet (B2C) plutôt que les organisateurs leurs mettent à disposition (B2B). Cependant, nous avons vu que le B2B semble être une meilleure stratégie de vente. De plus, 79% des participants pensent plus consommer s’ils ne manipulent pas d’argent liquide, ce qui est un argument solide pour obtenir un partenariat avec des organisateur d’évènements.
* Le dernier résultats présenté si dessus est de loin le plus important. Il va nous permettre dans la suite de terminer l’étude de marché et ensuite de réaliser le Business Model:

**F- Business model**

Pour conclure notre étude de marché, nous allons réponde à la question suivante:

Combien de bracelets vendrons nous selon leur prix? On cherche un prix maximal idéal qui cherche à maximiser le chiffre d’affaire sans maximiser le revenu (dans un premier temps). On suppose que le coût de production d’un bracelet est de 10 euros (approximation grossière mais d’après le prix des composants ce sera à peu près ça.

Exploitation des résultats de la dernière question présentée si dessus:



Revenus = Chiffre d’affaire - Coûts

Coûts = Coûts fixes + Coûts variables

Coûts fixes = frais de ports, main d’oeuvre

Coûts variables = prix des éléments du bracelets

Dans la suite on suppose: (approximation)

Cf= 0

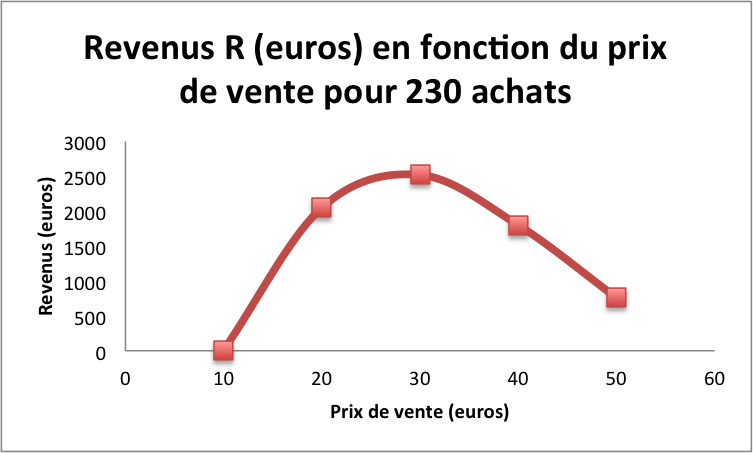
Cv= 10 euros par bracelet = 10\*Q

Conséquences:

On peut écrire: R= Q\*P - (Cf + Cv\*Q)

R= Q\*P - 10\*Q

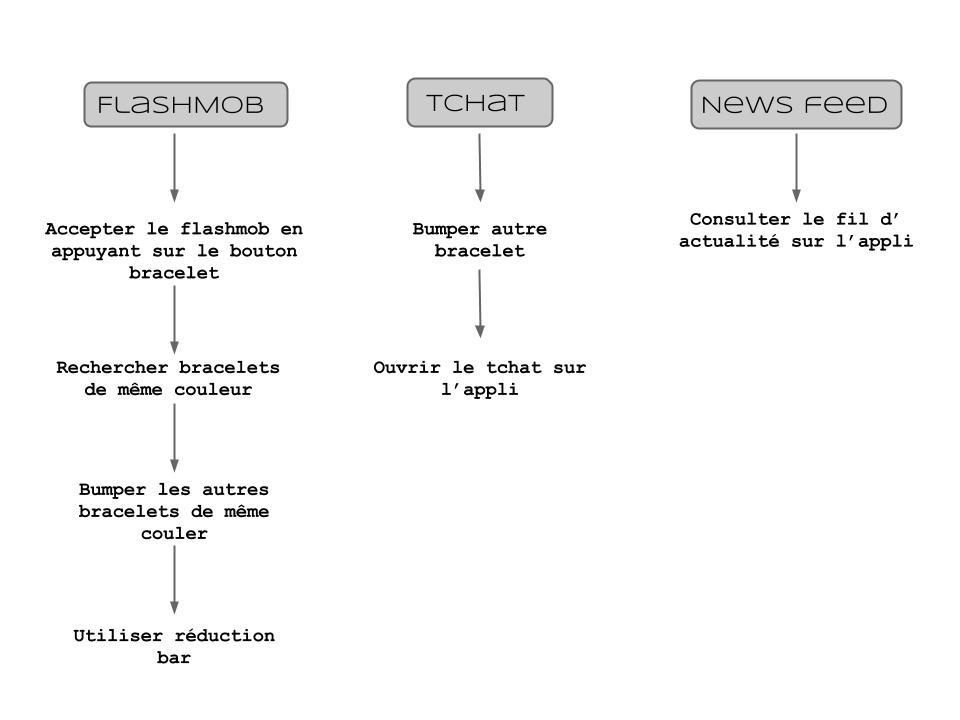
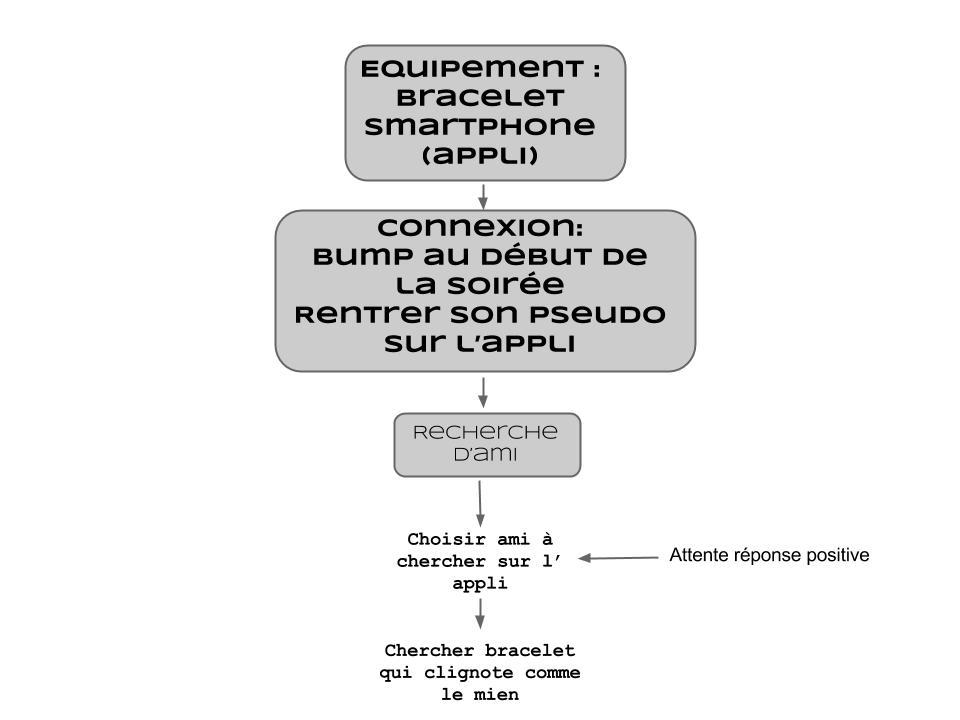
D’où le graphique suivant:

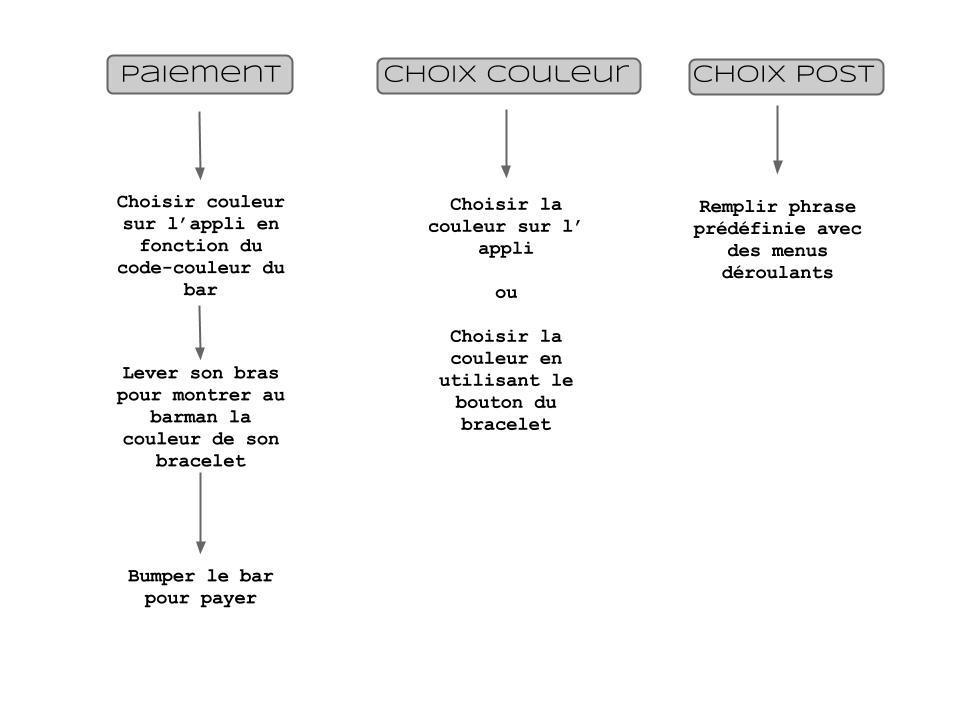


Conclusion : Il faut vendre notre bracelet à un prix approximatif de 30 euros afin de maximiser le revenu. Afin d’obtenir un prix plus précis, il faut obtenir précisément la valeurs des coûts fixes et variables.

