الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
المركز الجامعي بلحاج بوشعيب لعين تموشنت
Centre Universitaire BLHADJ Bouchaib d'Ain Témouchent

Institut : Sciences et Technologies
Département : Génie Electrique

PROJET DE FIN D'ETUDES

Pour l'obtention du diplôme de **MASTER** en :

Domaine : Sciences et Technologies
Filière : Electronique
Spécialité : Génie des Télécommunications

Thème

Réalisation d'une Application Mobile de Mesure de Fréquence Cardiaque en temps Réel sous Android

Présenté Par :
M^lle Lakhdari Kheira

Devant le jury composé de :

Président	:	M^r Benmoussat Chems Eddine,	MAB, CUBBAT
Encadreur	:	M^me Ferouani Souhila,	MCB, CUBBAT
Examinateur	:	M^lle Badir Houaria,	MAB, CUBBAT

Promotion : Juin 2016

Remerciement

Nous remercions ALLAH le Tout-puissant de nous avoir donné le courage, la volonté et la patience de mener à terme le présent travail.

Ce travail s'inscrit dans le cadre de projet de fin d'études, Spécialité Génie des télécommunications au Centre Universitaire Belhadj Bouchaib Ain Témouchent.

Je remercie tout d'abord mes chers parents, ma famille; qu'ils sont toujours près de moi pour me soutenir et m'encourager.

Je remercie M^{me}FEROUANI, MCB au Centre Universitaire Belhadj Bouchaib Ain Temouchent pour la qualité de son accueil. Je lui exprime particulièrement toutes mes reconnaissances pour m'avoir fait bénéficier de ses compétences scientifiques, ses qualités humaines, sa clairvoyance, son dynamisme et sa constante disponibilité.

Je tiens à exprimer mes plus vifs remerciements à M^{lle} Badir Houaria, MAB au Centre Universitaire Belhadj Bouchaib Ain Temouchent, qui m'a fait l'honneur d'examiner mon Travail.

Je remercie sincèrement M^{r}. BENMOUSSAT, MAB au Centre Universitaire Belhadj Bouchaib Ain Temouchent d'avoir accepté de présider le jury de mon projet de fin d'étude.

J'exprime également, mes remerciements au Médecin Cardiologue M^{r} BENOSMANE et M^{r} MERZOUGUI pour la transmission d'informations essentielles à la compréhension du contexte de recherche étroitement lié à leurs spécialités.

Je tiens à remercier également l'équipe pédagogique du Centre Universitaire Belhadj Bouchaib Ain Témouchent. La richesse et le contenu de l'information que j'ai eu au sein de leur établissement m'ont donné des outils puissants pour mener à bien ce travail.

Finalement, j'exprime mes remerciements à tous ceux qui m'ont aidé à faire de ce mémoire une bonne opportunité sur le plan humain et professionnel.

Résumé

L'objectif principal de ce projet de fin d'études est de réaliser une application mobile ambulatoire, de télésurveillance ayant pour but la transmission des données médicales d'un patient souffrant d'une insuffisance cardiaque chronique.

L'application « *BasmaApp* » améliore la relation médecin-patient; on transmettant les mesures de la fréquence cardiaque de ce dernier, prises par un « *Holter ECG* » le long d'une période de 24h jusqu'à 48h; directement et en temps réel. Ce travail est destiné aux Smartphones développés sous la plateforme Android. Il est basé sur l'utilisation de l'IDE simplifié de Google ; nommé « *MIT App Inventor2* ».

Ce mémoire représente l'initiative de l'intégration des Technologies de l'Information et de Communication (TIC) dans le domaine de la cardiologie médicale.

Mots clés : télésurveillance, données médicales, application mobile, fréquence cardiaque, Holter ECG, Smartphones, Plateforme Android, IDE, MIT App Inventor2, TIC.

Abstract

The main objective of this project of end of studies is to achieve an ambulatory mobile application of monitoring aimed at the transmission of medical data of a patient with chronic heart failure.

The "BasmaApp" application improves the doctor-patient relationship; on transmitting the heart rate of the latter measures, taken by a "Holter ECG" along a period of 24 to 48 h. directly and in real time. This work is intended for smartphones developed under the Android platform. It is based on the use of the simplified Google IDE; named 'MIT App Inventor2'.

This memory is the initiative of the integration of Information and Communication Technologies in the field of medical cardiology.

Key words: monitoring, medical data, mobile application, heart rate, Holter ECG, Smartphones, Android platform, IDE, MIT App Inventor2, ICT.

الملخص

الهدف الرئيسي لهذا المشروع المندرج ضمن مذكرة نهاية الدراسة الجامعية هو انجاز تطبيق للهاتف النقال يهدف لرصد و نقل البيانات الطبية للمريض الذي يعاني من قصور مزمن في القلب.

تطبيق "بسمة اب" يحسن علاقة الطبيب بالمريض؛ وذلك بنقل معدل ضربات القلب لهذا الأخير، التي رصدها جهاز "هولتر لتخطيط القلب" على مدى 24 إلى 48 ساعة مباشرة وفي نفس الوقت. إن هذا العمل معد خصيصا للهواتف الذكية المطورة تحت منصة الاندرويد. ويستند اساسا على استخدام التطبيق المبسط المسمى" ام أي تي اب انفتر لمعهد ماساتشوستس للتكنولوجيات".

تعد هذه المذكرة مبادرة إدماج تكنولوجيات المعلومات والاتصالات في مجال أمراض القلب الطبية.

الكلمات المفتاحية :

رصد، بيانات طبية، تطبيق، معدل ضربات القلب، جهاز تخطيط القلب هولتر ، الهواتف الذكية، منصة الاندرويد، التطبيق في "معهد ماساتشوستس للتكنولوجيات"، تكنولوجيات المعلومات والاتصالات.

Table des matières

Chapitre 2 :

Développement d'applications sous Android.

Chapitre 3 :

Réalisation et mise en application de la solution proposée.

Table des figures

Acronymes et abréviations

A

ADB: *Android Debug Bridge.*
AFT: *Arrhythmia Full Tracker.*
AJAX: Asynchronous JavaScript and XML.
API: *Application Programming Interface.*
ARM: *Advanced RISC Machines.*
AVD: *Android Virtual Device.*

B

BAV: *Bloc Auriculo-Ventriculaire*
BPM: *Battement Par Minute*.

C

CDMA: *Code Division Multiple Access.*

D

2D: *2 Dimensions.*
3D: *3 Dimensions.*

E

ECG: *ElectroCardioGramme.*
EDGE: *Enhanced Data for GSM Evolution.*
EEG: *ElectroEncéphaloGramme.*

G

2G: *Seconde Génération de téléphonie mobile.*
3G: *Troisième Génération de téléphonie mobile.*
GPRS: *General Packet Radio Service.*
GPS: *Global Positioning System.*
GSM: *Global System for Mobile communication.*
GTK: *Gimp Tool Kit.*

H

HSDPA: *High Speed Downlink Packet Access.*
HTC: *High Tech Computer Corporation.*
HVGA: *Half Video Graphics Array*.

I

IBM: *International Business Machines.*
IDE: *Integrated Development Environment.*
IEEE: *Institute of Electrical and Electronic Engineers.*
iOS: *iPhone Operating System.*
IP: *Internet Protocol.*

IrDa: *Infrared Data Association.*

J

JSE: *Java Standard Edition.*

L

LED: *Light-Emitting Diode.*

M

MAC: *Media Access Control*.
MID: *Mobile Internet Device.*
MIT: *Massachusetts Institute of Technology.*
MMS: *Multimedia Messaging System.*
MS-CHAP v2: *Microsoft Challenge Handshake Authentication Protocol Version 2*.

N

NTIC: *Nouvelles Technologies de l'Information et de la Communication.*

O

OHA: *Open Handset Aliance.*
OpenGL: *Open Graphics Library.*
OS: *Operating System.*

P

PC: *Personal Computer.*
PCS: *Personal Communications Service.*
PDA: *Personal Data Assistant.*
PDC: *Personal Digital Cellular.*
PDF: *Personal Data Form.*
PIN: *Personal Identification Number.*

Q

QR: *Quick Response.*

R

RAM: *Random Acces Memory.*
RIL: *Recoverable Items List.*
RIM: *Research In Motion.*
RTC: *Réseau Téléphonique Commuté.*

S

SD: Secure Digital.
SDK: *Software Development Kit.*
SE : *Système d'Exploitation.*
SGL: *Script Graphics Library.*
SIG: *Special Interest Group.*

SMS: *Short Message Service.*
SQLite: *Structured Query Lite.*
SSL: *Secure Sockets Layer*.

$\mathcal{T}$

TCP: *Transport Control Protocol.*

$\mathcal{U}$

UDP: *User Datagram Protocol.*
UMTS: *Universal Mobile Telecom System.*
USB: *Universal Serial Bus.*
USSD: Unstructured Supplementary Service Data.

$\mathcal{W}$

Wi-Fi: *Wireless Fidelity.*
Wi-MAX: *Worldwide Interoperability for Microwave Access*.
WLAN: *Wireless Local Area Network.*
WPAN: *Wireless Personnal Area Network.*

$\mathcal{X}$

XHTML: *eXtensible HyperText Markup Language.*
XML: *eXtensible Markup Language.*

Introduction générale

La relation médecin-patient est usuellement restreinte dans l'espace et dans le temps ; fréquemment statique, au cabinet du praticien, lors de visites à domicile, ou de même à l'hôpital. Elle est constituée d'observations à intervalles donnés, parfois irréguliers, s'appuyant initialement sur une information exprimée par ce patient, donc subjective.

Le suivi n'est donc que rarement en temps réel, direct et permanent. La possibilité d'être répertorié pour une pathologie ou une dépendance et même d'avoir un diagnostic ou une consultation directe est donc difficile à aboutir.

La révolution technologique et l'évolution remarquable dans le domaine d'échange d'information et de communication a apporté une grande amélioration dans la qualité des services de santé, dont la dimension ambulante et mobile du patient est enfin appelée à entrer dans le champ d'analyse de la médecine de ville. La convergence entre les technologies de l'information et de communication, et l'ensemble des spécialités médicales a donné naissance à un nouveau domaine : « la télémédecine ». Des capteurs non intrusifs placés et laissés sur le corps humain, au quotidien pour le suivi des personnes par la mesure de leur célérité, leur température corporelle, leur pression artérielle ou leur fréquence cardiaque présentent une voie prometteuse en médecine.

La révolution de la téléphonie mobile représente également un facteur principal dans l'évolution du domaine médical. L'arrivée des Smartphones -téléphone polyvalent- développés sous Android, iOS, Windows Phone, ou BlackBerry, a évoqué l'apparition d'une nouvelle génération des terminaux mobiles, qui dépasse largement leur fonction principale (téléphoner).

Ces innovations remarquables dans le domaine de la téléphonie mobile ont été accompagnées par une immense évolution des réseaux de télécommunication (internet, 3G, 4G…), ce qui a donné du nouveau sens à la pratique de la médecine. L'ensemble de ces technologies interconnectées via un protocole (IP,…etc.) sous forme d'architecture en réseau tournée vers le confort de la pratique médicale et une amélioration des soins délivrés au patient tend à devenir à moyen terme un réel dispositif médical.

Dans notre travail nous allons réaliser une application mobile ambulatoire pour enrichir le suivi d'un patient par un appareil de mesure de fréquence cardiaque ; en ajoutant l'aspect direct à l'échange d'information médecin-patient où les mesures seront transmises directement et en temps réel au médecin pour donner les procédures à faire à fin d'éviter la mise en danger de la santé de patient.

Le premier chapitre sera dédié aux généralités sur la télémédecine, l'appareil de mesure utilisé « Holter ECG » et les différentes technologies utilisés pour la réalisation de cet œuvre.

Dans le deuxième chapitre, nous allons détailler la plateforme utilisée pour le développement de notre application « Android ».

Le troisième chapitre sera consacré à la réalisation de l'application envisagée et sa mise en œuvre.

Chapitre 1:
Généralités sur la télémédecine

1. Introduction

Ce travail portera sur la mise en application de la télémédecine dans le cadre du suivi d'un patient par la mesure de sa fréquence cardiaque, pour cela lors de ce premier chapitre ; nous allons donner un aperçu sur la télémédecine et ses différentes disciplines. Nous allons parler par la suite de la fréquence cardiaque et les différentes techniques de mesure, en donnant ainsi une description fonctionnelle de l'appareil utilisé dans notre projet « Holter ECG ». Nous terminerons par une approche technologique des différents réseaux qui participent au fonctionnement de notre application.

2. Télémédecine

2.1 Définition de la télémédecine

La loi n°2009-879 du juillet 2009 ainsi que le décret du 19 octobre 2010 en France a donné une définition précise de la télémédecine et fixé son cadre réglementaire « *La télémédecine est une forme de pratique médicale à distance utilisant les technologies de l'information et de la communication* ».

La télémédecine regroupe l'ensemble des pratiques issues de la rencontre des technologies de l'information et de la communication (TIC) et de la médecine. Cela couvre un domaine assez large allant du transfert de données à la relation patient –médecin. Elle représente alors les actes médicaux effectués à distance nécessitant l'intervention d'un médecin, notamment: examens, partage de données médicales, suivi, diagnostic… etc.

En général, la télémédecine a pour rôle l'accès aux soins à distance, et l'échange de l'information médicale afin d'évaluer l'état du patient. Elle représente un enjeu considérable pour l'amélioration des conditions de soin et de vie de beaucoup de personnes.

Dans les années soixante à soixante-dix, les premiers programmes de télémédecine ont été adoptés par les pays les plus vastes où la densité de population est faible pour répondre au problème d'isolement géographique de certaines populations. En effet, ce type d'organisation propose une solution liée à la difficulté d'accès aux centres de soins spécialisés. Selon, les premières expérimentations ont ainsi été implémentées et installées, par exemple en Australie (suivi psychothérapique à distance), en Écosse (dermatologie et médecine à distance pour les plateformes pétrolières) et dans les zones rurales des États Unis (télé soin).

2.2 Actes de télémédecine

Les applications multiples de la télémédecine et qui couvre un grand nombre de spécialités médicales, peut être classée en quatre principales actes à savoir :

- La téléconsultation ;
- La télé expertise ;
- La télésurveillance ;
- La téléassistance ;

2.2.1 Téléconsultation

Acte médical réalisé en présence du patient qui dialogue avec le médecin. Elle permet au patient d'avoir un diagnostic à distance par un médecin ; «*La consultation repose sur l'écoute, c'est donc facile de l'organiser en visioconférence. La distance est même appréciable pour des patients qui se sentent agressés par une présence physique*. » [**1**]. Le patient peut ainsi être assisté par un autre professionnel de santé.

2.2.2 Télé expertise

Tout acte diagnostique et/ou thérapeutique qui se réalise en dehors de la présence du patient. Il s'agit en général d'un échange entre deux ou plusieurs médecins qui arrêtent ensemble un diagnostic et/ou une thérapeutique sur la base des données cliniques, radiologiques ou biologiques qui figurent dans le dossier médical du patient.

2.2.3 Télésurveillance

Transmission à distance de données physiologiques et biologiques aux fins de suivi, d'interprétation et de prise de décision clinique. La prise en charge et le suivi des patients à domicile seront grandement facilités par le soutien d'appareils innovants de traitement d'informations visuelles et sonores, avec déclenchement et gestion des alarmes. L'autonomie des personnes âgées est également un secteur où l'innovation monte en puissance. La possibilité d'aider à la prévention de la dépendance en mettant en place un suivi des troubles liés à la mobilité, et au déroulement des actes du quotidien, est envisagée.

2.2.4 Télé assistance

L'Acte médical au cours duquel un médecin assiste à distance un autre médecin en train de réaliser un acte médical ou chirurgical. Elle a pour but de fournir quand c'est possible à

distance une aide directe à la personne à risque. *« Il y a peu de temps, j'étais en visioconférence avec la maison de santé de Bletterans où un médecin avait deux patients présents à ses côtés et me demandait un deuxième avis. Efficace et pas onéreux »* [**2**].