Remarque: Cette étude n’est vrai dans le cas d’un modèle de vente de B2C. Si nous souhaitons vendre notre bracelet en B2B, il faut réaliser cette même étude avec des organisateurs, ce qui n’a pas pu être réalisé dans le cadre du PACT (il est difficile d’obtenir ces informations pour le cas de boites de nuit: ils ne peuvent pas savoir combien de bracelets ils achèteraient, et pour quel prix).

**F- Parcours d’usage**





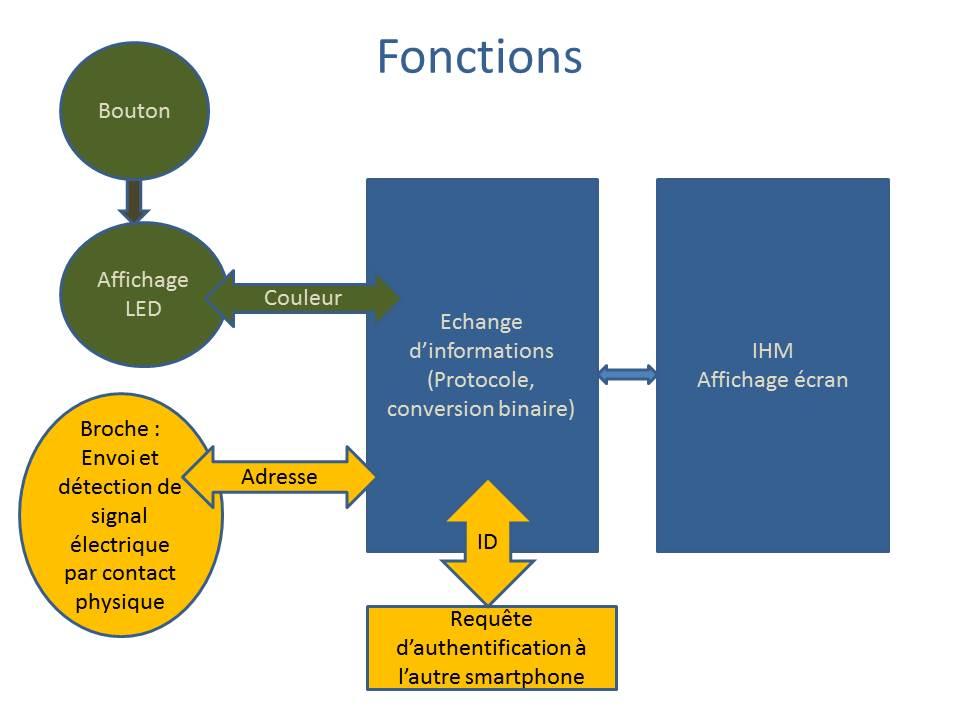
***4. Description fonctionnelle du prototype allégé:***

***A. Description littérale:***

Le bracelet permettrait via “Bump”(infrarouge) de bracelet à bracelet d’ajouter une personne en ami(qui sera stockée sur le bracelets), ainsi que de changer de couleur via boutons sur le bracelet ou virtuellement par l’interface du smartphone.

***B. Description fonctionnelle du prototype:***

* Communication bracelet – portable
* Communication portable – portable
* *BUMP* incluant:
* Communication infrarouge avec échange d’identifiants personnels (IP) entre 2 bracelets
* Définition et mise en place du protocole de transmission d’identité (adresse IP) au Smartphone depuis le bracelet
* Définition et mise en place du protocole d’authentification
* Réglage de la couleur sur le bracelet :
* Via le bouton intégré
* Via le smartphone
* Mise en place d’une IHM simplifiée pour contrôler les quelques fonctionnalités (liste de BF)
* Pas d’administrateur serveur à cette étape, ni d’envoi de fil d’actualités



***C. Liste des modules et répartition des tâches :***

**Liste des modules :**

**Module SES - Business model**

**Expert :** Rémi Maniak

**Elèves** : Marc Faddoul, Cédric Meston, Alexandre Tadros

**Description :** Dans ce module, nous étudierons l’aspect économique du projet. Nous chercherons à identifier les acteurs, les réseaux de valeurs associés et établir un buisiness plan pour une éventuelle exploitation commerciale (stratégie de vente, pricing, branding…)

**Module électronique - GPIO/Smartphone**

**Experts** : Guillaume Duc, Tarik Graba

**Elèves** : Damien Auffret, Charles Deschard, Marc Faddoul

**Description :** Ce module a pour objectif d'établir la communication Bluetooth entre le Smartphone et le bracelet et et la communication IR entre deux bracelets. La réalisation de ce module permettra de piloter le bracelet via le Smartphone (demander l’allumage des LED, vibreur…) et de transformer le bracelet en interface pour le resau social. Il est central pour le projet, sa valeur ajoutée venant dans l’interaction sociale enrichie par les fonctionnalités du bracelet. D’un point de vue pédagogique, il nous permettra d'appréhender des protocoles de communication répandus et les entrées/sorties d’un module embarqué.

**Module électronique - Bracelet**

**Experts :** Guillaume Duc, Tarik Graba

**Elèves :** Damien Auffret, Charles Deschard

**Description :** Ce module nous permettra de connecter et de gérer les différents composants autour du micro-controleur afin de pouvoir le programmer. Il s’agit de construire le BumpBand et d’en assurer le fonctionnement interne. D’un point de vue pédagogique, il permettra de nous introduire à quelques notions d’électronique et à la programmation protocolaire d’un module embarqué.

**Module Androïd - outils de communication**

**Expert :** Jean-Claude Dufourd

**Elèves :** Arturo Castellanos Salinas, Félix Richart, Julien Roméro

**Description :** Le but de ce module est de créer les modules qui vont permettre aux appareils de communiquer entre eux. Il se découpera principalement en deux partie : la communication Bluetooth (avec le bracelet) et la communication Wifi (avec le portable et l’administrateur).

**Module Androïd - services internes et interface graphique**

**Expert :** Jean-Claude Dufourd

**Elèves :** Félix Richart, Alexandre Tadros

**Description** : Dans ce module, nous nous occuperons de coder toute la partie de l’application qui n’utilise que des données locales au téléphone, ainsi que l’interface graphique de l’application.

**Module programmation distribuée**

**Expert :** Thomas Robert

**Elèves :** Arturo Castellanos Salinas, Cédric Meston, Julien Roméro

**Description :** Le but de ce module est de créer les protocoles de communication entre les différents modules. Il faudra donc sans cesse communiquer avec les différents individus de groupe afin d’unifier les moyens de communication. Il faudra donc maîtriser différentes notions aidant à décrire les procédés et réussir à les expliquer.