2.3. Intérêts et apports de la télémédecine

L'intérêt de la télémédecine se situe à trois niveaux : Elle permet d'assurer la continuité de l'accès aux soins pour une partie importante de la population, elle est un outil précieux pour traiter des urgences médicales et enfin sa mise en œuvre favorise les réseaux multidisciplinaires dont la médecine de demain aura besoin.

La télémédecine est un outil indispensable de l'aménagement du territoire car sa mise en place est la condition de survie des hôpitaux ruraux et le gage de qualité de la médecine libérale. Les hôpitaux ruraux ne peuvent pas assurer la présence permanente de médecins et de spécialistes de la même façon qu'un hôpital général plus important. Si ces procédés sont intelligemment mis en œuvre on peut assurer une meilleure continuité des soins et la permanence de ces derniers tout en réalisant des économies par la diminution des transferts. *«L'hospitalisation en urgence s'avère en effet très dommageable pour les personnes et extrêmement coûteuse pour la collectivité»* [**3**]. On observe ainsi à ce niveau que l'intérêt du patient et l'intérêt de la société se rejoignent. L'hôpital a probablement pour vocation d'être le pivot de la mise en œuvre des procédés de télémédecine.

La mise en œuvre de la télémédecine présente donc ces atouts majeurs :

- **l'intérêt du patient**. Il reste primordial : la télé expertise permet d'éviter, grâce à la relation à distance entre compétences réparties, les transferts inutiles, et parfois dangereux, pour des patients cliniquement fragiles.
- **l'aide diagnostique** apportée permet également d'ajuster la prise en charge thérapeutique du patient, que ce soit sur un plan local ou grâce à l'orientation vers un service plus adapté à son état.

L'innovation informatique bénéfice aussi au transfert d'informations médicales, elles s'accroissent en nombre, en qualité, fichiers ou images. De nos jours, le suivi du patient s'en trouve déjà grandement amélioré à l'échelle mondiale. Une personne isolée peut bénéficier d'une consultation et d'un diagnostic à distance. Les applications de télédiagnostic permettent également le partage des compétences entre praticiens, pour de meilleures expertises. Evoquons

aussi la télé-psychiatrie, qui est en plein déploiement, et qui favorise le suivi des patients et vient lutter efficacement contre la pénurie des psychiatres qui frappe les campagnes. Au-delà encore, avec une technologie encore plus précise et entièrement maitrisée, la télé-chirurgie permettra, peut-être bientôt et couramment, à un chirurgien d'opérer un patient à plusieurs centaines de kilomètres, grâce à une technologie intégrant des logiciels d'imagerie médicale couplés à des applications de robotique.

2.4. Inconvénients

La télétransmission des chiffres et d'informations médicales nécessite la formation des acteurs (patients comme professionnels). Il existe des coûts de mise en œuvre des appareils, tant dans leur acquisition, que leur entretien ou leur maintenance. Actuellement, nous ne disposons pas d'évaluation économique qui permettrait de juger de la place de la télémédecine en dehors d'un contexte expérimental. De même, nous ne savons pas actuellement déterminer le profil de patients qui seraient susceptibles de bénéficier au mieux de la démarche de télétransmission et de télésurveillance. Les avantages médicaux de la télétransmission des résultats d'auto-mesure de n'importe quel paramètre physique depuis le domicile vers un professionnel sont à mettre en regard des inconvénients de coûts, lesquels évoluent rapidement en ce qui concerne les nouvelles technologies de l'information et de la communication (NTIC).

2.5. Télé-cardiologie

Application de la télémédecine au domaine de la cardiologie, la télé-cardiologie permet de suivre à distance les porteurs de stimulateurs ou de défibrillateurs cardiaques implantables, et d'intervenir au plus près de la survenue d'une anomalie [**4, 22**]. Grâce à ce système, les médecins reçoivent ces informations de façon quasi instantanée et peuvent ainsi intervenir immédiatement. Certains systèmes transmettent en effet les informations quotidiennement, à heure fixe, au centre cardiologique surveillant le patient. En cas d'évènement grave, potentiellement mortel, la transmission est immédiate et peut être étendue au cardiologue de ville en charge du patient. "*En cas d'alertes rythmiques, notamment, les patients souffrant de troubles du rythme pourront être convoqués sans délai au centre cardiologique pour des examens complémentaires, des ajustements du traitement médical ou des réglages de l'appareil. Quant aux insuffisants cardiaques traités par resynchronisation, les fonctions diagnostiques des appareils sont aujourd'hui suffisamment fines pour détecter les signes annonciateurs d'une décompensation (NDLR : aggravation du mauvais fonctionnement du cœur) avant même que le patient ressente les*

premiers symptômes. De nombreuses hospitalisations pourront ainsi être évitées par une prise en charge plus précoce". Pour ce cardiologue, « *la téléconsultation règle un problème géographique et c'est un vrai gain de temps pour de nombreux patients, pas toujours mobiles et qui doivent consacrer au minimum une demi-journée pour venir en consultation spécialisée* » [**5**].

Par ailleurs, en cas d'accident cardiaque, le temps presse ; « *Pour un arrêt cardiaque, une minute de perdue, c'est 10 % de chance de survie en moins* ».Alors que les secours mettent en moyenne 9 à 10 minutes pour arriver, « *l'urgence est de gagner du temps pour préserver le cœur et le cerveau, et prévenir des séquelles cardiaques et neurologiques. Il est important que chacun d'entre nous soit capable de pratiquer les "gestes qui sauvent" et de savoir qu'il est simple d'utiliser un défibrillateur* » [**6**].

Sans pour autant remplacer la consultation conventionnelle, ce mode de suivi des patients représente une avancée majeure tant sur le plan médical qu'économique puisqu'il réduit les coûts de transport et de consultations.

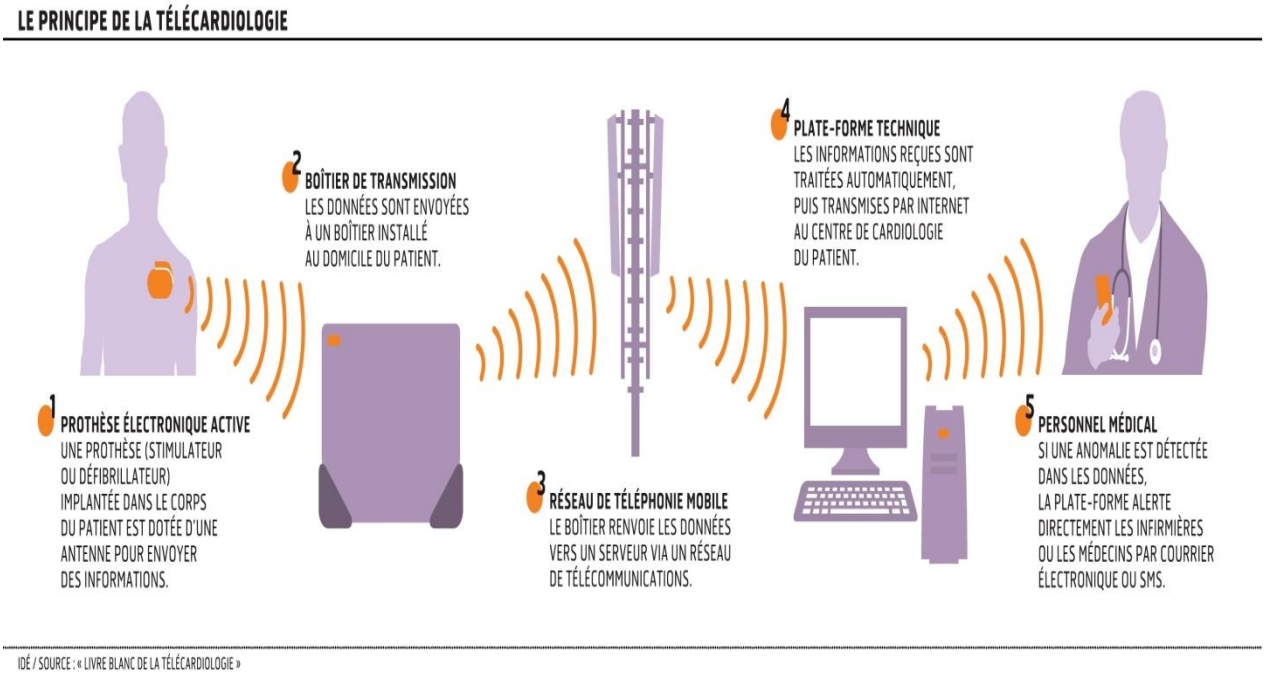

Fig.1.1. Principe de la télé-cardiologie.

3. Rythme cardiaque

Le cœur est un muscle qui se contracte spontanément. Son rythme ne se commande pas volontairement. La régulation cardiaque est sous la dépendance de deux systèmes nerveux coordonnés entre eux, l'un interne au cœur et l'autre externe.

Le système nerveux externe ajuste la fréquence et la force de contraction du cœur pour répondre aux besoins de l'organisme (selon les efforts physiques fournis par exemple).

Le système interne permet de coordonner les battements des oreillettes et des ventricules. Le battement des oreillettes déclenche normalement celui des ventricules. Il existe une synchronisation entre les battements des oreillettes et ceux des ventricules. Chez les personnes atteintes de troubles du rythme cardiaque, le rythme cardiaque ou la synchronisation entre oreillettes et ventricules sont perturbés.

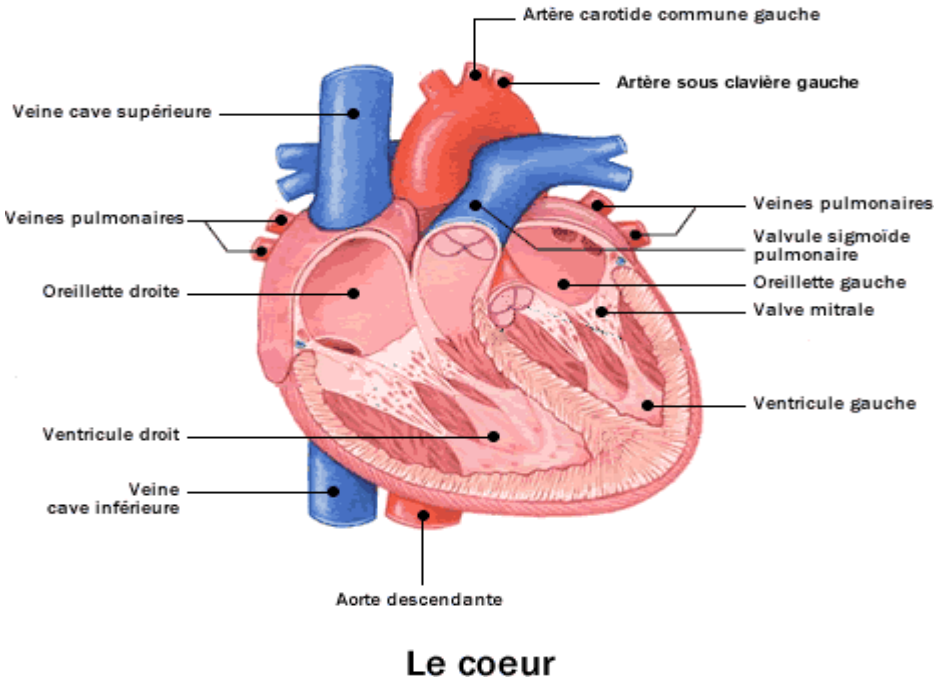

Fig.1.2. Le cœur.

3.1. Fréquence cardiaque

Indicateur des plus fiables de notre condition physique, la fréquence cardiaque est au cœur du débat. De repos, maximale ou d'effort, elle varie en fonction de notre activité physique.

La fréquence cardiaque se mesure en nombre de battements par minute. Lors de chaque battement le cœur éjecte du sang oxygéné dans les artères du corps via l'aorte et du sang désoxygéné vers le poumon via l'artère pulmonaire.

Chez un individu sain, la fréquence cardiaque n'est pas toujours constante, mais elle change sur la base du niveau d'activité et du stress. Selon le tableau de la fréquence cardiaque idéale, le rythme cardiaque normal pour les adultes au repos est compris entre 60 et 100 battements par minute (BPM). Le taux peut descendre à 40 BPM quand on dort. Certains athlètes peuvent avoir un rythme au-dessous de 60 parce que le muscle cardiaque est plus fort et il pompe une quantité de sang supérieure. Un niveau plus élevé de fréquence cardiaque indique que le cœur doit travailler plus durement. Le battement cardiaque peut arriver jusqu'à 150 ou à 200BPM à l'effort. Le battement cardiaque pour un nouveau-né est d'environ 110 BPM.

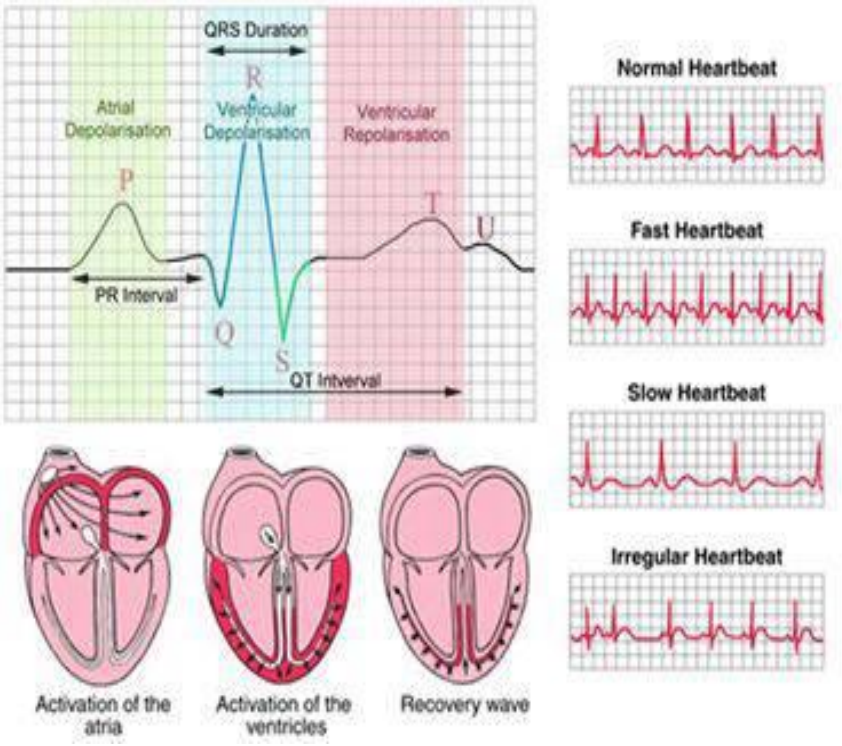

Fig.1.3. Le battement de cœur normal et les cas d'anomalies.

3.2. Insuffisance cardiaque

L'insuffisance cardiaque est une maladie chronique, c'est-à-dire une maladie qui nécessite un traitement médicamenteux quotidien et un suivi régulier. L'insuffisance cardiaque survient lorsque le muscle du cœur n'arrive plus à se contracter normalement et à envoyer dans l'organisme l'oxygène et les substances nutritives nécessaires. Elle est due le plus souvent à une atteinte des artères du cœur (appelées artères coronaires) ou à une hypertension artérielle.

3.3. Troubles de rythme cardiaque

Lorsque les troubles se déclarent dans les oreillettes, nous parlons d'arythmie supra ventriculaire. S'ils apparaissent dans les ventricules, nous parlons d'arythmie ventriculaire. Les principaux troubles du rythme sont les suivants :

- La bradycardie

  C'est une anomalie de la conduction électrique qui entraîne une diminution de la fréquence cardiaque. Nous parlons de bradycardie quand la fréquence cardiaque descend au-dessous de 60 battements par minute.

- La tachycardie

  Elle se traduit par une augmentation de la fréquence cardiaque. Nous parlons de tachycardie quand la fréquence cardiaque est au-dessus de 100 battements par minute au repos.

- La fibrillation auriculaire

  Il s'agit d'une désorganisation des contractions au sein des deux oreillettes *q*ui deviennent alors inefficaces. C'est le plus fréquent des troubles du rythme.