**Répartition des tâches :**

1. Contact ***(Module électronique-Bracelet)***
   1. Transmission physique via IR courte portée
   2. Authentification à l’aide de l’horaire et d’adresse
2. Affichage ***(Module Electronique-Bracelet)***
   1. Affichage LED
   2. Bouton de commande pour piloter à travers le micro-contrôleur
3. Communication bluetooth du bracelet ***(Module Electronique-GPIO + Module programmation distribuée)***
   1. Transmission infos bump
   2. Pilotage du bracelet
4. Envoi et réception depuis le smartphone:
   1. Protocole et socket avec établissement de la communication et interprétation des messages ***(Module programmation distribuée + Module Androïd-Communication)***
   2. Gestion interne de la communication avec stockage des listes d’amis ***(Module Androïd-Communication + Module Androïd-Services internes)*** pour :
      1. Communiquer en bluetooth avec le bracelet(réception d’une demande d’ami, envoi de couleur)
      2. Communiquer en wifi avec le smartphone(certifier l’identification)
5. Interaction avec l’écran ***(Module Android-Services internes)***
   1. IHM pour :

i. choisir une couleur

ii. voir ami ajouté et liste

* 1. avec gestion parallèle des changements d’écrans et des changements dans les autres applications(modifications de données)

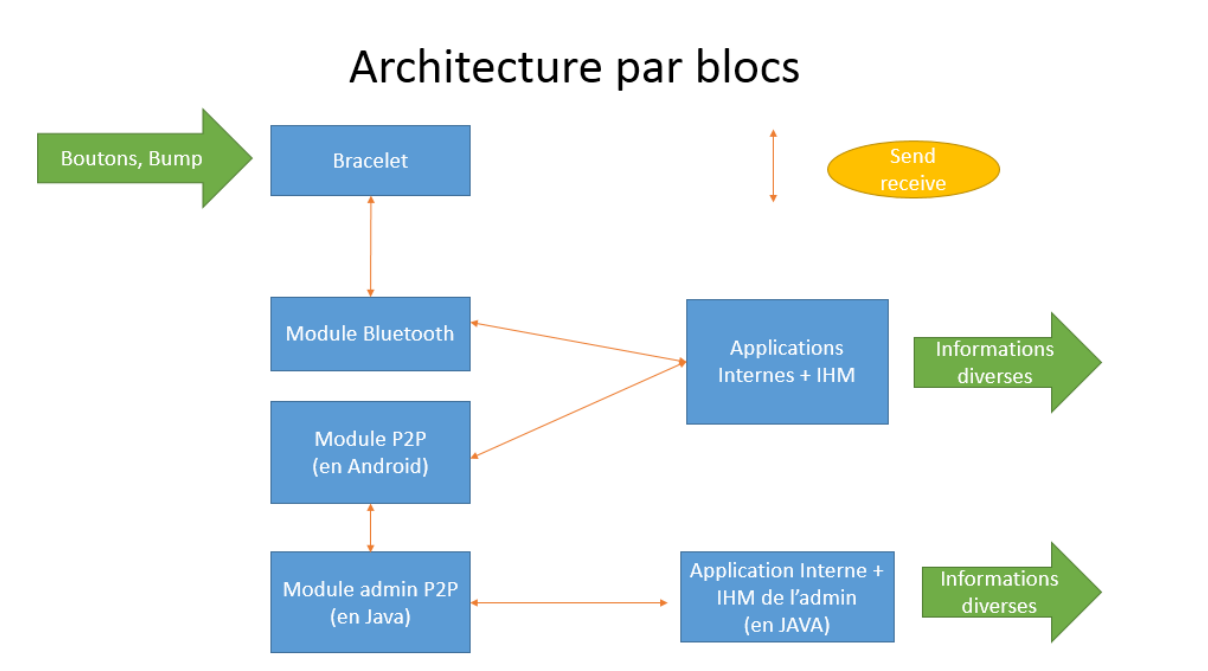
***D. Liste du matériel demandé :***

Voici la liste du matériel demandé afin réaliser trois bracelets ainsi qu’un administrateur (le bar par exemple). Il est également possible que l’on demande à emprunter des téléphones Android.



***5. Architecture informatique du prototype allégé:***

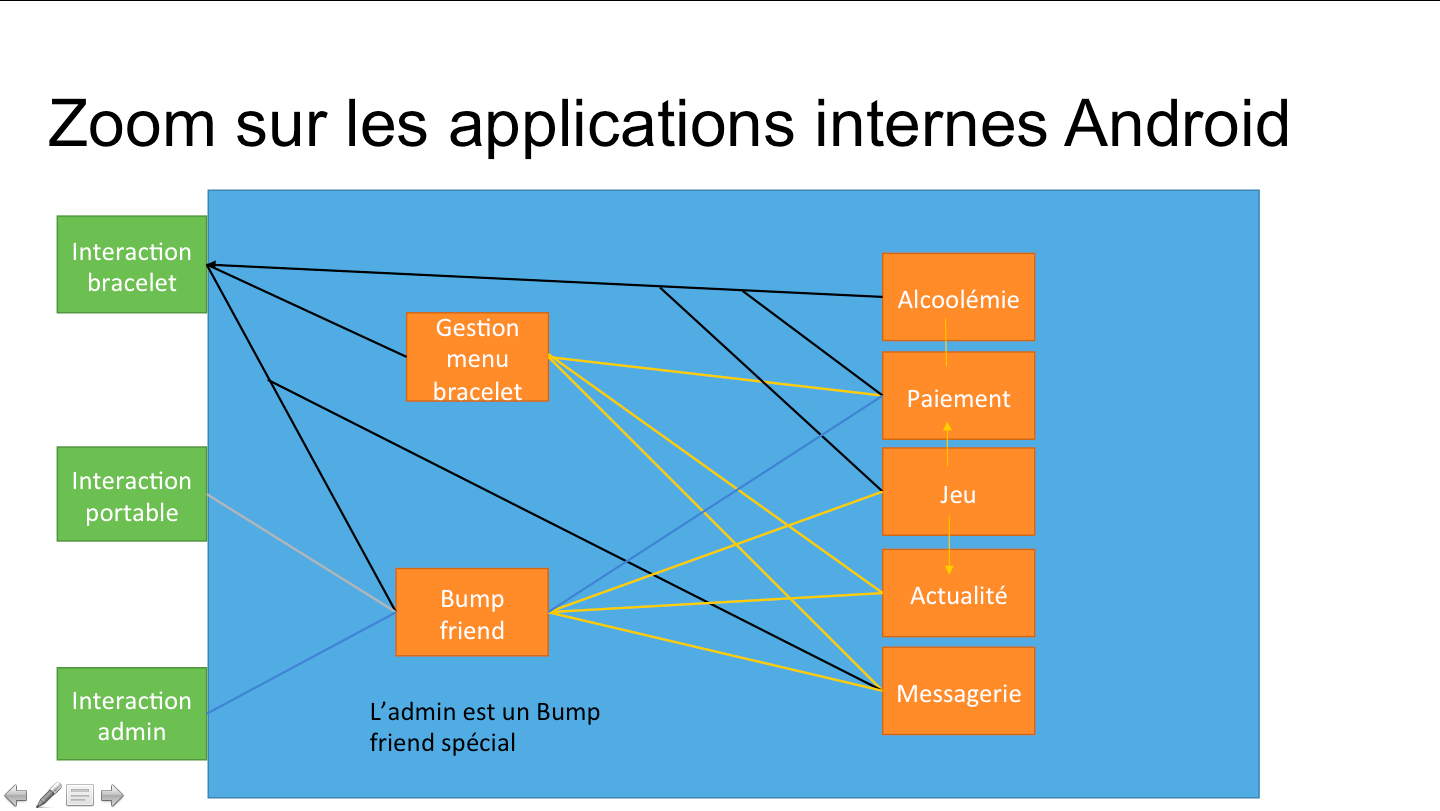
Lors de la journée de Génie Logiciel, nous avons défini les différents blocs qui définiront notre projet. La principale difficulté était le nombre d’appareils entrant en jeu. Il nous fallait donc un découpage prenant en compte cela. De plus, les entrées/sorties sont très peu nombreux mais le nombre de communications internes est très important. Il fallait donc définir les différents protocoles de communication entre les blocs. Voici le découpage auquel nous sommes arrivés :



En fait, les entrées de l’utilisateur se font principalement sur le bracelet (afin d’éviter la sortie du téléphone de manière répétée) mais l’utilisateur pourra interagir aussi avec le téléphone. L’administrateur aura lui aussi une interface graphique et une borne de synchronisation. Concernant les sorties, on a au niveau du bracelet l’indication lumineuse, l’écran et le vibreur. Ensuite, au niveau du portable et de l’administrateur, des informations diverses.

Chaque appareil va comporter une partie communication avec les autres agents. Du fait des différents dispositifs de communication (Bluetooth et Wifi), des blocs différents seront nécessaires. De plus, si l’administrateur possède un ordinateur, il lui faut son propre module de communication.

Une fois ce schéma réalisé, il nous a paru bien pour décrire de manière générale notre système mais n’indiquait guère les fonctionnalités futures. Nous avons donc décidé de faire un schéma des applications internes :



Les entrées (autre que celles de l’utilisateur sur l’écran) se font par le biais des modules de communication. Le module gestion des menus permettra principalement de gérer les interactions avec les boutons et à affecter la bonne action. Ensuite, la gestion de liens sociaux passe par un module Bump Friend qui répartira les actions. Ensuite viennent les différents modules qui traduisent les fonctionnalités.

**6. Planification temporelle:**



***7. Perspectives pour le prototype final:***

En plus des fonctionnalités du prototype allégé (affichage de couleur et ajout en bump-friend), le prototype final :

* Sera muni d’un vibreur et d’un écran (en plus des LED)
* Permettra à l’utilisateur d’envoyer des messages à ses bump-friends via un système de messagerie locale, via le smartphone.
* Inclura un administrateur, qui pourra à tout moment lancer le jeu de rencontre (Tous les bracelets se colorent. Chaque utilisateur doit trouver et Bumper les autres utilisateurs ayant la même couleur que lui pour gagner une réduction par exemple, favorisant ainsi les rencontres entre inconnus.)
* Permettra de payer ses consommations à l’administrateur, ainsi que son vestiaire.Inclura un fil d’actualité sur le smartphone, qui nous indiquera ce que nos bump-friends sont en train de faire. (Un code couleur sera instauré. Par exemple, jaune signifierait “je vais bientôt partir”. Alors, lorsqu’un utilisateur met son bracelet en jaune, ses bump-friends sont avertis sur le fil d’actualité qu’il va bientôt partir.)

Nous avons jugé ces fonctions comme constituant une bonne bibliothèque capables de démontrer l’esprit du projet. Elles pourront ensuite être enrichies par un tissu applicatif personnalisé par l’utilisateur final. Si le projet est commercialisé à grande échelle, les fonctions que nous allons implémenter ne sont qu’un exemple des fonctionnalités que le bracelet pourrait avoir.

Tout d’abord, des fonctionnalités simples peuvent être ajoutées, comme la possibilité de voter pour la musique via le bouton du bracelet, ou utiliser le micro du smartphone pour faire bouger la lumière du bracelet en rythme avec la musique.

On peut également envisager la création d’un réseau social sur le web reposant sur le bracelet. On serait alors obligé d’avoir rencontré la personne pour pouvoir être bump-friend avec lui sur le réseau social.

Et si le bracelet est utilisé en extérieur, alors on peut se servir des coordonnées GPS de l’utilisateur et faire vibrer le bracelet lorsqu’un autre utilisateur partageant des points communs avec le premier passe à proximité de lui. Cela renforcerait d’autant plus le côté social, allié à la proximité. En plus en extérieur, il n’y a pas le problème d’y avoir trop de gens aglutinés les uns aux autres, donc la détection avec la couleur sera plus facile.

**Structure de Test**

Afin de savoir si un programme fonctionne correctement, nous avons du mettre en place un séquence de tests. Celle-ci vise à parcourir toutes les fonctionnalité de l’appareil pour vérifier le bon fonctionnement après chaque mise à jour.

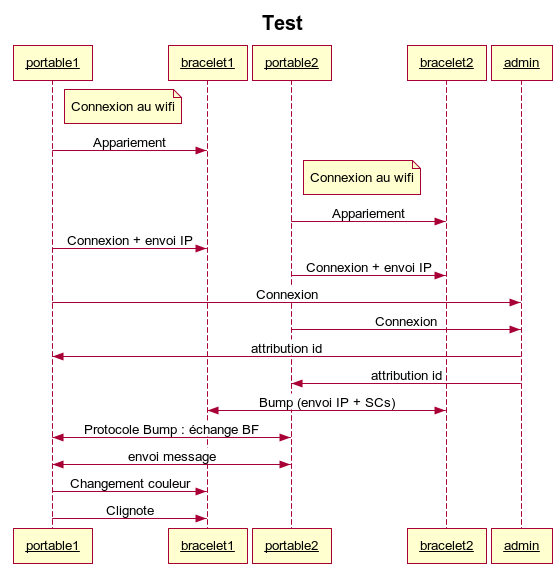
Il va comprendre cinq acteurs :

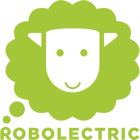
- deux portables. En effet, la communication est a coeur de notre projet. Dans un premier temps, nous utilisons deux portables pour la simplicité de l’écriture du diagramme mais il est tout à fait possible et même recommandé de faire les tests avec au moins trois portables.

- deux bracelets. Ils sont inséparable de leurs portables.

- un administrateur. C’est lui qui gère la liste des participants avec leur identification. Il est unique et sera en général contrôlable depuis le bar.

Pour illustrer la situation, nous avons choisi de nous appuyer sur un diagramme de séquence. Celui-ci s’adaptera à l'avancée des fonctionnalité dans les divers acteurs.



Pour tester les classes d’Android, nous utilisons une librairie appélée Robolectic.

Que propose robolectric. Tout d’abord, il est open source. Ensuite il permet de réaliser des tests très rapidement. En effet, le programme n’utilise pas Dalvik mais directement la JVM. Il est aussi plus simple à utiliser que Junit.

L’installation de robolectric fut des plus périeuse : il s’intègre assez mal dans Android Studio, ou du moins il n’existe pas de méthode directe pour le faire. Il nous a donc fallut manipuler gradle, ce qui est toujours compliqué et au final, il ne peut se lancer qu’à travers la console.

Grâce à cette librairie, nous pouvons tester de nombreuses manières nos classes. Cependant, il est difficile de tester les connexions réseaux en TCP. Or il s’agit là d’une grande partie de notre projet. Il faut donc une approche plus manuelle, qui suit le schéma de Test décrit plus haut.

Pour trouver nos fichiers de test, vous pouvez vous rendre sur notre dépot git, sur la branche la plus récente (https://github.com/projetpact41/bumpband).

**8.Résumé de l’état des modules**

**MODULE PROGRAMMATION CONCURENTE ET/OU DISTRIBUEE**

Définitions :

- Thread : similaire à un processus, il permet de séparer le programme en plusieurs files d ‘exécution et ainsi permet d’effectuer des actions en parallèle.

- Variable partagée : variable commune à plusieurs threads/processus. Il faut faire attention lorsqu’on les modifie/lit depuis plusieurs threads car des problèmes peuvent apparaître.

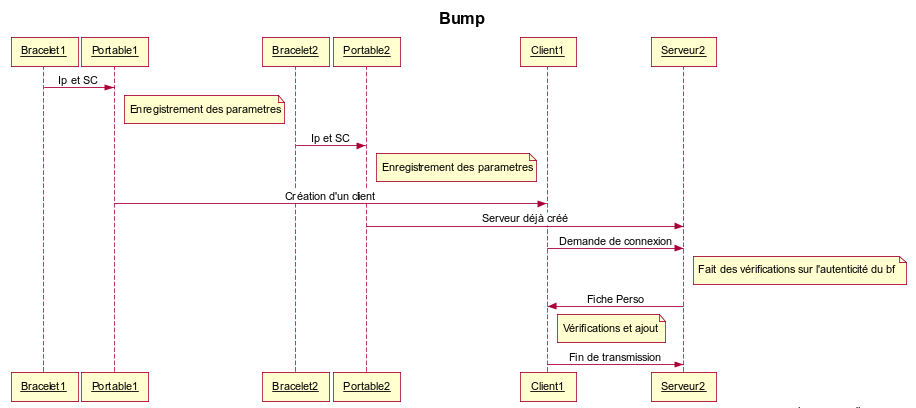
- Messages : entités transmises au cours d’une communication.

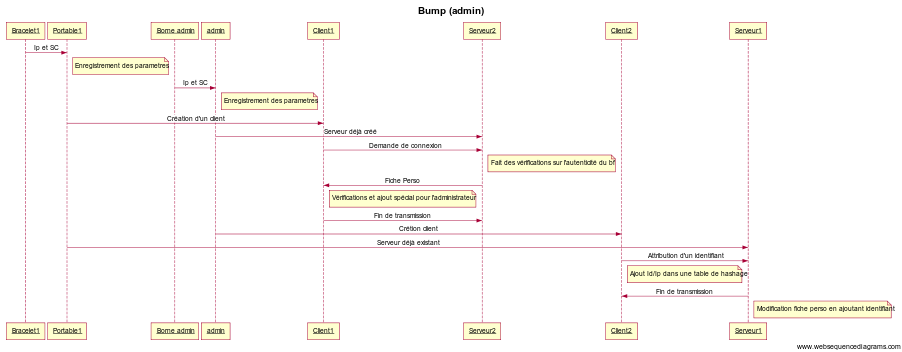
- Protocole : définit la façon de coder les informations dans les messages transmis.

- Interface de communication : technologie utilisée pour transmettre l’information.

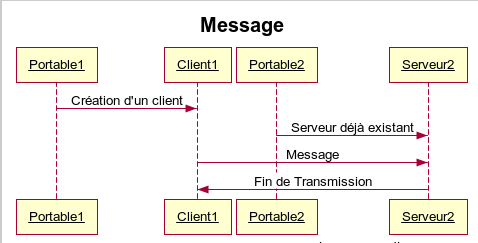
Ce module est le lien entre plusieurs modules : GPIO, programmation Android, et programmation de l’administrateur. Il définit les différents protocoles de communication pour les interfaces de communications suivantes : wifi, Bluetooth et infrarouge. Il aiguille les différents acteurs sur l’utilisation de différents threads pour permettre l’utilisation d’applications concurrentes.