- L'arythmie ventriculaire

  Les ventricules peuvent, eux aussi, se comporter de façon autonome. Ainsi, ils n'obéissent plus à la commande des oreillettes : c'est l'arythmie ventriculaire (tachycardie et fibrillation ventriculaire).

- Les extrasystoles

  Ce sont des battements du cœur qui surviennent en dehors des battements normaux. Ces battements perturbent le rythme cardiaque normal et s'accompagnent d'une sensation de "palpitations" ou de "pause cardiaque".

- Les troubles de la conduction cardiaque

  Ils résultent d'une mauvaise transmission des signaux électriques dans les oreillettes, dans les ventricules ou entre les oreillettes et les ventricules.

Nous distinguons donc différentes formes de troubles de la conduction cardiaque en fonction de l'endroit où le trouble se situe.

3.4. Mesure de fréquence cardiaque

Elle se fait traditionnellement en appliquant les doigts sur le trajet de l'artère radiale au niveau du poignet, Une légère pression des doigts permet de percevoir les pulsations du sang dans l'artère, ce geste est communément appelé « prendre le pouls ». Il existe d'autres endroits du corps où cette palpation est possible (par exemple là où les artères sont assez proches de la peau : au niveau du cou, à l'aine …). La mesure peut également être faite automatiquement avec des appareils : électrocardiogrammes, appareils oscillométriques de mesure de la pression artérielle, oxymétries de pouls.

Depuis 2011, des applicatifs pour téléphones mobiles (iPhone et Androïd) proposent une mesure du pouls par simple application de l'index sur l'objectif de l'appareil photo au dos du Smartphone.

3.4.1. Holter ECG

L'enregistrement ECG par la méthode Holter (appelée aussi enregistrement Holter) est habituellement utilisé pour diagnostiquer les anomalies du rythme cardiaque, plus spécifiquement pour trouver l'origine des palpitations ou des étourdissements. Pour ce faire, vous devez porter un petit appareil d'enregistrement de l'ECG, appelé *moniteur Holter*, qui est relié à de petits disques métalliques (électrodes) placés sur votre poitrine, qui permettent une lecture de votre pouls et de votre rythme cardiaque au cours d'une période minimale de 24 heures. Votre rythme cardiaque est alors transmis et enregistré sur bande, puis numérisé par l'ordinateur, afin d'être analysé pour découvrir ce qui cause votre arythmie. Il est à noter qu'avec certains moniteurs, vous devez appuyer sur un bouton pour enregistrer le rythme cardiaque au moment où vous ressentez des symptômes.

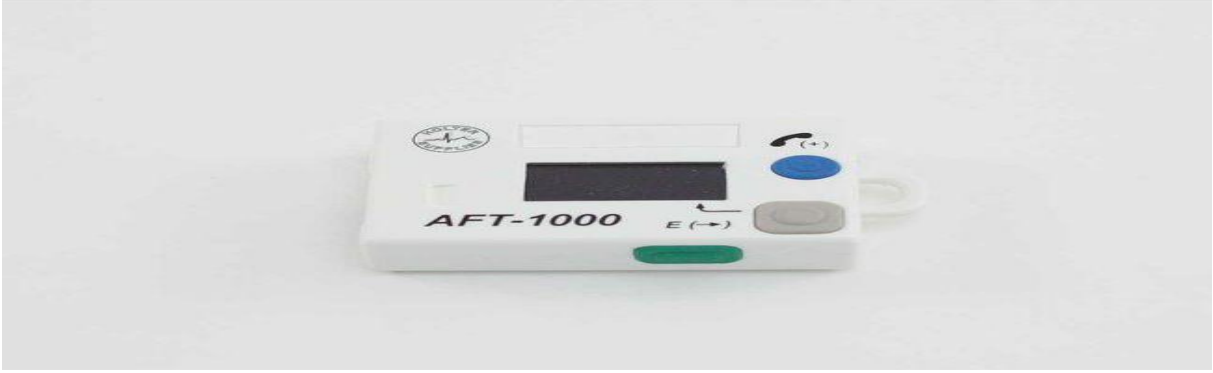

Fig.1.4. Exemple d'un « *Holter ECG*» ; nommé « *AFT 1000* ».

L'holter comporte deux parties :

- un boîtier d'enregistrement confié au patient et relié à ce dernier par des électrodes (cinq le plus souvent, permettant d'enregistrer deux dérivations à peu près orthogonales, la dernière électrode faisant office de neutre) fixées à la peau par un adhésif. Ce boîtier comporte un bouton que le sujet peut actionner s'il ressent quelque chose durant l'enregistrement (palpitations, douleurs thoraciques ou autre) : un repère sera inscrit alors sur l'enregistrement, permettant au médecin d'aller directement consulter le tracé à ce moment et en faire ainsi le diagnostic.
- une console de traitement : il s'agit d'un ordinateur muni d'un logiciel permettant la visualisation et l'analyse semi-automatique de l'enregistrement.

3.4.1.1. Logiciel de traitement des données « *Quick Reader* »

Les enregistreurs « AFT 1000 » et « 250 » de « Holter Supplies » embarquent sur sa mémoire le logiciel de lecture « Quick Reader ». En branchant un PC correctement équipé à l'enregistreur à l'aide d'un simple câble USB, l'utilisateur peut installer immédiatement sur son ordinateur le logiciel de lecture. Il peut ainsi transférer les données enregistrées, les vérifier, et libérer l'enregistreur pour un nouvel enregistrement. Une clé USB ("Quick Reader Soft Key" distribué par Holter Supplies) permet alors d'accéder à toutes les fonctionnalités. Il permet d'obtenir un temps de seulement 4s pour un enregistrement de 2 pistes standard de 24h ; d'où vient son nom « Quick ».

Fig.1.5. Logiciel Quick Reader.

Exemples d'anomalies présentées par Quick Reader

- Bradycardie :

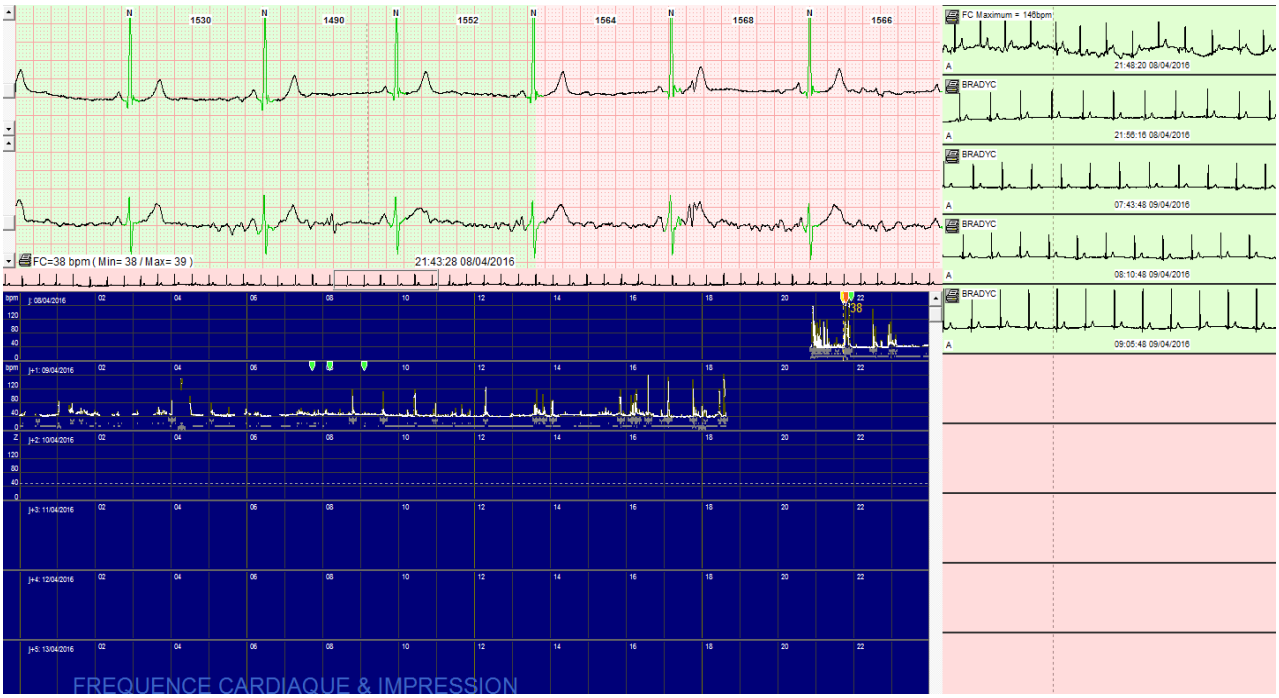

Fig.1.6. Exemple réel d'une Bradycardie.

- Tachycardie :

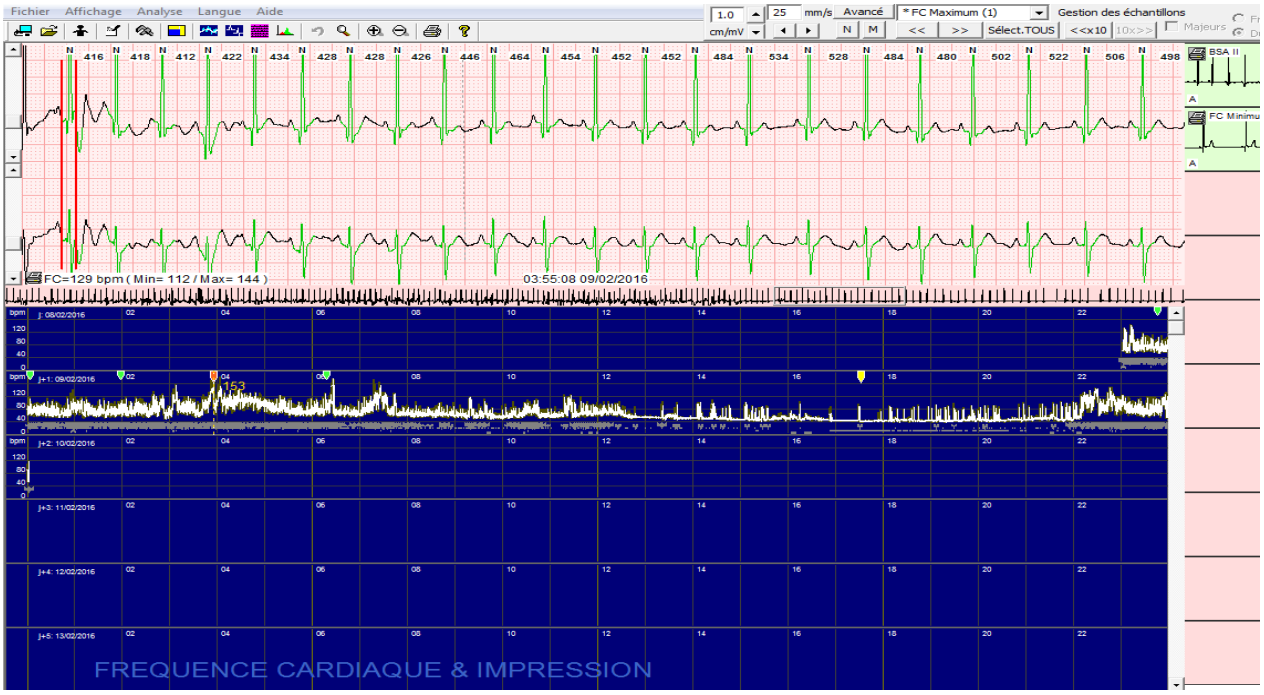

Fig.1.7. Exemple réel d'une Tachycardie.

- BAV :

Fig.1.8. Exemple réel d'une Arythmie ventriculaire.

4. Réseaux Bluetooth et GSM

La télémédecine représente le transfert électronique des données médicales comprenant le son et les images pour pratiquer la médecine à distance. Elle utilise donc des technologies de communication sans fils, pour permettre en premier lieu la mobilité des services de santé offerts. Parmi ces technologies nous présentons les deux principaux réseaux utilisés dans la suite de cet œuvre.

4.1. Réseau sans fil Bluetooth

La technologie Bluetooth a été originairement mise au point par Ericsson en 1994. En février 1998 un groupe d'intérêt baptisé Bluetooth Special Interest Group (Bluetooth SIG), réunissant plus de 2000 entreprises dont Agere, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia et Toshiba, a été formé afin de produire les spécifications Bluetooth 1.0, qui furent publiées en juillet 1999.

Bluetooth est une technologie de réseau personnel sans fils (noté WPAN pour Wireless Personal Area Network), c'est-à-dire une technologie de réseaux sans fils d'une faible portée permettant de relier des appareils entre eux sans liaison filaire. Contrairement à la technologie IrDa (liaison infrarouge), les appareils Bluetooth ne nécessitent pas d'une ligne de vue directe pour communiquer, ce qui rend plus souple son utilisation et permet notamment une communication d'une pièce à une autre, sur de petits espaces. L'objectif de Bluetooth est de permettre de transmettre des données ou de la voix entre des équipements possédant un circuit radio de faible coût, sur un rayon de l'ordre d'une dizaine de mètres à un peu moins d'une centaine de mètres et avec une faible consommation électrique. Le Bluetooth permet d'obtenir des débits de l'ordre de 1 Mbps, correspondant à 1600 échanges par seconde en full-duplex.

4.4.1. Normes Bluetooth

Le standard Bluetooth se décompose en différentes normes :

- IEEE 802.15.1 définit le standard Bluetooth 1.x permettant d'obtenir un débit de 1 Mbit/sec.
- IEEE 802.15.2 propose des recommandations pour l'utilisation de la bande de fréquence 2.4 GHz (fréquence utilisée également par le WiFi). Ce standard n'est toutefois pas encore validé.
- IEEE 802.15.3 est un standard en cours de développement visant à proposer du haut débit (20 Mbit/s) avec la technologie Bluetooth.

- IEEE 802.15.4 est un standard en cours de développement pour des applications Bluetooth à bas débit.

4.2. Réseau GSM

Les progrès technologique dans le domaine des réseaux de télécommunications mobiles, ont vu l'apparition des technologies numériques au début des années 1990. En Europe (*GSM*), au Japon (*PDC*) et aux Etats Unis (*PCS*).Le réseau **GSM** constitue au début du 21ème siècle le standard de téléphonie mobile le plus utilisé en Europe. Il s'agit d'un standard de téléphonie dit « de seconde génération » (2G) car, contrairement à la première génération de téléphones portables, les communications fonctionnent selon un mode entièrement numérique. Baptisé « Groupe Spécial Mobile » à l'origine de sa normalisation en 1982, il est devenu une norme internationale nommée « Global System for Mobile communications » en 1991.

La norme GSM autorise un débit maximal de 9,6 kbps, ce qui permet de transmettre la voix ainsi que des données numériques de faible volume, par exemple des messages textes (**SMS**, pour *Short Message Service*) ou des messages multimédias (**MMS**, pour *Multimedia Message Service*). Le standard GSM utilise les bandes de fréquences 900 MHz et 1800 MHz.

5. Conclusion

Lors de ce premier chapitre, nous avons essayé, dans un premier temps, de faire un petit tour d'horizon de l'une des applications des nouvelles technologies les plus remarquables, en l'occurrence la télémédecine; qui n'arrive toujours pas à être déployée tel qu'il se doit. Puis nous avons abordé le domaine de cardiologie médicale pour lequel sera appliquée notre solution de télésurveillance.

Nous avons également définit la méthode utilisée dans le projet pour la mesure de la fréquence cardiaque y compris l'appareillage adéquat. Vers la fin de ce chapitre, nous avons donné une petite approche concernant les différentes technologies innovantes dans la réalisation de notre application de télésurveillance.

Le chapitre suivant va être consacré au développement des applications sous l'environnement « *Android* ».

Chapitre 2:
Développement d'applications sous Android

1. Introduction

De par son ergonomie et les capacités du matériel, l'arrivée de l'iPhone d'Apple a bouleversé le paysage des systèmes d'exploitation mobiles, tant par les usages proposés que par les possibilités offertes aux utilisateurs avec le marché d'applications Apple Store. La firme de Cupertino a su en quelques mois instaurer sa suprématie dans le monde de la téléphonie et surtout initier de nouveaux usages. Aujourd'hui, les utilisateurs ne cessent de comparer leur téléphone à l'iPhone : ergonomie, écran tactile, simplicité, connexion Internet quasi permanente, multitude d'applications, etc. De créateur de nouveaux usages, l'iPhone est devenu un véritable standard pour les utilisateurs.

Si l'iPhone d'Apple semble marquer d'une pierre blanche le début de cette nouvelle ère, la concurrence n'est pas en reste et a bien suivi le mouvement. Nokia a sorti son 5800, BlackBerry le Storm et Palm pourrait bien renaître de ses cendres avec le Pré. Parmi ces alternatives à l'iPhone, il y a une qui sort du lot, à la fois par ses qualités intrinsèques et son mode de développement ouvert. Elle a d'inédit qu'il ne s'agit pas vraiment d'un téléphone mais plutôt d'un système qui s'installe sur du matériel issu de différents constructeurs. Cette particularité pourrait bien faire que ce système s'impose sur le marché des mobiles à l'instar du PC sur le marché des micro-ordinateurs. Ce système, c'est bien sûr Android.

Android est un sujet d'étude très vaste qui mérite un livre à part entière. En effet, son modèle de programmation, son interface graphique, ses composants logiciels, ses fonctionnalités de sauvegarde de données ou encore ses API réseau et de géo-localisation sont uniques. Il participe d'une façon efficace dans le changement des comportements des patients et des systèmes de santé. L'objectif est de déplacer les lieux de soin : aller des hôpitaux vers les soins à domicile, et aller même jusque dans les poches des citoyens (avec les smartphones).

Pour cela, et lors de ce deuxième chapitre nous allons essayer de présenter les concepts de base des smartphones en nous intéressant plus précisément à la plateforme Android, ainsi que l'outil de développement d'applications « MIT App Inventor ».

2. Téléphone polyvalent « *Smartphone* »

2.1. Présentation du Smartphone

Le «Smartphone » (téléphone intelligent) désigne un téléphone portable, à multifonctions et qui dépasse largement la fonction principale d'un téléphone mobile ordinaire (téléphoner). Similaire à un ordinateur, il peut exécuter divers applications grâce à un système d'exploitation spécialement conçu pour mobiles, et donc, en particulier, fournir de nouvelles fonctionnalités comme: l'agenda, la télévision, le calendrier, la navigation sur le Web, la consultation et l'envoi de courrier électronique, la géo-localisation, la calculatrice, la boussole, l'accéléromètre, le gyroscope, la messagerie vocale, la cartographie numérique, etc. Ces nouvelles applications sont publiées dans un magasin d'applications en ligne différent pour chaque système d'exploitation comme Google Play Store sur Android ou, encore, l'AppStore sur iOS [**7**].

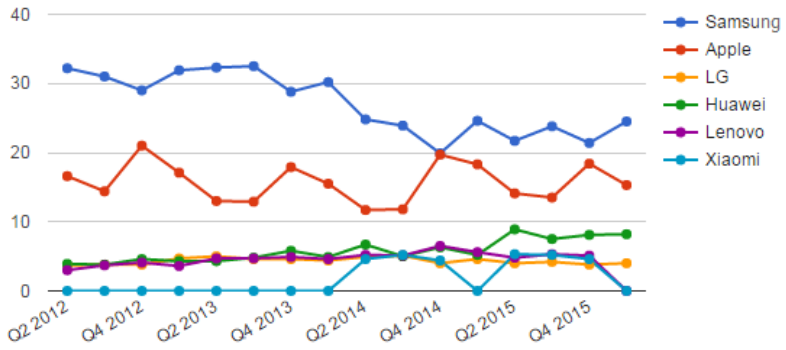

Fig.2.1. croissance des ventes des smartphones.

Le meilleur moyen d'utiliser Android est donc de disposer d'un téléphone ou d'un PDA sur lequel la plate-forme est installée. Les premiers modèles (téléphones G1) sont arrivés sur le marché aux USA fin 2008, puis ont été diffusés dans de nombreux autres pays. En outre, pour répondre à la demande de nombreux développeurs, Google a mis sur le marché un téléphone qui leur est dédié, le Dev Phone 1. Contrairement aux téléphones destinés au grand public, dont les couches basses sont parfois verrouillées, le Dev Phone 1 est prévu pour être modifié facilement : il est en effet possible de remplacer l'image système (firmware) par une image personnalisée. Précisons que tous les téléphones Android acceptent les programmes écrits en Java.

Si début 2009 seul le téléphone de HTC (le G1) était disponible, le nombre de modèles d'appareils Android a augmenté depuis à un rythme soutenu. Du Motorola Droid aux HTC Hero,

Nexus One, en passant par le Sony Ericsson Xperia X10 et autres, de plus en plus de modèles sont équipés d'Android.

Fig.2.2. Téléphone G1 de HTC.

2.2. Systèmes d'exploitation

« *Le système d'exploitation (SE, en anglais Operating System ou OS) est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur et les applications informatiques de l'utilisateur (traitement de texte, jeu vidéo, etc.). Il fournit aux programmes applicatifs des points d'entrée génériques pour les périphériques.* » [**8**].

Les systèmes d'exploitation pour les mobiles sont des systèmes d'exploitation prévus pour fonctionner sur les mobiles. Ces derniers sont des appareils à ressources limitées ou faibles. C'est pourquoi, on doit adapter le SE à leurs limites. Le système d'exploitation pour les mobiles détermine les caractéristiques, la performance et la sécurité, en fournissant des API pour déployer et gérer les applications (la fonction principale des mobiles, c'est communication entre eux par les services comme : téléphone, SMS, MMS, etc.).

Actuellement, il existe plusieurs SEs pour les mobiles comme Windows Mobile, Palm OS, Symbian, BlackBerry qui sont les systèmes propriétaires. De plus, il y a les plateformes libres ou code source ouvert comme Moblin.org, Ubuntu MID Edition, Android, etc. Chaque type de SE est souvent approprie à quelques modèles concrets des mobiles. Cependant, chaque type de SE a des avantages et des limitations (Annexe I).

2.2.1. BlackBerry

BlackBerry a été créé par Research In Motion (RIM), une société Canadienne. Il est une plateforme très populaire particulièrement dans Amérique du Nord. Au commencement, BlackBerry est développé pour le businessman. Il y a donc des applications pour le business comme : les messages électronique, les messages PIN, les messages texte (SMS), les messages MMS, le BlackBerry Messenger, le web navigateur, la tâche, le mémo, etc.

Il est donc optimal pour les grands déploiements, mais pour les moyens ou les petits déploiements, il est préférable de choisir d'autres plateformes.

2.2.2. Palm OS

Palm OS est développé par la société Palm. En général, Palm OS est logique, intuitif et simple. En détaille, Palm OS est facile d'utiliser et simple d'apprendre. Il optimise les étapes pour naviguer entre les écrans et choisir les applications. Il est mono tâche. C'est-a-dire, le système d'exploitation ne peut gérer qu'une seule tâche en même temps tandis que les autres plateformes sont multitâches.

Il présente alors un bon choix pour les consommateurs mais non pas les sociétés.

2.2.3. Symbian

Symbian est développé par la société Symbian. « *Symbian a commandé 57 pour cent des ventes globales à des utilisateurs finaux au cours du deuxième trimestre, en 2008 dans le marché des systèmes d'exploitation pour les Smartphones* » [**9]**. Pour le système de messagerie électronique, Symbian est soutenu par plusieurs systèmes de messagerie électronique comme BlackBerry (via BlackBerry Connect) et Microsoft (via Exchange ActiveSync), etc. Pour l'utilisation, Symbian est plus facile que Windows Mobile mais plus difficile que Palm OS.

En conclusion ; Symbian est puissance et très populaire particulièrement en Europe. Il est un bon choix pour les consommateurs et les sociétés grâce au ses appareils Nokia.

2.2.4. Ubuntu MID Edition

Ubuntu MID Edition est développé par la communauté Ubuntu. Il est soutenu par les sociétés Canonical et Intel. Les MIDs (Mobile Internet Device) sont Des appareils intermédiaires entre le téléphone portable et l'ordinateur. C'est un projet en code source ouvert, très souple et personnalisable. Il est utilisé dans quelques modèles d'appareils utilisant une plateforme Intel Atom, ou McCaslin comme Samsung Q1 Ultra, HTC Shift, Nokia N800 web tablette.

En général, Ubuntu MID Edition est une plateforme pour les MIDs mais n'est pas pour le téléphone portable.

2.2.5. Windows Mobile

Windows Mobile est développé par la corporation Microsoft. Il a la capacité de faire fonctionner des logiciels sur « Windows » (seulement Windows). Il soutient beaucoup de types d'audio, de vidéo…etc. Il soutient aussi la capacité de télécharger, de jouer les chansons et de regarder le TV en ligne. Grâce à ces caractères, dans le domaine de loisir, il est le plus bon.
En conclusion, grâce à la popularité de Microsoft et la variété d'applications offertes, Windows Mobile est supporté par beaucoup de sociétés.

2.2.6. Android

Android est développé par l'Open Handset Alliance. Il a été annoncé en 2007. De plus, en 2008, il est devenu une plateforme en code source ouverte. Selon Google qui est un majeur distributeur, Android est une plateforme puissante, moderne, sécurisée et ouverte. Android est basé sur le noyau Linux et utilise le code java pour les applications.
Android est donc une plateforme très flexible et innovante dans le domaine de la téléphonie mobile intelligente. Dans la suite de cet œuvre, nous allons détailler cette plateforme qui sera utilisée pour la réalisation de notre projet.

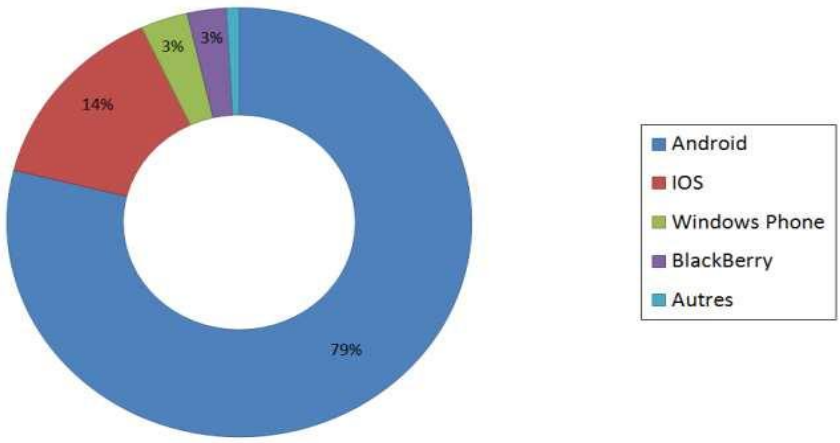

Fig.2.3. Comparaison entre les différentes versions des OS pour téléphonie mobile.

3. Plateforme « *Android* »

Devant les lacunes des offres des constructeurs historiques, une coalition s'est créée pour mettre au point et promouvoir un système d'exploitation mobile concurrent. Ce rassemblement, qui a vu le jour fin 2007, se nomme l'*Open Handset Alliance* et se compose aujourd'hui de acteurs qui ont pour objectif de créer et de promouvoir un système complet, ouvert et gratuit dans le monde du mobile : « Android ».

Dès son origine, la démarche de Google a été d'ouvrir le développement d'Android en rassemblant autour de lui et au travers de l'Open Handset Alliance (OHA) un maximum de sociétés. Les membres de ce consortium sont très variés : nous y trouvons des fabricants de téléphones connus tels que Sony Ericsson, Samsung ou Motorola, des opérateurs de téléphonie comme Sprint, T-Mobile ou NTT DoCoMo, des sociétés Internet, Google évidemment mais aussi eBay, des constructeurs de puces électroniques Intel, nVidia, ou encore des acteurs du marché du GPS comme Garmin.

Comparer les offres concurrentes à celle d'Android est compliqué tant les stratégies et les approches sont différentes ; certains proposent un système intégré (téléphone+ OS), d'autres tentent d'unifier le modèle d'exécution des applications entre les ordinateurs de bureau et les mobiles. Android quant à lui se positionne comme un système d'exploitation et un environnement de développement open source, dédié aux appareils mobiles, et indépendant de toute plateforme matérielle spécifique. « *Android est une plateforme logicielle destinée aux appareils mobiles comprenant un système d'exploitation, des applications, et les couches intermédiaires nécessaires pour faire fonctionner le tout* » **[10]**.

Nous remarquons d'entrée de jeu que Google se place dans le domaine du logiciel. Android n'est donc pas lié à un appareil donné ; il a au contraire vocation à être intégré par différents constructeurs dans des appareils mobiles, moyennant le respect de quelques règles de compatibilité. Tordons cependant le cou aux simplifications et notamment à celle trop souvent faite : « Android = Google phone ». Google est certes l'acteur majeur de ce rassemblement autour d'Android, mais les téléphones Android disponibles sur le marché ne sont pas toujours reliés à cette firme, mis à part le « Nexus One », lancé tout début 2010 et qui est fabriqué spécialement pour Google, et les différentes versions successives de Nexus jusqu'au « Nexus 5 ». L'offre des constructeurs s'élargit d'ailleurs chaque jour davantage. Ainsi, les versions se succèdent rapidement et les changements qui les accompagnent sont souvent conséquents en termes de nouvelles fonctionnalités et d'améliorations. Cependant, ces évolutions n'ont pas toujours été

sans impact sur le bon fonctionnement et la compatibilité des applications entre versions, « *En 2010 Android était présent sur 26 smartphones, traduit dans 19 langues et vendu dans 48 pays par 59 opérateurs* » [**11**].

3.1. Versions Android

Les versions Android sont des mises à jour pour ce système d'exploitation mobile open source développées sous les noms de code dessert d'inspiration, avec chaque nouvelle version, qui arrivent dans l'ordre alphabétique avec nouvelles améliorations au niveau du SDK Android [**12]**.

Voici un aperçu des mises à jour Android, qui ont déjà été libérés :

3.1.1. **Cupcake** (1.5) - a fait ses débuts à l'automne 2008 : ajouts de clé : outils de reconnaissance vocale, un clavier virtuel, téléchargement de vidéo pour YouTube et de flux de données en temps réel.

3.1.2. **Donut** (v1.6) - a fait ses débuts à l'automne 2009: principaux ajouts: Support pour CDMA smartphones, des écrans supplémentaires et un moteur de synthèse vocale.

3.1.3. **Eclair** (v2.0) - a fait ses débuts en octobre 2009 : principaux ajouts : Support des appareils multitouches, nouvelle interface du navigateur, support de Microsoft Exchange, une interface unique pour gérer plusieurs comptes en ligne, support de touches programmables et une application de caméra améliorée (avec zoom numérique et le support de flash).

3.1.4. **FroYo** (v2.2) - a fait ses débuts à l'automne 2010 : principaux ajouts : USB support (pour transformer un smartphone en hotspotWi-Fi), des améliorations significatives de vitesse, support de Flash 10.1, la numérotation vocale via Bluetooth, la possibilité de stocker les applications sur les cartes de mémoire externe, mise à jour de navigateur avec JavaScript V8 de Google Chrome.

3.1.5. **Gingerbread** (v2.3) - a fait ses débuts en décembre 2010: ajouts de clé: Google Voice over Wi-Fi, des fonctionnalités de jeu améliorées, amélioration de Google Apps.

3.1.6. **Honeycomb** (v3.0) - a fait ses débuts en février 2011 : principaux ajouts : une mise à jour tablette orientée qui a livré une nouvelle interface optimisée pour les appareils

avec une taille d'écran supérieure (en particulier les comprimés), chat vidéo de soutien basé sur les protocoles de Google Talk, nouveau système barre Statut global et notifications et barre d'Action pour l'application de contrôle, onglets navigation Web, clavier logiciel optimisé et une nouvelle interface de courrier électronique.

3.1.7. **IceCream Sandwich** (v4.0) – a fait ses débuts en octobre 2011: principaux ajouts: une mise à jour Smartphone orientée selon le v3.0.1 du noyau Linux qui apporte de nombreuses fonctionnalités de Honeycomb de Smartphones, y compris le logiciel de reconnaissance de visage Unlockfacial, onglets de navigation Web capacités, contacts de réseaux sociales unifiés, capacités d'enregistrement vidéo 1080p et chat vidéo de soutien basés sur des protocoles de Google Talk.