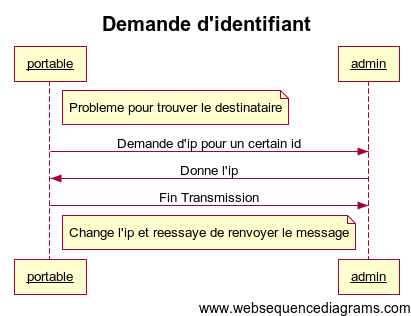
Pour chaque actions, il nous faut définir des diagrammes de séquence. Ceux-ci nous permettent de connaître l’agencement d’une communication pour arriver à un certain objectif. En voici la liste :

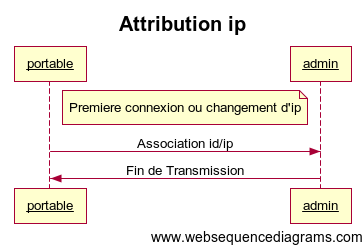




(Il nous faut considérer les cas du jeu de couleur.)







Dans notre projet, nous utilisons de nombreux threads afin de pouvoir faire plusieurs actions en parallèle. Le serveur est sur un thread pour qu’il puisse tourner en tâche de fond. A chaque fois qu’un client se connecte, un nouveau thread est créé afin de pouvoir traiter plusieurs clients simultanément. Lorsque l’on crée un nouveau client, un thread est aussi créé. Il y a aussi un thread créé pour le serveur Bluetooth.

Il y a quelques variables partagées, qui se révèlent surtout être des fichiers. Tout d’abord, la liste des bump friend est partagée, ainsi que la fiche personnelle, l’identité de l’administrateur, la liste des messages. Il faudrait penser à utiliser des sémaphores afin de gérer l’accès à ces fichiers. Ce sera utile si l’on bumpe trop rapidement ou que l’on reçoit trop de messages d’un coup.

Comme nous l’avons déjà précisé plus haut, il y a trois interfaces de communication. D’abord, le wifi. L’envoi se fait à l’aide du protocole TCP, mais cela se révèle assez transparent pour nous. Ensuite, il y a le Bluetooth, avec son protocole. Enfin, il y a aussi l’infrarouge entre les deux bracelets.

Ce qui nous intéresse plus, c’est la manière dont sont codés les messages transmis, c’est à dire le protocole de communication que nous mettons en place. Pour faire simple, nous résumons cela dans des tableaux.

Pour le Bluetooth :

La séquence de base est la suivante :

Nbr de bytes transmis(sur un byte) + message.

Voici le tableau servant à coder le message :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bracelet vers portable |  | Envoyez des bytes depuis le bracelet de la forme |  |  |  |
| Fonctions | Bump | Bouton1 | Bouton2 |  |  |
| Code | 0 | 1 | 2 |  |  |
| Envoie | 0+IPadresse+SC1+SC2 | 1 | 2 |  |  |
| Codage | IPadresse : 4 bytes, SC1 et SC2 : 1byte chacun pour l'instant |  |  |  |  |
| Portable vers bracelet |  | Envoie des bytes au bracelet |  |  |  |
|  |  |  |  |  |  |
| Fonctions | sendMessage | vibre | sendColor | clignote | sendIp |
| Renvoie la string | 0+message+durée | 1+durée | 2+numeroCanal+couleur | 3+fréquence+duree | 4+IP |
|  |  |  |  |  |  |
|  | message est une string (codé en ascii?) transformée en bytes |  | couleur sera une string de 3 bytes | fréquence aura une valeur entière (en bytes) | IP sous forme de 4 bytes. |
|  | durée est un byte (en secondes) |  | numeroCanal sera un tableau de bytes | duree en s sur un byte |  |
|  |  |  |  |  |  |

Pour le Wifi :

Le motif de base est le suivant :

Taille + numéro fonction + arguments

La taille représente le nombre de bytes envoyés. Cela facilite la réception. Le tableau suivant précise l’encodage :

|  |  |  |
| --- | --- | --- |
| BumpFriend | 0+nom+µ+ip | Nom est le nom du bf, µ sert de séparateur, ip est un chaine de caractere représentant l’ip. Tout est en ascii. |
| Color | 1+rouge+vert+bleu | Rouge, vert et bleu sont des octets, codant la couleur. |
| Connexion | 2+ip+SC1+SC2 | Ip est l’ip de celui qui fait la demande, SC1 et SC2 sont les codes de sécurités, codés sur un octet chacun |
| Message | 3+message+|+expéditeur | Message et expéditeur sont les chaines de caractères, | sert de séparateur. |
| Transmission | 4+ConnexionReussie(+erreur) | Connexion reussie = 1 si tout s’est bien passé, 0 sinon. Erreur est une chaine de caracteres codant l’erreur. |
| Indentifiant  (spécial admin) | 5+identifiant | Identifiant sur 4 bytes |
| IdentifiantAnswer | 6+ID+IP | id (int)  ip (String) |

Ce tableau sera à compléter à chaque fois qu’une nouvelle fonction est ajoutée.

Infrarouge :

C’est le groupe GPIO qui s’en occupe.

Nous avons codé les parseurs pour le Bluetooth et pour le Wifi. Ceux-ci vont être utilisés par les modules programmation android et administrateur. Il a cependant été nécessaire d’adapter selon que le programme marche sur Android ou sur l’ordinateur. En effet, certaines fonctions et classes existent sur l’une et non sur l’autre, comme par exemple les logs ou le context.

En pratique :

Pour le wifi, nous auront besoin d’utiliser les classes Sockets et SocketServer, autant pour la programmation android que pour la programmation de l’admin en java « pur ». Les communications vont se faire en P2P (Peer-to-Peer), c’est à dire que chaque téléphone est à la fois serveur et client. Nous imposons donc aux modules programmation androïd et admin de respecter cette norme.

Pour le Bluetooth, on commence par chercher dans la liste des appareils appareillés celui se nommant « Bracelet ». Ensuite, on se connecte à une socket et on récupère les flux d’entrée et de sortie. Un thread est lancé pour qu’un serveur écoute ce qu’il se passe.

Le code est dans un état plutôt avancé. Il est disponible sur le serveur git. Nous avons créé les communications wifi et les communications Bluetooth.

Au départ nous avions opté pour un envoi d’objets directement entre les plateformes android. Cependant, cela posait quelques problèmes de portabilité, et l’impossibilité de développer l’application sur IPhone ou Windows Phone. Nous avons donc du refaire le protocole et recoder ce qui l’avait déjà été.

Les objets pouvant être transmis implémentent l’interface Transmissible. Celle-ci demande à la classe de réaliser deux fonctions : execute (qui est une fonction onReceive) et toBytes qui permet de transformer l’objet en bytes pour être transmis. Le protocole est donc en partie codé dans chaque objets.

Nous venons d’intégrer en partie le bracelet dans le portable. Nous avons réussi à transmettre de l’information et à générer le début des protocoles. Cependant, nous nous sommes heurté à un obstacle : certains messages Bluetooth était scindés en plusieurs parties, rendant alors la lecture impossible. Il a donc fallut modifier le protocole bluetooth : maintenant, les séquences commencent par la taille du message transmis (ce qui n’était pas le cas avant).

Cette phase d’intégration s’est révélée plus longue que prévue et nous as permis d’exhiber des erreurs que nous n’avions pas décelé plus tôt.

Dans ce module, nous sommes en avance par rapport aux attendus du pan3 : le code est proche d’être complet même certains ajustement sont nécessaire. Nous devrons sans doute chercher à optimiser d’avantage le programme durant le pan4. Le protocole est déjà intégré dans le téléphone et dans le bracelet. Il nous restera à intégrer l’administrateur au protocole.

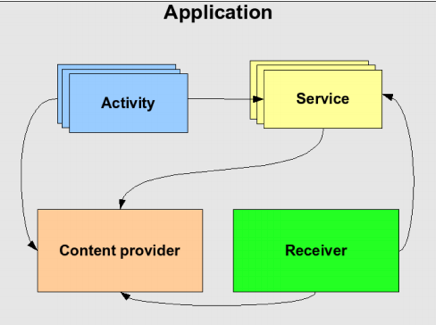
Pour ce qui concerne les tests, nous suivrons le processus décrit dans le cahier des charges. Il est difficile d’automatiser ces tests car ils font intervenir beaucoup de connexions à des réseaux très différents.

**Module Android : Outils de communication**

Dans un premier temps, nous avons mis en place les outils nécessaire au développement. Nous avons choisi d’utiliser Android Studio et d’émuler sur GenyMotion. Nous avons rencontré un certain nombre de problème, notamment pour porter un projet d’un ordinateur à l’autre.