3.1.8. **Jelly Bean** (v4.1, v4.2 et v4.3) – fait ses débuts en juin 2012: principaux ajouts: Advanced voix langage naturel, des capacités de commandement semblable à Apple Siri, interface améliorée et la réactivité globale Via « Project Butter», support Google Now, un navigateur web amélioré,… etc.

3.1.9. **KitKat** (v4.4) - a fait ses débuts en novembre 2013. Ajouts de clé: plein écran mode immersif, nouveau cadre de transitions, « Project Svelte, » un projet initié afin de réduire les besoins en mémoire de l'OS Android. À l'origine, cette version est dénommé Key Lime Pie, et annoncée en début en septembre 2013, ce qui signifie la barre de sucrerie emblématique.

Pour notre application, nous allons utiliser cette version vue quelle est présente dans la plupart des téléphones mobiles offerts sur marché.

3.1.10. **Lollipop** (v5.0) - a fait ses débuts en novembre 2014. Ajouts de clé: amélioré l'interface utilisateur de conception de matériel, continuité améliorée sur les appareils Android, support d'utilisateurs multiples, une option de compte d'utilisateur invité, un nouveau système de notification, prise en charge pour les processeurs 64 bits et plus.

3.1.11. **Marshmallow** "M Release" (version 6.0) - débuts en novembre 2015. Ajouts de clé : maintenant sur la fonctionnalité de robinet et autre Google maintenant améliorations, authentification par empreinte native prend en charge, intégration Android payer,

support de l'USB Type-C, amélioré la vie de la batterie, meilleure gestion des applications et bien plus encore.

Fig.2.4. Les différentes versions d'Android.

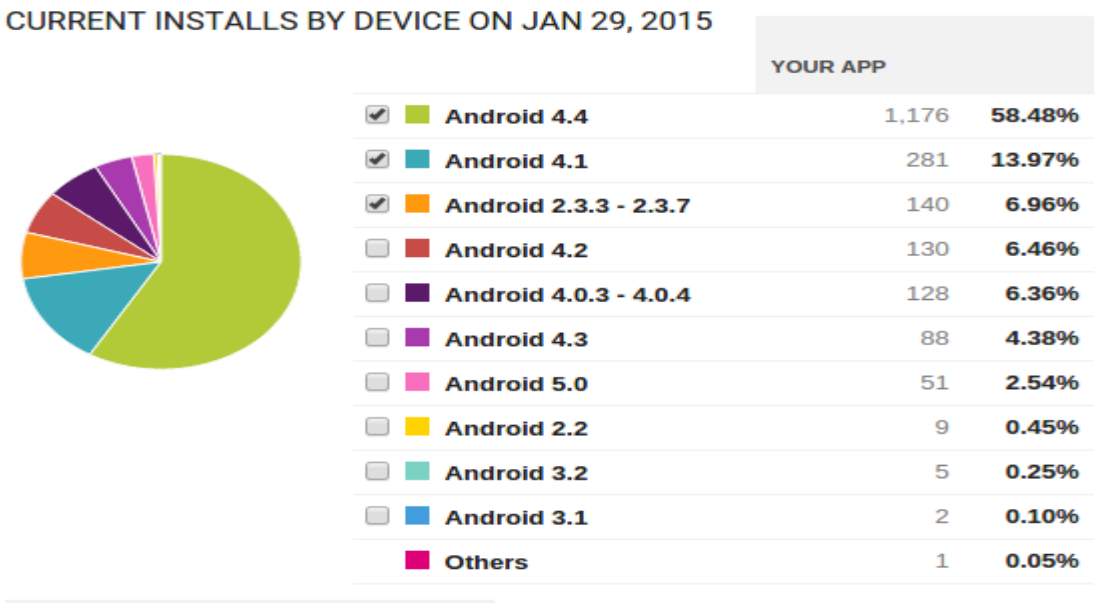

Fig.2.5. Comparaison entre les versions Android en 2015.

3.2. Architecture autour du noyau Linux

Android est conçue pour des appareils mobiles au sens large. Nullement restreinte aux téléphones, elle ouvre d'autres possibilités d'utilisation des tablettes, des ordinateurs portables, des bornes interactives, des baladeurs…

La plate-forme Android est composée de différentes couches :

- un noyau Linux qui lui confère notamment des caractéristiques multitâches ;
- des bibliothèques graphiques, multimédias ;
- une machine virtuelle Java adaptée : la *Dalvik Virtual Machine* ;
- un framework applicatif proposant des fonctionnalités de gestion de fenêtres, de téléphonie, de gestion de contenu... ;
- des applications dont un navigateur web, une gestion des contacts, un calendrier…etc.

Les composants majeurs de la plate-forme Android se trouvent dans le schéma suivant [**13]** :

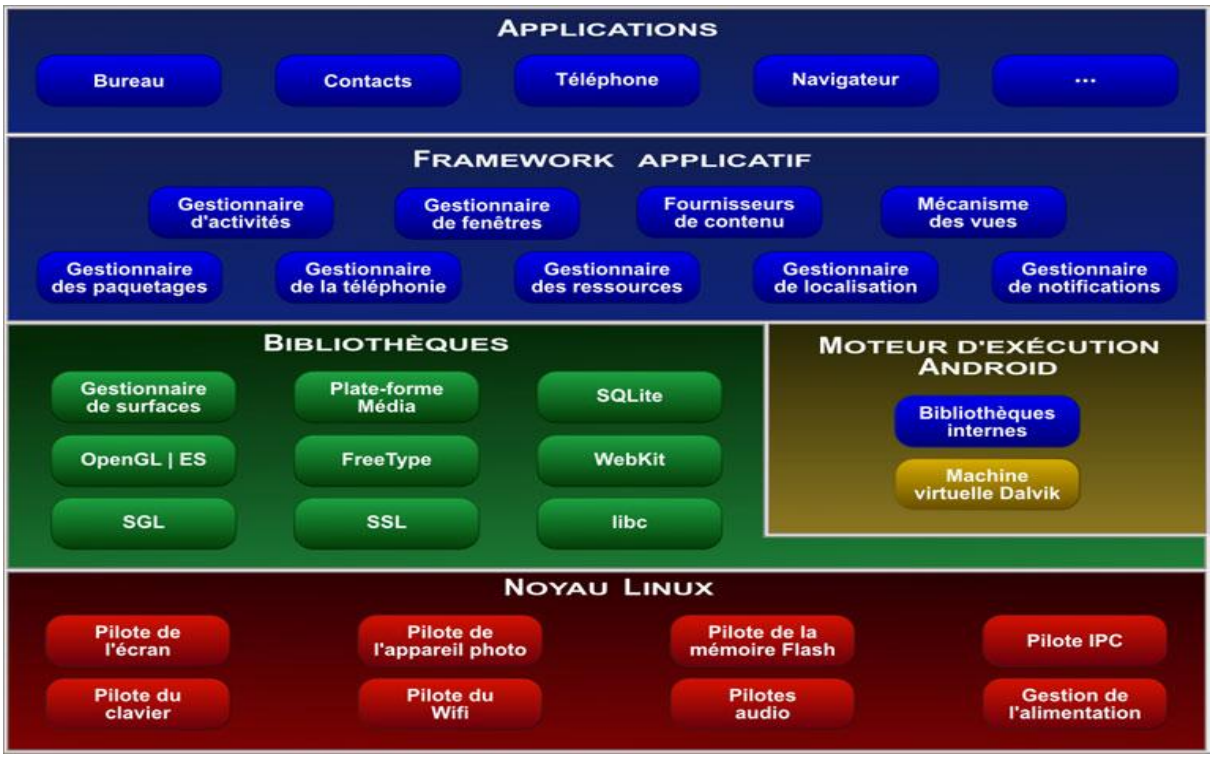

Fig.2.6. Architecture d'Android.

3.2.1. Noyau Linux :

Au plus près du matériel, nous trouvons le système d'exploitation. Il s'agit d'un noyau Linux Kernel2.6 standard, auquel Google a apporté quelques améliorations, notamment dans le domaine de la gestion de l'énergie (appareil mobile oblige) et de la communication interprocessus [**14, 23, 24**]. Précisons que pour l'instant, Android tourne sur des processeurs embarqués de type ARM, qui sont nativement gérés par Linux.

3.2.2. Librairies et environnement d'exécution :

3.2.2.1. Librairies :

Vient ensuite la couche des bibliothèques. Google a écrit pour Android sa propre bibliothèque C, bionic, dont la faible taille est adaptée à un environnement embarqué.

Nous trouvons aussi d'autres bibliothèques, dont beaucoup seront familières aux utilisateurs de Linux: Webkit, SQLite, OpenGL, SGL, SSL, ainsi qu'une bibliothèque multimédia.

3.2.2.2. Environnement d'exécution :

Directement sur cette couche vient se greffer le runtime Android, qui comprend la machine virtuelle Java et ses bibliothèques. Cette machine virtuelle, développée spécifiquement pour Android, porte le nom de « *Dalvik virtual machine* » avant la version 5. Conçue pour fonctionner dans un environnement embarqué limité en ressources, elle utilise un format d'exécutable compressé (.dex), afin de minimiser l'empreinte mémoire.

3.2.2.2.1. Core Libraries :

Les Core libraries fournissent le langage Java, disponible pour les applications. Le langage Java fournit avec Android reprend en grande partie l'API JSE1.5. Il y a des choses qui ont été mises de coté, car cela n'avait pas de sens pour Android (comme les imprimantes, swing, etc.), et d'autres APIs spécifiques requises pour Android ont été rajoutées.

3.2.2.2.2. Dalvik :

Les applications Java, développées pour Android, doivent être compilées au format Dalvik exécutable (.dex) avec l'outil (dx). Cet outil compile les (.java) en (.class) et ensuite il convertit ces (.class) en (.dex).

3.2.3. Module de développement d'applications :

La couche « plate-forme logicielle » permet quant à elle de mutualiser au maximum les ressources entre applications Java. Elle propose également un moyen pour ces applications d'échanger des données. Nous retrouvons par exemple dans cette couche la boîte à outils graphique qui permet d'afficher des boîtes de dialogue, des boutons, des menus, etc. C'est cette couche qui est rendue disponible au développeur Java au moyen d'un ensemble d'API.

A ce niveau nous distinguons deux types de services :

3.2.3.1. Core Plateform Services

Android introduit la notion de services. Les services cœurs de la plateforme (Core Plateform Services) fournissent des services essentiels au fonctionnement de la plateforme:

- Activity Manager: gère le cycle de vie des applications et maintient une pile de navigation permettant d'aller d'une application à une autre.
- Package Manager: utilisé, par l'Activity Manager, pour charger les informations provenant des fichiers (.apk) (android package file).
- Window Manager: il gère les fenêtres des applications (quelle fenêtre doit être affichée devant une autre à l'écran).
- Resource Manager: gère tous ce qui n'est pas du code, toutes les ressources (images, fichier audio, etc.).
- Content Provider: gère le partage de données entre applications.
- View System: fournit tous les composants graphiques: listes, grilles, boutons, etc.

3.2.3.2. Hardware Service :

Les services matériels (Hardware Services) fournissent un accès vers les API matérielles :

- Telephony Service: permet d'accéder aux interfaces téléphoniques (GSM, 3G, etc.).
- Location Service: permet d'accéder au GPS.
- Bluetooth Service: permet d'accéder à l'interface Bluetooth (Annexe II).
- Wi-Fi Service: permet d'accéder à l'interface Wi-Fi.
- USB Service: permet d'accéder aux interfaces USB.
- Sensor Service: permet d'accéder aux capteurs (capteur de luminosité, de proximité, d'accélération, etc.).

3.2.4. Applications :

La dernière couche, la seule finalement dont l'utilisateur aura à se préoccuper, est celle des applications. Elles sont sous forme de paquets .apk, qui permettent une installation et une désinstallation facile. Certaines sont fournies par l'équipe Android : navigateur web, gestionnaire de contacts, agenda… etc.

3.3. Environnement de développement Android

3.3.1. SDK Android et Android Studio

3.3.1.1. SDK Android

Le SDK est un ensemble d'outils qui permet aux développeurs et aux entreprises de créer des applications.

Il contient :

- les librairies Java pour créer des logiciels.
- les outils de mise en boîte des logiciels.
- un émulateur de tablettes pour tester les applications AVD.
- un outil de communication avec les vraies tablettes ADB.

Android Studio offre :

- un éditeur de sources.
- des outils de compilation et de lancement d'AVD.

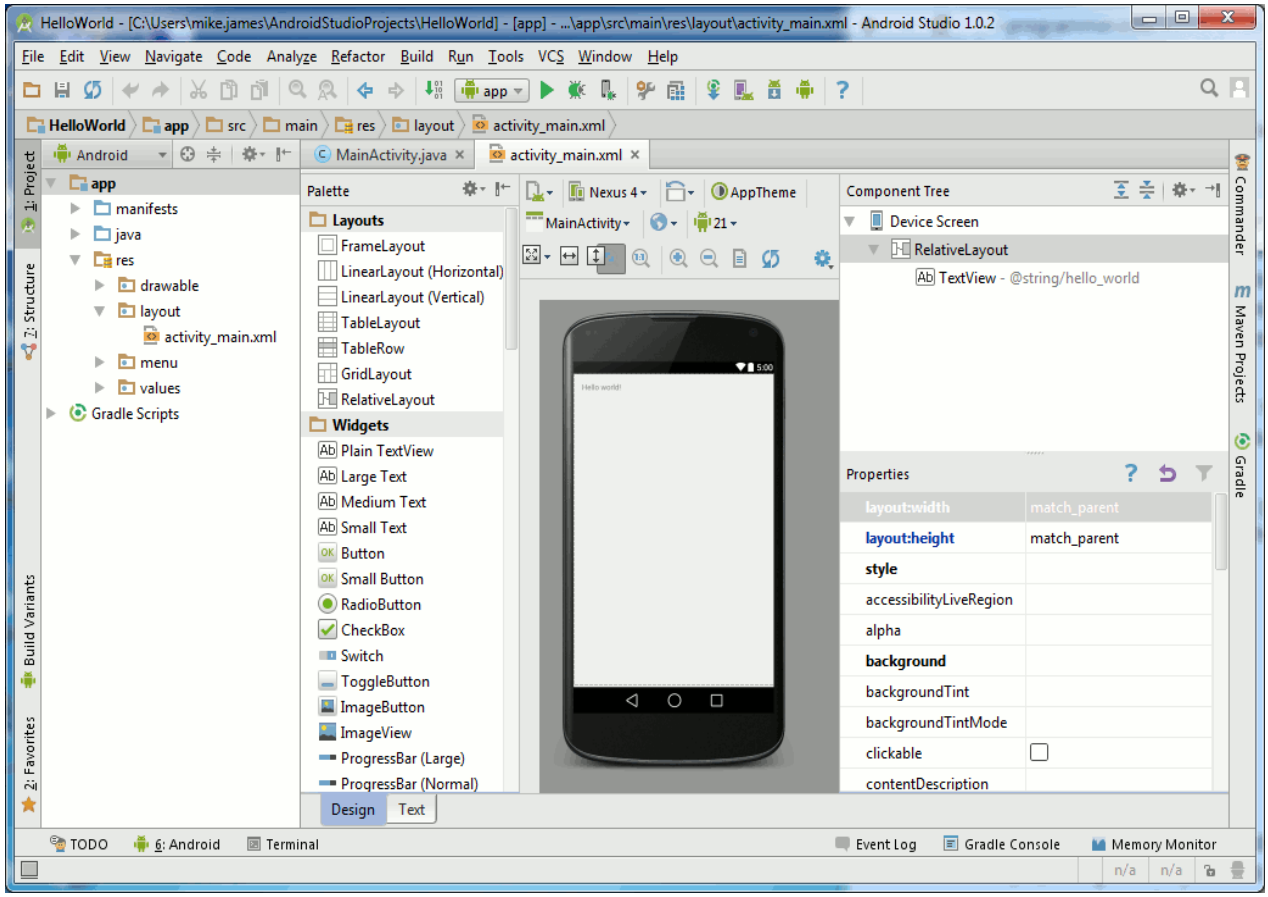

Fig.2.7. Android Studio.

3.3.1.2. SDK Manager

Le SDK est livré avec un gestionnaire. C'est une application qui permet de choisir les composants à installer. Le gestionnaire permet de choisir quelles versions d'Android installer parmi une liste des APIs qui représentent un ensemble de classes regroupant des fonctions mises à disposition des développeurs. Ces fonctions ou méthodes peuvent être regroupées dans des bibliothèques logicielles ou des services. Le plus souvent, elles effectuent des traitements de bas niveau et proposent au développeur une interface de plus haut niveau pour qu'il puisse accéder à des fonctionnalités plus facilement et surtout plus immédiatement. Par exemple, la plupart des systèmes proposent une API graphique permettant d'afficher des éléments graphiques à l'écran (fenêtres, boutons, etc.) sans avoir à gérer le périphérique dans son intégralité et ce, pixel par pixel.

Comme :

- Android 6 (API 23).
- Android 5.1.1 (API 22).
- Android 5.0.1 (API 21).
- Android 4.4W.2 (API 20).
- . . ..
- Android 1.5 (API 3).

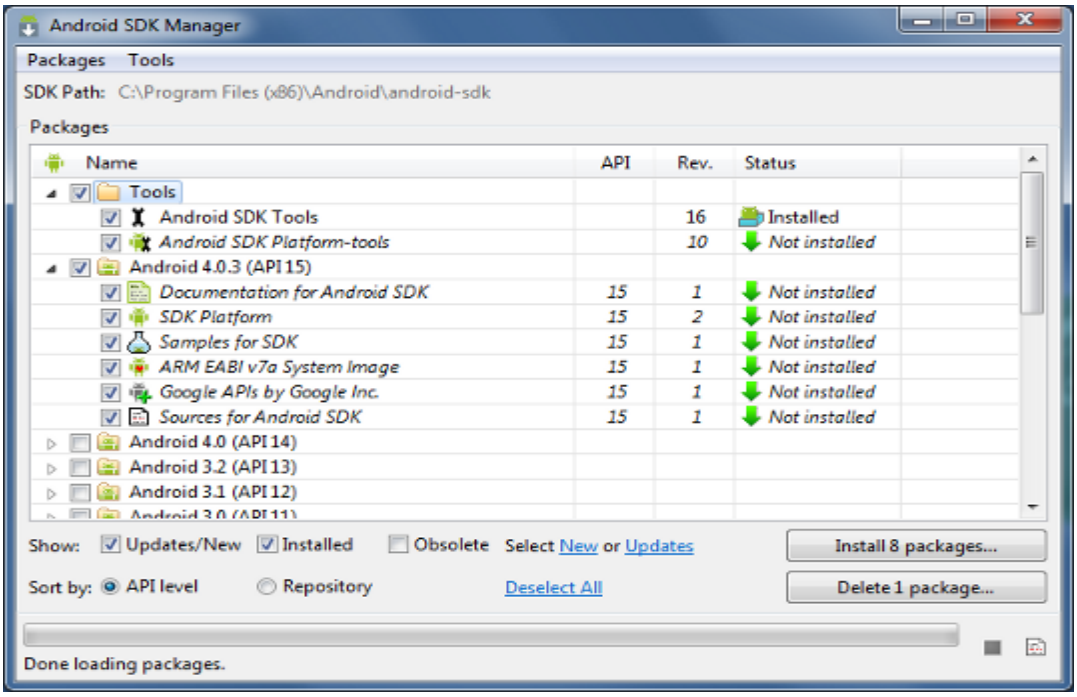

Fig.2.8. Android SDK Manager.

3.3.1.3. Dossiers du SDK

Le gestionnaire télécharge environ 800Mo de fichiers :

- SDK Tools : indispensable, contient le gestionnaire.
- SDK Platform-tools : indispensable, contient ADB.
- SDK Platform : indispensable, contient les librairies.
- System images : pour créer des AVD.
- Android Support : divers outils pour créer des applications.
- Exemples et sources.

3.3.2. AVD

Comme son nom l'indique, un AVD (*Android Virtual Device*), est un terminal virtuel par opposition aux vrais terminaux Android comme le G1 ou le Magic de HTC. Les AVD sont utilisés par l'émulateur fourni avec le SDK et nous permettent de tester nos programmes avant de les déployer sur les véritables terminaux. Nous devons indiquer à l'émulateur un terminal virtuel afin qu'il puisse prétendre qu'il est bien le terminal décrit par cet AVD.

3.3.3. Structure d'une application

Une application sous Android est constituée des éléments suivants :

- Activité (android.app.Activity):Programme qui gère une interface graphique.
- Service (android.app.Service):Programme qui fonctionne en tâche de fond sans interface.
- Fournisseur de contenu (android.content.ContentProvider):Partage d'informations entre applications.
- Ecouteur d'intentions diffusées (android.content.BroadcastReceiver):Permet à une application de récupérer des informations générales (réception d'un SMS, batterie faible, …).
- Eléments d'interaction :
    - Intention (android.content.Intent) : permet à une application d'indiquer ce qu'elle sait faire ou de chercher un savoir-faire.
    - Filtre d'intentions (<intent-filter>) : permet de choisir la meilleure application pour assurer un savoir-faire.

La réalisation d'une première application sur « *Android Studio* » sera illustrée dans le fichier (Annexe III).

3.3.4. Cycle de vie d'une activité

La brique de base de l'interface utilisateur s'appelle « *activity* » (activité). Nous pouvons la considérer comme l'équivalent Android de la fenêtre ou de la boîte de dialogue d'une application classique [**15**].

La méthode OnCreate () est appelée à la création de notre activité. Elle sert à initialiser l'activité ainsi que toutes les données nécessaires à cette dernière. Quand la méthode OnCreate () est appelée, on lui passe un Bundle en argument. Ce Bundle contient l'état de sauvegarde enregistré lors de la dernière exécution de l'activité.

La méthode OnStart () signifie le début d'exécution de l'activité (début du passage au premier plan). Si l'activité ne peut pas aller en avant plan, pour une quelconque raison, elle sera transférée à OnStop ().

La méthode OnResume () est appelée lorsque l'activité commencera à interagir avec l'utilisateur juste après avoir été dans un état de pause.

La méthode OnPause () est appelée au passage d'une autre activité en premier plan. L'intérêt d'un tel appel est de sauvegarder l'état de l'activité et les différents traitements effectués par l'utilisateur. A ce stade, notre activité n'a plus accès à l'écran, nous devons arrêter de faire toute action en rapport avec l'interaction utilisateur (désabonner les listeners).

La méthode OnStop () est appelée quand l'activité n'est plus visible quelque soit la raison. Dans cette méthode nous devons arrêter tous les traitements et services exécutés par notre application.

La méthode OnDestroy () est appelée quand notre application est totalement fermée (Processus terminé). Les données non sauvegardées sont perdues.

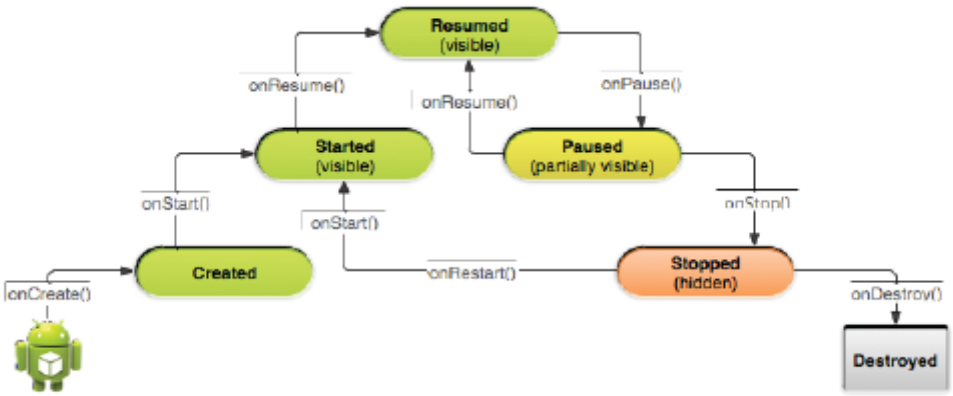

Fig.2.9. Cycle de vie d'une activité.

3.3.5. Les capteurs

Une bonne partie du succès des terminaux modernes est la gestion du mouvement. Cela ne serait pas possible sans l'extraordinaire démocratisation des capteurs embarqués. Désormais, votre appareil détecte le mouvement et peut se situer dans l'espace [**16]**.

La liste des différents types de capteurs est la suivante :

- TYPE_ACCELEROMETER: permet d'évaluer les mouvements du terminal ou tout simplement la gravité.
- TYPE_GYROSCOPE: permet de connaître la position angulaire du terminal en fonction de trois axes x, y, z.
- TYPE_LIGHT: le capteur de lumière permet d'évaluer l'exposition lumineuse subie par le terminal.
- TYPE_MAGNETIC_FIELD: permet de détecter les modifications des champs magnétiques environnants.
- TYPE_ORIENTATION: permet de déterminer la position du terminal en fonction de trois axes x, y, z.
- TYPE_PRESSURE: indique la pression atmosphérique.
- TYPE_PROXIMITY: indique la distance du terminal par rapport à un objet.
- TYPE_TEMPERATURE : indique la température à proximité du capteur.

Android possède donc un nombre important de capteurs permettant de prendre en compte l'environnement de l'utilisateur: orientation et localisation, température, champs magnétique,…etc. Tous ces capteurs peuvent être utilisés par nos applications.

Chaque appareil pouvant posséder ou non certains capteurs, c'est pour cela il faut savoir avant tout détecter si les capteurs sont disponibles et en état de marche (réseau accessible ? localisation GPS activée ? …etc.) [**17**].

3.4. Avantages de la plateforme « *Android* »

Depuis sa création, la popularité d'Android a toujours été croissante. Désormais, nous le retrouvons non seulement dans les tablettes et smartphones, mais aussi dans les téléviseurs, les consoles de jeux, les appareils photos, etc. Cette popularité est due à sa puissance remarquable et aux avantages offerts par cette plateforme de développement d'applications mobiles [**18]**.

Parmi ces avantages, nous pouvons citer les suivants :

- ***Open source***

Le contrat de licence pour Android respecte les principes de l' « *open source* », c'est-à-dire que nous pouvons à tout moment télécharger les sources et les modifier selon nos goûts. Il utilise des bibliothèques « *open source* »puissantes, comme par exemple « *SQLite* » pour les bases de données et « *OpenGL* »pour la gestion d'images 2D et 3D.

- ***Gratuit (ou presque)***

Android est gratuit, autant pour les développeurs que pour les constructeurs. Le « *Play Store* » offre une variété d'applications, gratuites et payantes.

- ***Facile à développer***

Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès. De manière un peu caricaturale, nous pouvons dire que nous pouvons envoyer un SMS en seulement deux lignes de code (concrètement, il y a un peu d'enrobage autour de ce code, mais pas tellement).

- ***Facile à vendre***

Le « *Play Store* » (anciennement *Android Market*) est une plateforme immense et très visitée ; c'est donc une mine d'opportunités pour quiconque possède une idée originale ou utile.

- ***Flexible***

Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes. Les smartphones, les tablettes, la présence ou l'absence de clavier ou de « *trackball* », différents processeurs...etc. Nous trouvons même des fours à micro-ondes qui fonctionnent à l'aide d'Android !

Non seulement c'est une immense chance d'avoir autant d'opportunités, mais en plus Android est construit de manière à faciliter le développement et la distribution en fonction des composants en présence dans le terminal (si l'application nécessite d'utiliser le Bluetooth, seuls les terminaux équipés de Bluetooth pourront la voir sur le Play Store).

- ***Ingénieux***

L'architecture d'Android est inspirée par les applications composites, et encourage par ailleurs leur développement. Ces applications se trouvent essentiellement sur internet et donnent la possibilité de combiner plusieurs composants totalement différents pour obtenir un résultat surpuissant. Par exemple, l'utilisation de l'appareil photo avec le GPS, pour poster les coordonnées GPS des photos prises.

4. MIT App Inventor 2

« *App Inventor* » pour Android est une application développée par Google. Elle est actuellement entretenue par le Massachusetts Institute of Technology (MIT). Elle simplifie le développement des applications sous Android et le rend accessible même pour les novices et ceux qui ne sont plus familiers avec les langages de programmation. Elle est basée sur une interface graphique similaire à « *Scratch* » et à celle de « *StarLogoTNG* ». Grâce à son interface entièrement graphique et à l'absence totale de ligne de code, elle est particulièrement adaptée à l'initiation des enfants à la programmation, et ce dès l'école primaire.

Google a publié l'application le 15 décembre 2010 et a mis fin à son activité le 31 décembre 2011. Depuis, c'est le centre d'études mobiles au MIT qui gère le support technique de cette application sous le nouveau nom « *MIT App Inventor* ». Le 4 Mars, 2012 , l'Institut de technologie du Massachusetts (MIT) à nouveau mis le projet sur Internet. En Décembre 2013, « *MIT AI 2* » montre, une nouvelle version de l'App Inventor, c'est la version que nous allons utiliser lors de la réalisation de notre application [**20]**.

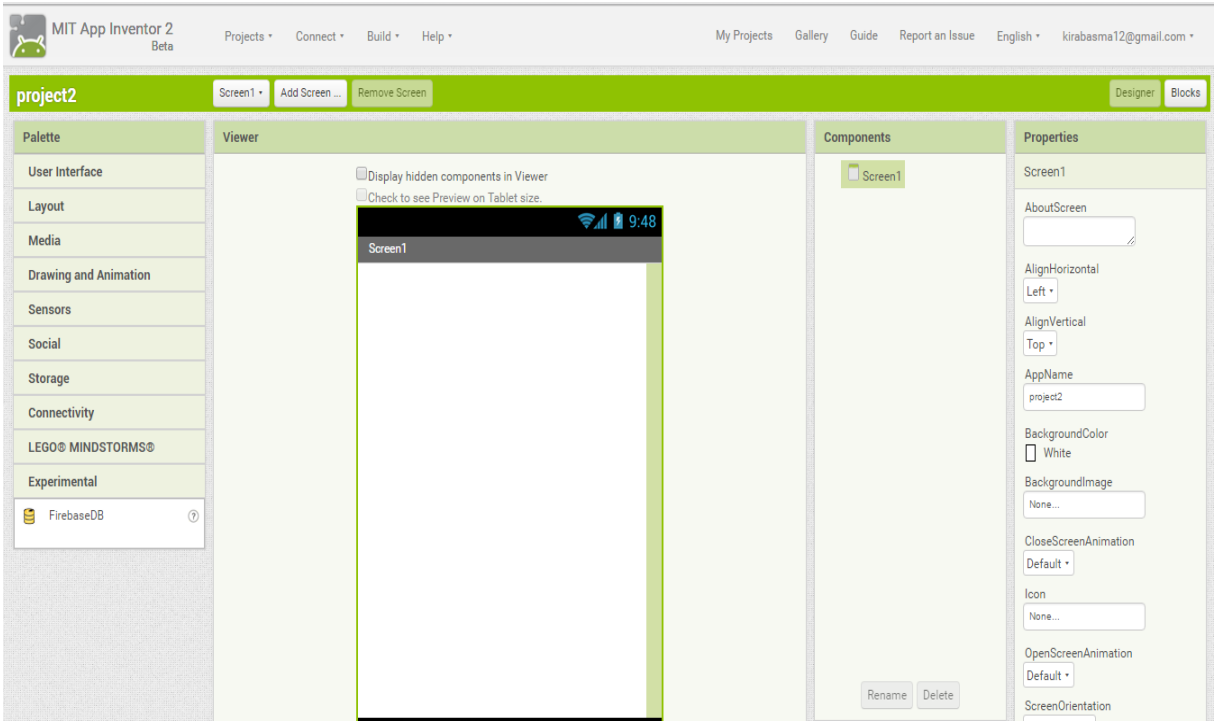

Fig.2.10. MIT App Inventor.