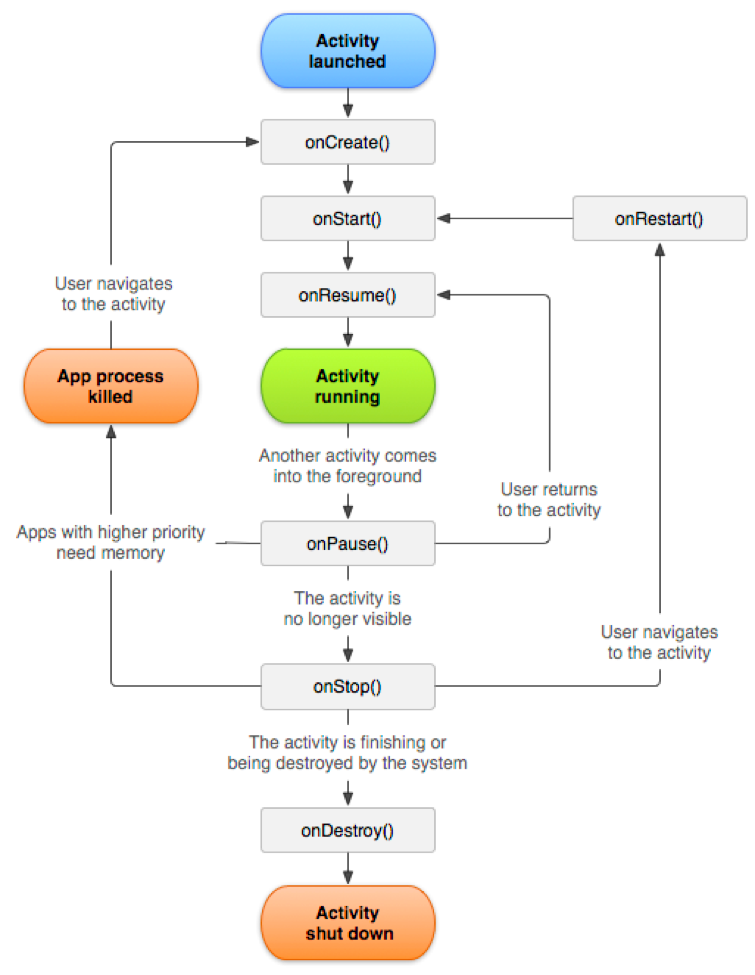
Nous nous sommes ensuite intéressé au cycle de vie d’une application, d’une activité et des Services.

Une application est composé de quatre composants : les « Activity », les « Service », les « BroadcastReceiver » et les « ContentProvider ».



(<http://www.iro.umontreal.ca/~dift1155/cours/ift1155/communs/Cours/1P/02CycledeVie.pdf>)

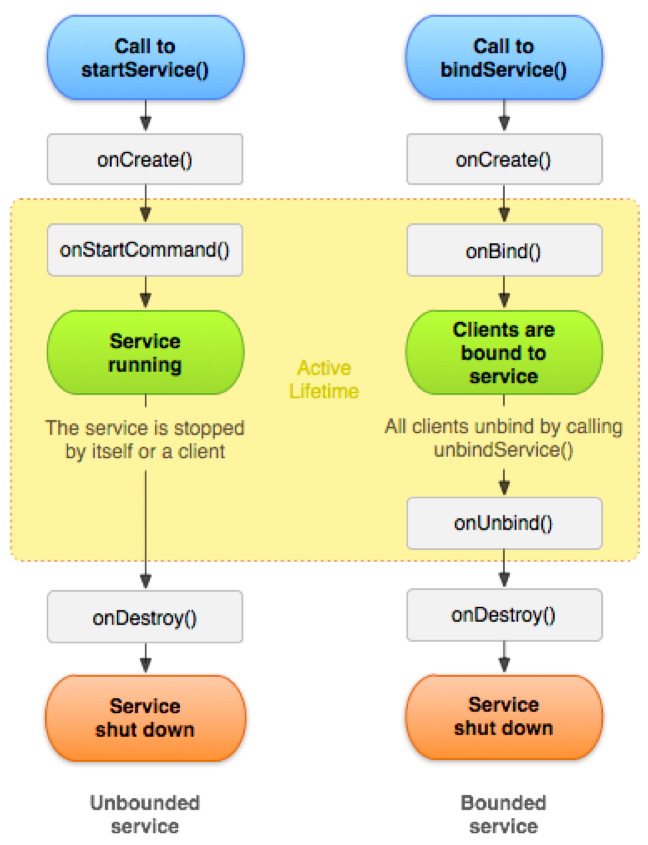
Une Activité représente l’interaction entre l’utilisateur et le téléphone. Il s’agit de ce qui s’affiche à l’écran. Il n’y a qu’un activité par page et chacune représente une page différente. Voici le cycle d’une activité :



(<http://developer.android.com/reference/android/app/Activity.html>)

Nous avons pu implémenter certaines de ces fonctions (qui héritent de la classe Activity).

Un service est une action qui est effectuée en arrière plan et qui n’est pas visible. Il prodigue un certain « service », c’est à dire qu’il effectue une action donnée sur demande. Voici son cycle de vie :



(<http://developer.android.com/guide/components/services.html>)

Un BroadcastReceiver est un récepteur d’intention. Lorsqu’il se produit un événement sur le portable, comme la réception d’un SMS ou une alerte de batterie faible, Android émet une intention pour informer le reste du système. Le BroadcastReceiver l’intercepte et agit en conséquence.

Les ContentProvider permettent de gérer des bases de données.

Nous avons ensuite créé l’application. Nous avons crée une activité principale qui nous demande un identifiant et une autre page afin de vérifier si nos communications se passent bien. Nous nous sommes ensuite attelé à programmer la communication entre deux téléphones. Dans un premier temps il s’agissait simplement d’un serveur écho.

Pour stocker la liste des BumpFriend, nous avons du nous intéresser au stockage des données persistantes. Nous avons essayé les Préférences, mais ils se sont révélés trop lourd pour ce genre de données. Nous avons ensuite utilisé une simple classe stockant un tableau de bump friend. Cependant, si l’application venait à se fermer, les données étaient réinitialisées. Nous avons finalement opté pour les fichiers textes. Nous aurions pu utiliser des bases de données, mais celles ci ont été jugées inutiles par un expert en début de projet et elles impliquaient la maitrise de la SQL.

Nos avons ensuite programmé la communication Bluetooth. Le module Protocole de communication nous a fourni les parseurs pour le protocole à utiliser. Nous les avons intégrer. De plus, ils nous ont aussi fournis des diagrammes de séquence qui nous ont permis de savoir comment se passe chaque action.

Dans un premier temps, nous envoyions directement des objets par WiFi. Cependant, cela posait quelques problèmes, notamment pour la portabilité. Nous avons donc suivi un nouveau protocole, en envoyant des octets. De l’ancienne programmation subsiste des objets implémentant l’interface Transmissible, possédant une fonction execute() (fonction du type onReceive) qui effectue la bonne action. Il y a aussi une fonction toBytes() traduisant l’objet pour le transmettre. A chaque extrémité d’une communication, l’objet est transformé en bytes et ensuite retransformé en objet. Ainsi, nous avons pu conserver une partie de notre ancien travail tout en réduisant la charge transportée et en augmentant la portabilité.

Une grosse étape a été celle de l’intégration du bracelet. Nous nous sommes retrouvé avec le module GPIO et nous avons mis au point la communication bracelet. Des erreurs sont apparus mais grâce aux messages du Log, elles ont pu être identifiées. Cela a même permis de débogger le module arduino. Un problème est aussi apparu avec l’utilisation des bytes. Sous Java, ces derniers sont signés donc vont de -128 à 127. Il nous a fallut manipuler des bits avec des >> et des & pour pouvoir les retransformer correctement en int.

Concernant l’avancement du module, les attentes du pan3 sont validées : les parties Bluetooth et Wifi ont été implémentée et intégré aux portables et aux bracelets. Pour le pan4, il nous restera à intégrer le module administrateur et à optimiser le code. Il pourra aussi apparaître de nouveaux objets Transmissible qui devront être intégré.

**Module Administrateur**

En accord avec le module protocole de communication, les protocoles ont été adaptés pour donner une priorité à l’administrateur. Au début de l’initialisation du programme sous Android, l’utilisateur doit se synchroniser avec l’administrateur. Ceci fait, il garde l’adresse ip de l’admin, qui sera supposée fixe, en mémoire. Quand il reçoit un message prioritaire, il vérifiera l’identité de l’administrateur.