L'utilisation du MIT App Inventor nécessite en premier lieu un compte Gmail, pour pouvoir accéder à tout moment à la liste de nos projets après authentification.

4.1. Partie « *Designer* »

C'est la partie de création de l'interface et les différents écrans dans une application mobile. Elle contient tous les éléments possibles qui peuvent intervenir lors de l'exécution d'un projet ou d'une activité. C'est avec cette partie du MIT App Inventor, que nous pouvons donner l'image finale de notre application désirée.

Elle contient les composants visibles dans l'écran comme ; les boutons, les étiquettes, les tableaux …etc. Et aussi les composants invisibles comme ; le Bluetooth Server, l'horloge (Clock), camera, …etc. Tous ces composants sont placés sur le coté gauche de la page de l'application MIT App Inventor, la partie droite sera alors dédiée à la mise en forme et la nomination des différents composants. Le milieu de la page représente la scène où nous pouvons seulement visualiser l'interface de notre application, et non pas la tester.

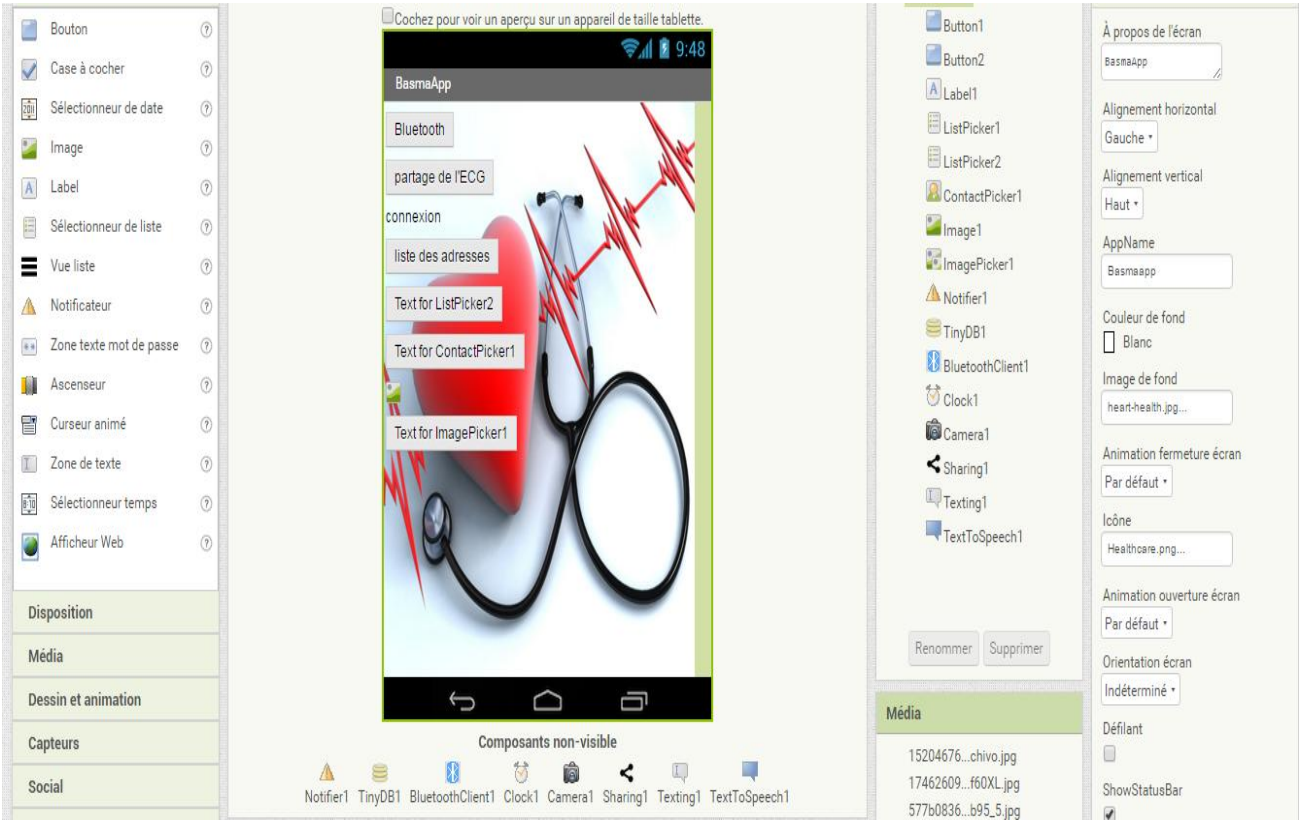

Fig.2.11. Partie Designer.

4.1.1. Les composants de la partie Designer

- L'interface utilisateur :

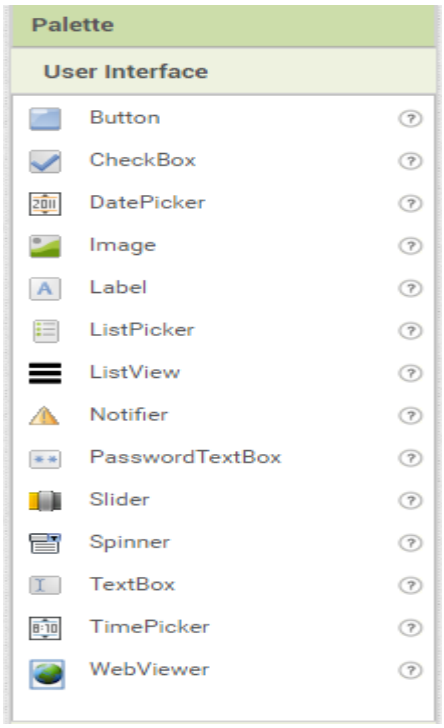

Fig.2.12. Interface Utilisateur.

- Layout (disposition) :

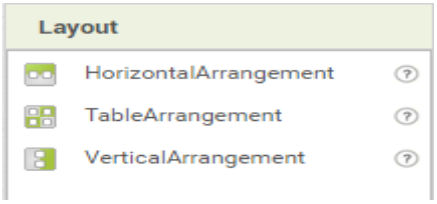

Fig.2.13. Disposition.

- Media :

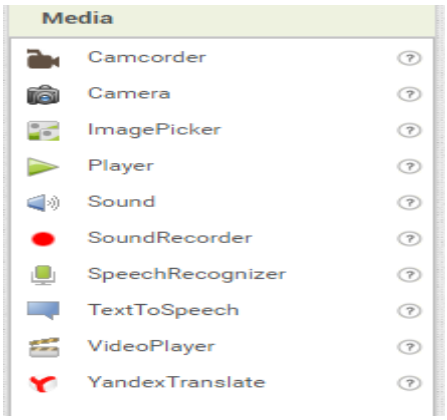

Fig.2.14. Media.

- Dessin et animation :

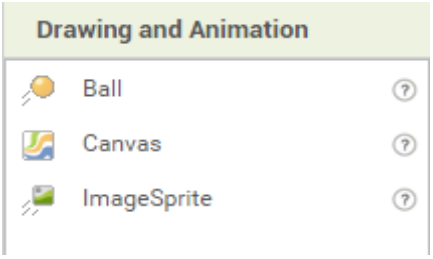

Fig.2.15. Dessin et animation.

- Sensors (Capteurs) :

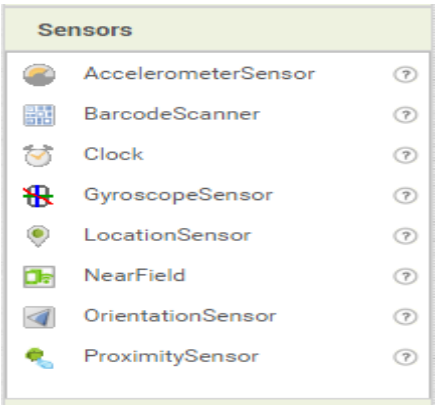

Fig.2.16. Capteur.

- Social (réseaux et sites de communication) :

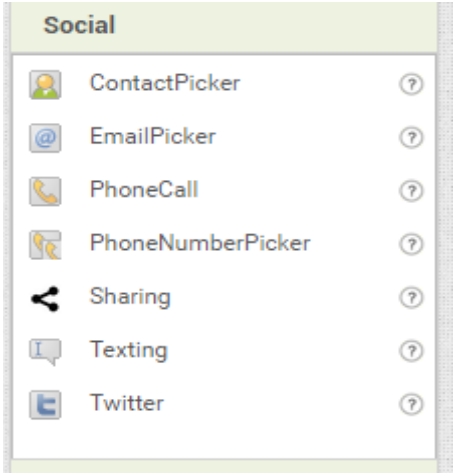

Fig.2.17. Social.

- Storage (stockage) :

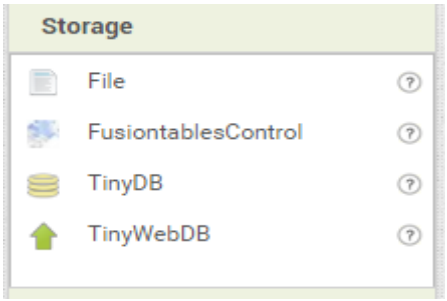

Fig.2.18. Stockage.

- Connectivité :

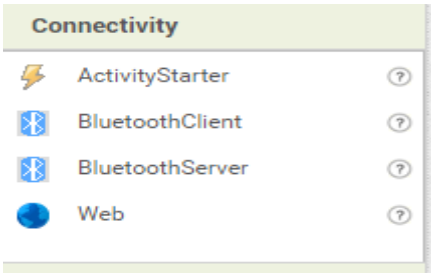

Fig.2.19. Connectivité.

4.2. Partie « *Blocs* »

Cette partie est celle responsable de la programmation des composants choisis dans la partie précédente. Elle a le même concept d'algorithme ; elle utilise des fonctions et des boucles écrites en langage humain, sans avoir besoin d'utiliser ou bien de retenir une structure ou un code de programmation précis. Elle donne la possibilité de réaliser n'importe quelle application, et ouvre la porte à notre imagination sans limite. La difficulté réside juste dans l'enchainement des idées pour obtenir un résultat fiable et assuré.

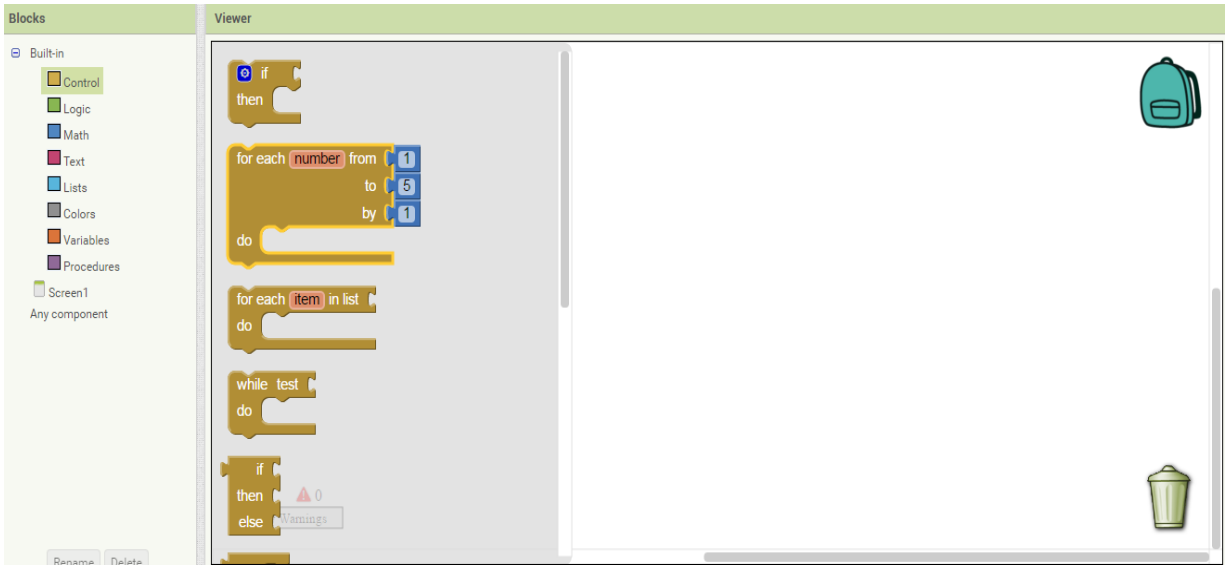

Fig.2.20. Partie Blocs.

4.2.1. Les blocs :

- De contrôle comme :

Fig.2.21. Blocs de Contrôle.

- Logique :

Fig.2.22. Blocs logiques.

- Math comme :

Fig.2.23. Blocs d'opérations mathématiques.

- De texte comme:

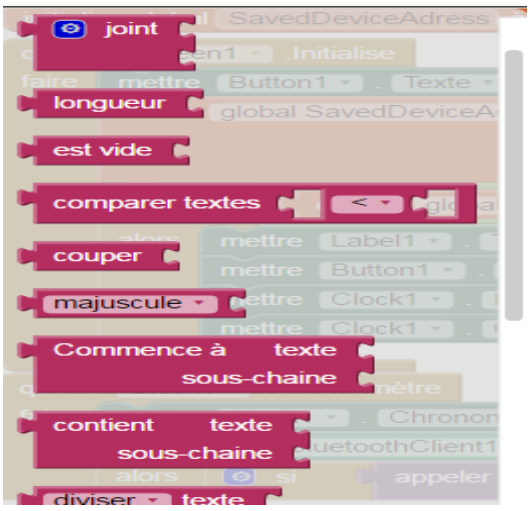

Fig.2.24. Blocs de textes.

- De listes comme :

Fig.2.25. Blocs de listes.

- De couleurs comme :

Fig.2.26. Blocs de couleurs.

- Variables :

Fig.2.27. Blocs de variables.

- Procédures :

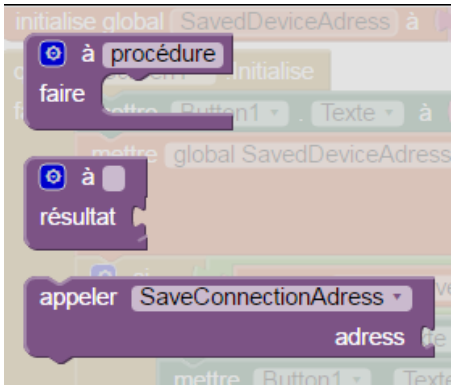

Fig.2.28. Blocs de procédures.

Il faut noter que cette liste donnée des blocs n'est pas complète, car chaque composant invisible utilisé (dans la partie Designer) possède ses propres blocs ; par exemple pour le composant Bluetooth Client on a comme blocs :

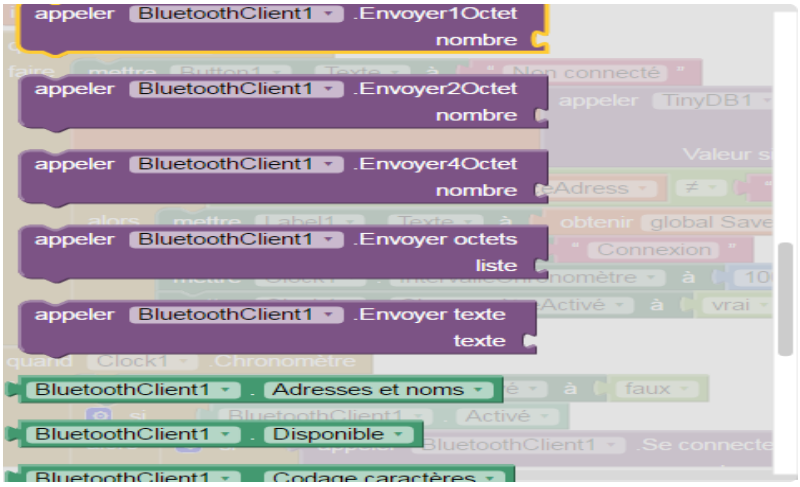

Fig.2.29. Blocs de Bluetooth Client.

4.3. Exécution d'une application dans MIT App Inventor

L'exécution de n'importe quelle application peut se faire avec trois manières possibles :

- Soit en utilisant notre terminal mobile réel « *Al Companion* » ; et ceci n'est possible que par l'installation de l'application « MIT App Inventor » depuis Google Play Store, et puis scanner le QR Code à l'aide de notre téléphone portable.