Nous avons réalisé les premiers essais de communication WiFi entre deux ordinateur. Cela nous a permis de comprendre le fonctionnement des Sockets. Ensuite, le programme a été adapté pour Android et des essais ont été réalisés entre un ordinateur et un smartphone. Après cela, l’application Android a subit de gros changements. Initialement, les communications se faisaient par envoi d’objets, mais cela impliqué trop de contraintes. Le programme a été revu dans son intégralité pour conservé le travail fait tout en changeant le mode de communication. Suite à cette phase de changement, un programme a été retransféré sur l’administrateur. Les protocoles ont du être adaptés et certains même créés.

Nous avons ensuite géré les identifiants. Les ip n’étant pas forcément fixes,il fallait trouver un moyen de quand même pouvoir retrouvé un destinataire en cas de changement d’ip. Après la synchronisation avec l’administrateur, le smartphone obtient un identifiant unique. L’administrateur contient une table de correspondance entre l’id et l’ip. En cas de problème, un smartphone peut redemander l’id à l’administrateur.

**Avancées diverses**

- Implémentation d’un récepteur de SMS : quand l’utilisateur reçoit un SMS, le bracelet s’allume dans une certaine couleur et clignote pendant 5 secondes. Ainsi, malgré le bruit ambiant, un stimulis visuel attire son attention.

-Implémentation d’appel : quand le téléphone reçoit un appel, de la même manière que pour les SMS, le bracelet s’allume d’une certaine manière et clignote.

***9. Bibliographie:***

**Sites Web :**

- <http://fr.openclassrooms.com/>

- <http://developer.android.com/>

- <http://docs.oracle.com/javase/tutorial/networking/>

- <http://www.cc.gatech.edu/fac/Russell.Clark/Classes/6250/PracticalSocketJava.pdf>

- <http://arduino.cc/en/Main/ArduinoBoardMini>

- <https://www.sparkfun.com/products/12580>

- <http://arduino.cc/en/Main/MiniUSB>

- <http://www.evola.fr/product_info.php/lcd-8x2-retro-eclairage-bleu-p-83>

- <http://www.evola.fr/product_info.php/mini-moteur-vibration-p-90>

**-** <http://www.digikey.com/catalog/en/partgroup/plcc-4-tri-color-rgb-round-with-flat-top-leds/37338>

- <http://www.evola.fr/product_info.php/mini-interrupteur-bouton-poussoir-p-235>

- <https://learn.sparkfun.com/tutorials/ir-communication/getting-started>

- <https://learn.sparkfun.com/tutorials/using-the-bluesmirf>

**Livre :**

- Programmation Android, W. frank Ableson, Robi Sen, Chris King, C. Enrique Or[tiz](http://developer.android.com/)

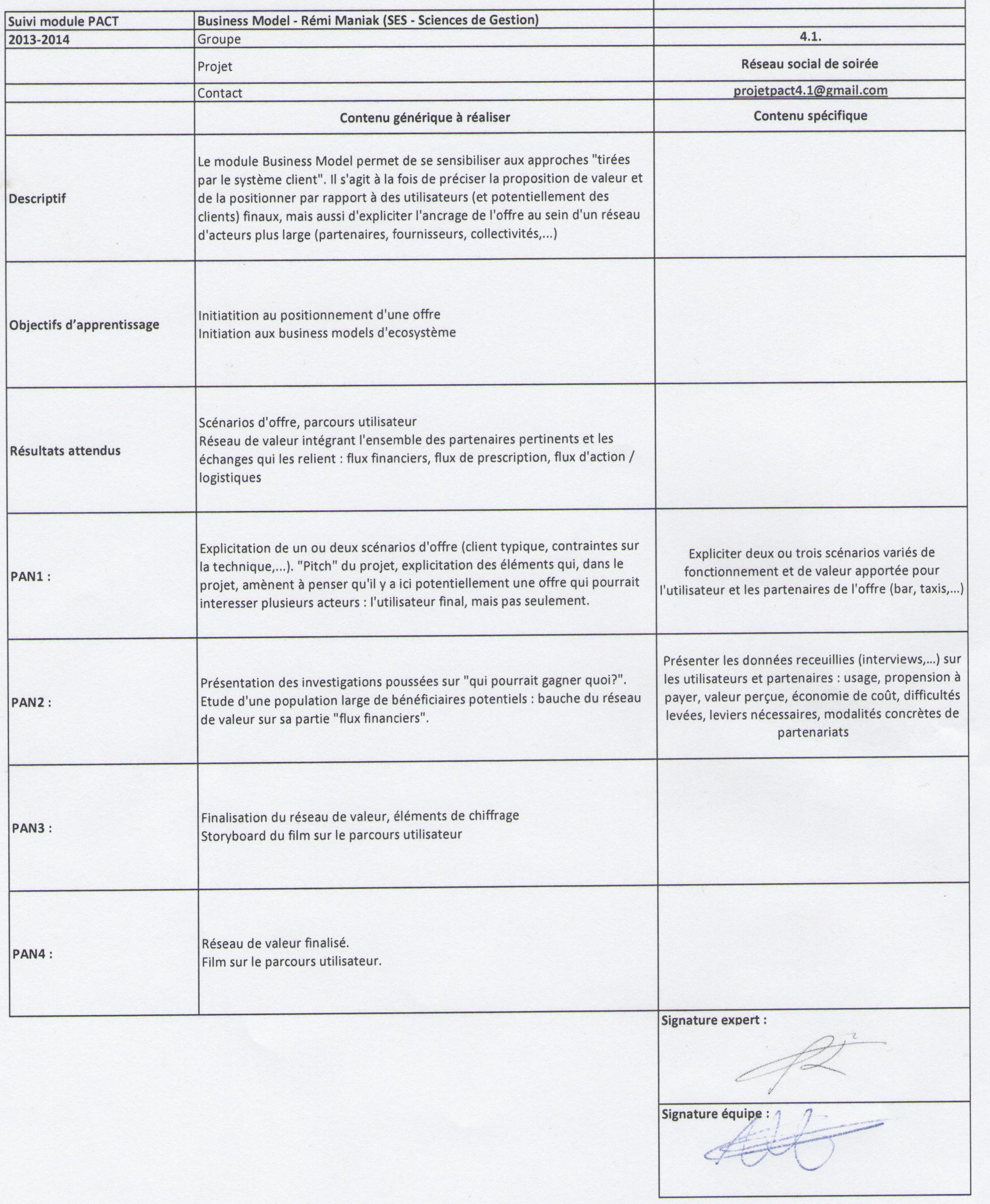
***Annexe 1 : fiche d’identité du groupe:***

Le groupe PACT4.1 s’est constitué, comme tous les autres groupes, par le hasard des algorithmes de Télécom Paristech. Mais le hasard a bien fait les choses. Les membres se complètent les uns les autres pour former une équipe solidaire, et solide dans tous les domaines. En effet, chacun a ses propres centres d’intérêts scolaires dans lesquels il est investi et pour lesquels il réussit à partager ses connaissances et sa passion au reste du groupe. En effet, ce dernier est composé de Damien Auffret (intérêt : électronique), Cédric Meston (aspect économique et social), Charles Deschard (communications sans fil, économie), Julien Romero (programmation), Arturo Castellanos (programmation), Félix Richart (informatique), Marc Faddoul (leadership et organisation), Alexandre Tadros (aspect économique et design). Cet éventail de préférences disciplinaires, allié à une idée de projet que chacun porte et en laquelle chaque membre se reconnaît, permet un travail en groupe efficace et une répartition des tâches facile à effectuer ( bien que chaque membre soit déterminé à profiter de PACT pour s'intéresser aussi aux domaines qu’il ne maîtrise pas et a donc de lui-même l’envie de participer à la réalisation d’autres modules que celui de son domaine de prédilection ).

Nos méthodes de travail en groupe se définissent en deux catégories : présence physique et travail à distance. Nous organisons au moins un rendez-vous hebdomadaire en plus des TH consacrées à PACT pour une mise au point collective des idées de chacun et des taĉhes à effectuer, et la répartition de celles-ci. Pour ce qui est du travail à distance, nous avons créé un groupe Facebook et un google drive, ce qui nous permet, d’une part de garder un historique de tout ce qui a été fait, de partager et modifier facilement (par chacun) les travaux de chacun, d’autre part de permettre au tuteur de se tenir précisément au courant de l’avancée du projet.

**Annexe 2 : fiches modules:**

**Module SES**



**Module – Arduino + Bluetooth**

**Projet PACT 4.1**

**Experts :**

- Guillaume Duc - Tarik Graba

**Périmètre du module :**

Le module est centré sur un micro-contrôleur. Ce micro-contrôleur fait tourner de façon autonome un programme qui pilote une ou plusieurs LED RGB. Le programme sur le micro-contrôleur peut également dialoguer, grâce à un module Bluetooth, avec une application sur un appareil Android. Enfin, deux dispositifs doivent pouvoir échanger des informations entre-eux via une liaison filaire.

**Matériel nécessaire :**

- Carte Arduino Pro Mini x 2 - Module Bluetooth Mate Gold (à commander) x 2

**Objectifs pédagogiques :**

- Savoir ce qu'est un micro-contrôleur - Découvrir l'utilisation d'un module Bluetooth - Programmer un micro-contrôleur et utiliser ses entrées/sorties

**Déroulement :**

- PAN 1 (décembre) : Savoir ce qu'est un micro-contrôleur, quelles sont ses possibilités et ses limites

- PAN 2 (janvier) : Choix du module Bluetooth, Étude de sa documentation, Première mise en œuvre de la carte Arduino (mise en place de l'environnement de développement, communication avec la carte, utilisation des entrées/sorties)

- PAN3 (mars) : Module opérationnel

**Élèves concernés : Charles Deschard, Damien Auffret, Marc Faddoul**

**Module Android - Interface graphique et services internes**

Proposé par: Jean-Claude Dufourd (jean-claude.dufourd\_at\_telecom-paristech.fr, 7733, bureau DA413)

Réalisé par : groupe 4.1

**Elèves :** Félix Richart, Alexandre Tadros

**Descriptif**

L’utilisation de smartphones ou de tablettes est devenue depuis quelques années presque

incontournable. Parmi les différents systèmes d’exploitation permettant de contrôler ces

smartphones et tablettes, le système Android représente près de la moitié des périphériques. La programmation des smartphones ou tablettes Android fait appel au langage Java, vu lors du cours INF103. Elle ne devrait pas poser de difficulté majeure, néanmoins une des difficultés souvent rencontrées est la prise en main de l’environnement de développement et des spécificités des applications Android. Ce module a pour vocation d’aider les élèves à surmonter cette difficulté.

**Ressources dont le module dépend**

Le site de référence pour l’environnement de développement Android (bibliothèques,

simulateur, environnement de développement, etc.) et pour l’auto-formation est le

site <http://developer.android.com> , Transparents du Mini-cours Android.

**Exemples d’utilisation du module:**

- Application Android avec une interface graphique simple

- Application Android utilisant la caméra d’un téléphone mobile

- Application Android utilisant l’écran tactile d’une tablette

**Objectifs d’apprentissage**

*- Informatique :*

prendre en main l’environnement de développement Android, comprendre

le fonctionnement d’une application Android, savoir lire et utiliser la documentation des

API Android, savoir créer une application Android de base et utiliser certaines

fonctionnalités avancées (affichage graphique, réseau, multimédia …), savoir debugger une

application.

**Résultats attendus:**

**PAN1 :**

- Démontrer que l’environnement de développement est en place et que des

applications simples fonctionnent sur un émulateur.

- Expliquer le cycle de vie d’une application et la notion d’activité

**PAN2 :**

- Réaliser une application Android sur machine virtuelle mettant en œuvre le cycle

de vie, l’enchaînement de plusieurs activités, et le lancement d’une autre

application.

- Expliquer le fonctionnement du stockage de données persistantes.

- Réaliser une application spécifique. Cette application fonctionnera sur un

smartphone ou une tablette et pourra mettre en œuvre.

**PAN3 :**

- Bluetooth/WiFi

***PAN4 :***

- Réaliser l’application finale

- Analyser comment le module est intégré dans le prototype, quelles pistes

d’améliorations seraient à envisager (performance, simplicité)

**Module Android - Outils de communication:**

Proposé par: Jean-Claude Dufourd (jean-claude.dufourd\_at\_telecom-paristech.fr, 7733, bureau DA413)

Réalisé par: groupe 4.1

**Elèves** : Arturo Castellanos Salinas, Félix Richart, Julien Roméro

**Descriptif:**

L’utilisation de smartphones ou de tablettes est devenue depuis quelques années presque

incontournable. Parmi les différents systèmes d’exploitation permettant de contrôler ces

smartphones et tablettes, le système Android représente près de la moitié des périphériques. La programmation des smartphones ou tablettes Android fait appel au langage Java, vu lors du cours INF103. Elle ne devrait pas poser de difficulté majeure, néanmoins une des difficultés souvent rencontrées est la prise en main de l’environnement de développement et des spécificités des applications Android. Ce module a pour vocation d’aider les élèves à surmonter cette difficulté.

**Ressources dont le module dépend**

Le site de référence pour l’environnement de développement Android (bibliothèques, simulateur, environnement de développement, etc.) et pour l’auto-formation est le

site <http://developer.android.com> , Transparents du Mini-cours Android.

**Exemples d’utilisation du module**

- Application Android avec une interface graphique simple

- Application Android utilisant la caméra d’un téléphone mobile

- Application Android utilisant l’écran tactile d’une tablette

**Objectifs d’apprentissage**

*- Informatique :*

Prendre en main l’environnement de développement Android, comprendre

le fonctionnement d’une application Android, savoir lire et utiliser la documentation des

API Android, savoir créer une application Android de base et utiliser certaines

fonctionnalités avancées (affichage graphique, réseau, multimédia …), savoir debugger une

application

**Résultats attendus**

PAN1 :

- Démontrer que l’environnement de développement est en place et que des

applications simples fonctionnent sur un émulateur

- Expliquer le cycle de vie d’une application et la notion d’activité

PAN2 :

- Réaliser une application Android sur machine virtuelle mettant en œuvre le cycle

de vie, l’enchaînement de plusieurs activités, et le lancement d’une autre

application.

- Expliquer le fonctionnement du stockage de données persistantes.

- Réaliser une application spécifique. Cette application fonctionnera sur un

smartphone ou une tablette et pourra mettre en œuvre.

PAN3 :

Bluetooth/WiFi

*PAN4 :*

- Réaliser l’application finale

- Analyser comment le module est intégré dans le prototype, quelles pistes

d’améliorations seraient à envisager (performance, simplicité)

**Module Programmation Concurrente et/ ou distribuée**

**Expert :**Thomas Robert

**Périmètre du module :**

Le module est centré sur l’élaboration de protocoles de communication, afin de permettre aux différents éléments de notre projet de se comprendre. De plus, certaines communications devront être sécurisées. Les différentes interactions sont la communication bracelet-portable, portable-portable et portable-administrateur. Nous devrons alors imposer aux programmeurs la manière dont les informations doivent être envoyées.

**Objectifs pedagogiques :**

- Assimilation du vocabulaire associé.

- Être capable d’utiliser les bibliothèques Java/Android pour implémenter les threads et les communications.

- Création d’un protocole de communication, avec les contraintes associées.

**Déroulement :**

**-** PAN 1 (décembre) : Connaitre le protocole associé, et l’utiliser pour expliquer l’intérêt du module dans notre projet.

- PAN 2 (janvier) : identification des interfaces utilisées et des bibliothèques offrant le motif socket + socket serveur, code des premières connexions, définition du protocole de communication entre les différents agents.

- PAN 3 (mars) : code Java/Android réalisant le protocole.

- PAN 4 : Intégration du protocole au projet.

**Elèves concernés**

**-** RICHART Félix

- ROMERO Julien

-CASTELLANOS Arturo

**Module administrateur**

IHM basique qui gère vestiaire, commandes au bar (paiement), jeu (événement lancé par l'admin propagé au portable)

admin superuser, doit avoir le privilège de changer les couleurs au dessus des utilisateurs

dans les jeux, il doit vérifier que les participants ont complété le jeu ou non (checké tous les autres)

pas dans le prototype allégé

pan 2: messages de l'admin et vers l'admin pris en compte dans le protocole

pan 3: simuler un demande de changement de couleur de l'admin qui a priorité sur les changements manuels de la personne sur le bracelet

pan 4: interface simple, jeu

module sécurité

(optionnel)

S�curiser les paiements par cryptage de messages

Pour cela, contacter expert s�curit� sur technologies simples, en mode consultation

Ensuite choix motiv� de la techno et impl�mentation

**Annexe 3 : Liste des modifications apportées au document (après PAN1):**

Tableau des modifications de fond apportées au projet avec validation des experts et encadrant informatique

|  |  |  |
| --- | --- | --- |
| Libellé / date | Description brève | Validé par : |
| Business Model enrichi (01/14) | Développement des offres  Réseau de valeurs  Sondage  Pistes de partenariats  Parcours d’usage | R. MANIAK |
| Changement Bump en IR  (01/14) | Le bump se fera desormais par IR courte portée | T. GRABA |
| Changement Architecture Informatique  (01/14) | Les schémas ont été mis à jour pour plus de clarté. Administrateur mis sur un ordinateur | M. CHOLLET |
| Modification diagramme de Gant  (12/13) | Le diagramme de Gant a été très légèrement revu. Les phases d’intégration pré PAN 3&4 sont nettement allongées. Définition de responsables sur les tâches (surtout hors module) | Jury PAN 1 |