Fig.2.30. Scan du code QR.

- Soit en utilisant l'émulateur (comme le AVD), après avoir installé l'outil software du MIT App Inventor ; nommé « *aistarter* » depuis l'internet.

Fig.2.31. Le software aistarter.

Après l'exécution de l'« aistarter » et lors du choix de l'émulateur, une fenêtre va apparaitre:

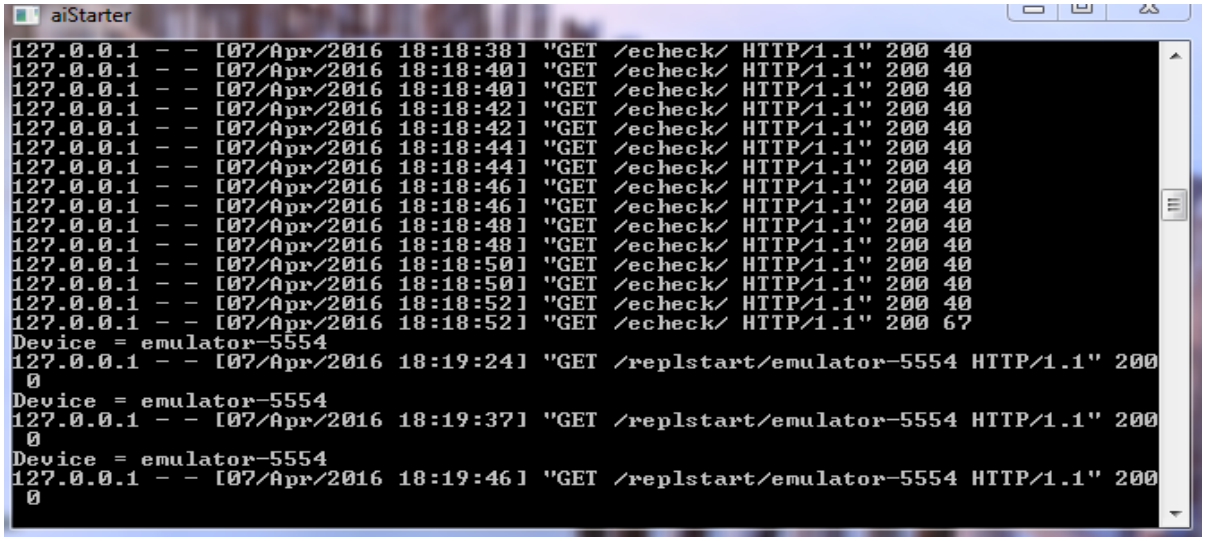

Fig.2.32. L'aistarter lors de l'exécution.

L'émulateur passe par les étapes suivantes dans un intervalle de 1 ou 2 minutes :

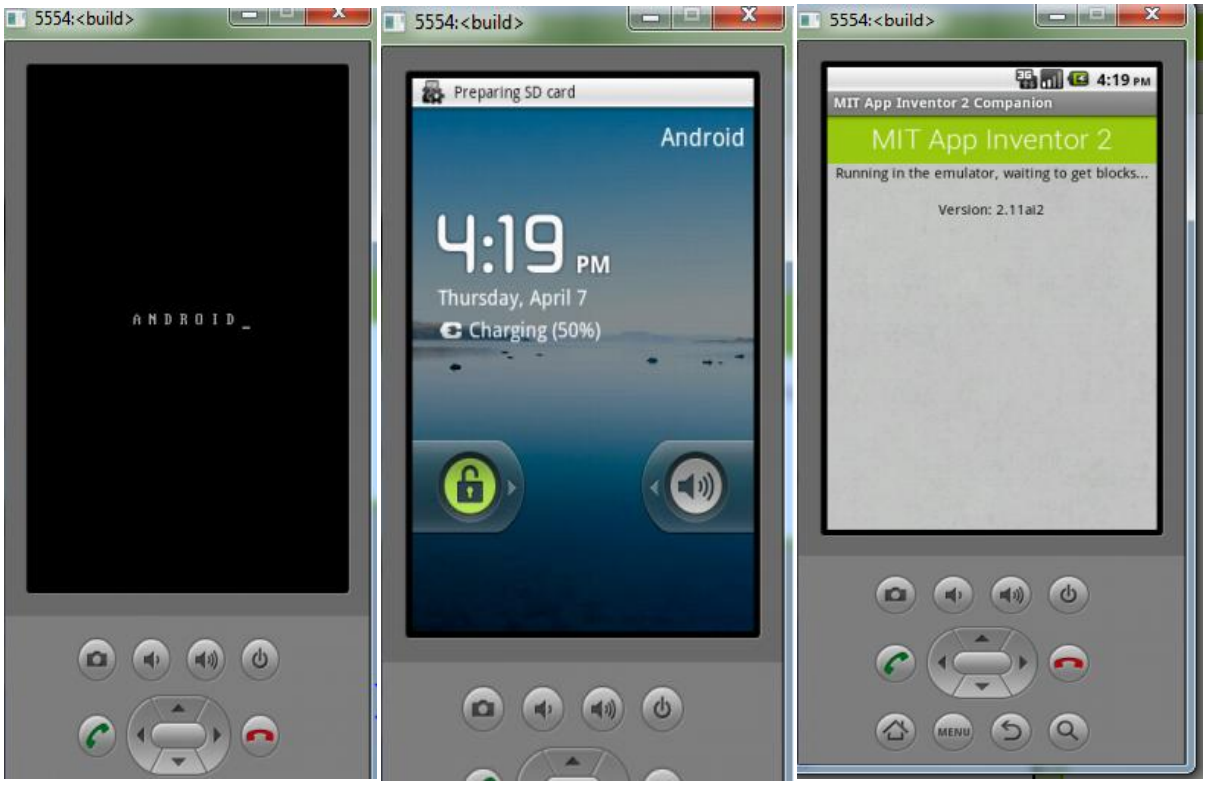

Fig.2.33. L'émulateur et ses différentes étapes lors de l'exécution.

- Soit par l'utilisation d'un câble USB :

Pour cette méthode ; nous devons installer le software « *aistarter* » et le « MIT Companion » par le scan du QR Code. Après l'exécution du « *aistarter* », nous pouvons visualiser et tester notre application facilement sur notre terminal mobile.

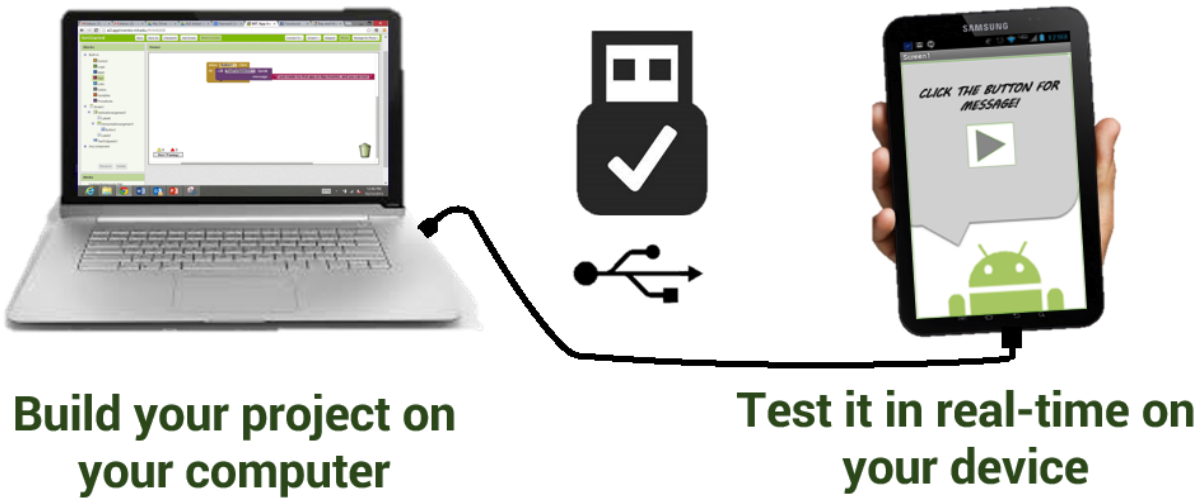

Fig.2.34. Visualisation via téléphone à l'aide d'un USB.

5. Conclusion :

Ce deuxième chapitre détaille le développement des applications sous l'environnement Android; passant par son architecture, ses différents composants softwares et les principales caractéristiques d'une application mobile sous Android.

Nous avons également définit l'outil principal utilisé lors dans la réalisation de notre application ; le « *MIT App Inventor* », citant ainsi ses éléments, ses parties et composants. Vers la fin de ce chapitre, nous avons donné un aperçu sur les méthodes d'exécution d'un projet ou d'une application développée sous MIT App Inventor.

Lors du dernier chapitre, nous allons détailler la phase principale dans cet œuvre, qui est la mise en application de notre solution proposée « *BasmaApp* ».

Chapitre 3:
Réalisation et mise en application de la solution proposée

1. Introduction

Ce dernier chapitre sera dédié au développement de notre application mobile de télésurveillance d'un patient à domicile, ainsi que le transfert de ses données médicales directement et en temps réel. Nous allons donner une description approfondie de toutes les étapes indispensables à la création et à la réalisation de cette solution innovante nommée « BasmaApp ».

Nous allons présenter au début le schéma descriptif de la solution. Nous allons définir par la suite les différentes étapes et programmes incluent dans la réalisation de l'application ambulatoire, et de même les difficultés rencontrées lors de la mise en œuvre de cette solution et les nouvelles améliorations proposées et effectuées pour aboutir au résultat finale. Vers la fin, nous allons présenter l'application en étape finale.

2. Solution proposée de télésurveillance

L'envoi et la récupération de données sont des fonctions indispensables : qu'elle présente de l'information, dialogue sur les réseaux sociaux, mette à jour des bases de données distantes ou affiche de la publicité, une application doit communiquer.

Pour répondre aux exigences imposées par le patient souffrant d'une insuffisance cardiaque chronique, et dans le but d'améliorer l'état de santé des patients et la qualité des services offertes aux ces derniers ; une solution de télésurveillance parait très utile dans le carde du transfert des données médicales d'urgences dont le but major est la survie des malades.

Notre solution proposée consiste au premier plan à transférer les informations médicales de la fréquence cardiaque mesurée par un Holter ECG via Bluetooth à un téléphone mobile du malade ou bien l'un de ses proches. Ces données seront ensuite partagées en temps réel avec le mobile du médecin via le réseau téléphonique cellulaire. Une procédure adéquate proposée par le médecin à distance va être le facteur ambulatoire et innovant qui donne du nouvel sens à la pratique de la médecine et au suivie des malades souffrants d'une telle maladie chronique.

Cette solution illustrée dans le schéma suivant, se compose donc de deux parties ; la première partie consiste à surveiller le malade durant une période de 24h au minimum, où les mesures de fréquences cardiaques seront établies toutes les demi-heures. La deuxième partie représente la télésurveillance proprement dite. Le médecin ici peut suivre son malade à distance et en temps réel, ainsi interagir dans un cas d'urgence et qui nécessite une intervention rapide.

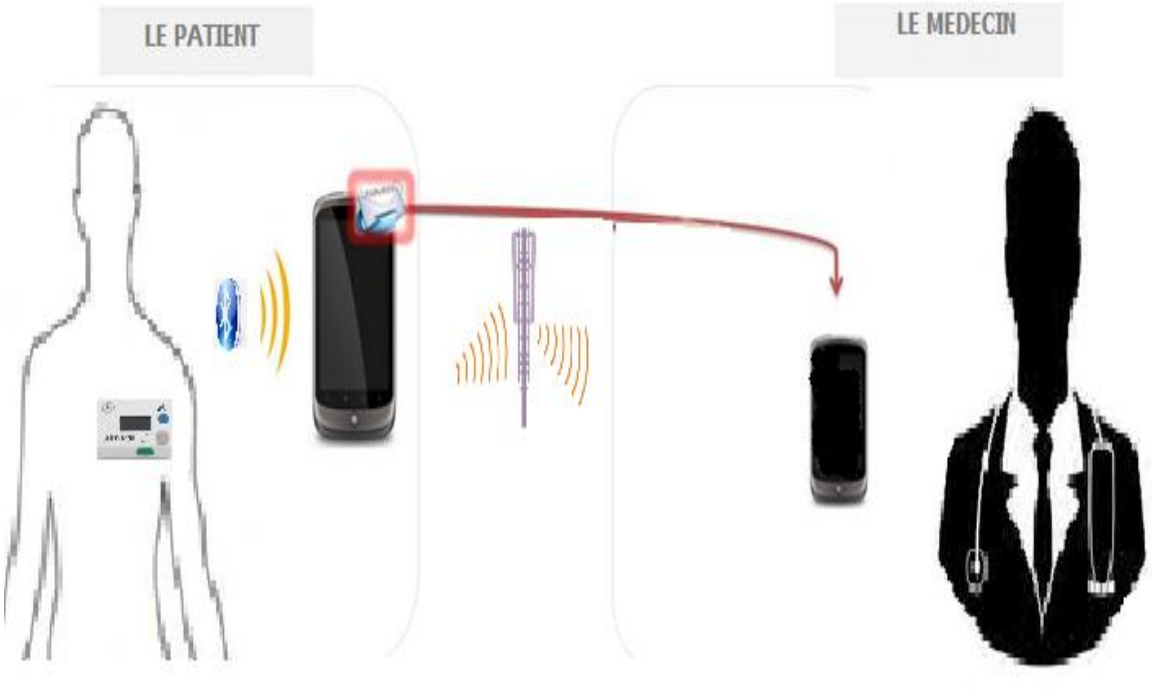

Fig.3.1. Schéma fonctionnel de la solution de télésurveillance proposée.

3. Réalisation de la solution proposée

3.1. Partie Bluetooth

La plate-forme Android supporte le Bluetooth afin de permettre à des appareils d'échanger des données entre elles. Pour pouvoir profiter de ce système de connexion, Android apporte un ensemble d'API. Ces API supportent la connexion à un appareil dans une configuration point à point (l'appareil est connecté à un seul autre appareil) ou alors en multipoints (l'appareil peut être connecté à plusieurs appareils simultanément). Pour établir une connexion entre deux appareils, nous devons implémenter chaque côté de notre logique de communication : le côté serveur et le côté client.

Dans notre cas nous allons utiliser les composants « *Bluetooth Server* » et « *Bluetooth Client* » pour établir la liaison et faire échanger les données entre l'appareil utilisé «*Holter ECG* » et le mobile du patient (ou bien un de ses proches).

Fig.3.2. Une partie du premier code pour la liaison Bluetooth.

Le problème majeur de cette solution réside dans la possibilité de connexion à n'importe quelle adresse MAC de l'adaptateur Bluetooth de l'appareil « Holter ECG » utilisé.

Alors cette solution ne convient pas à nos besoins ; Pour cela nous devons utiliser un code qui permet la découverte des adresses et puis la connexion à l'adresse choisie.

La nouvelle solution sera comme suit :

- Lorsque l'application démarre, la fonction « *Screen1.Initialize* » est appelée. Il récupère l'adresse Bluetooth enregistrée de la « *TinyDB*» et la copie dans la variable globale « *savedDeviceAddress* », puis, il démarre l'horloge. Si il n'y a pas une adresse enregistrée la variable a la valeur "" et l'horloge ne démarre pas [**25]**.

Fig.3.3. Recherche des adresses MAC par le mobile.

- Nous utilisons un « *timer* ». Si les blocs de connexion sont tous placés dans la fonction « *Screen1.Initialize* », le processus de connexion fonctionne mais l'écran n'est pas correctement actualisé et le texte du bouton « *Connexion* » n'est pas affiché.
- Lorsque le « *timer* » se déclenche, la fonction « *Clock1.Timer* » est appelée. C'est ici que l'application essaie de se connecter à l'adresse enregistrée de Bluetooth. Si une connexion est réussie le texte « *BUTTON1*» est remplacée par « Connecté ». Si la connexion n'est pas réussie, un message d'erreur s'affiche. La fonction vérifie également que le Bluetooth est activé avant d'essayer d'établir une connexion [**26,27]**.

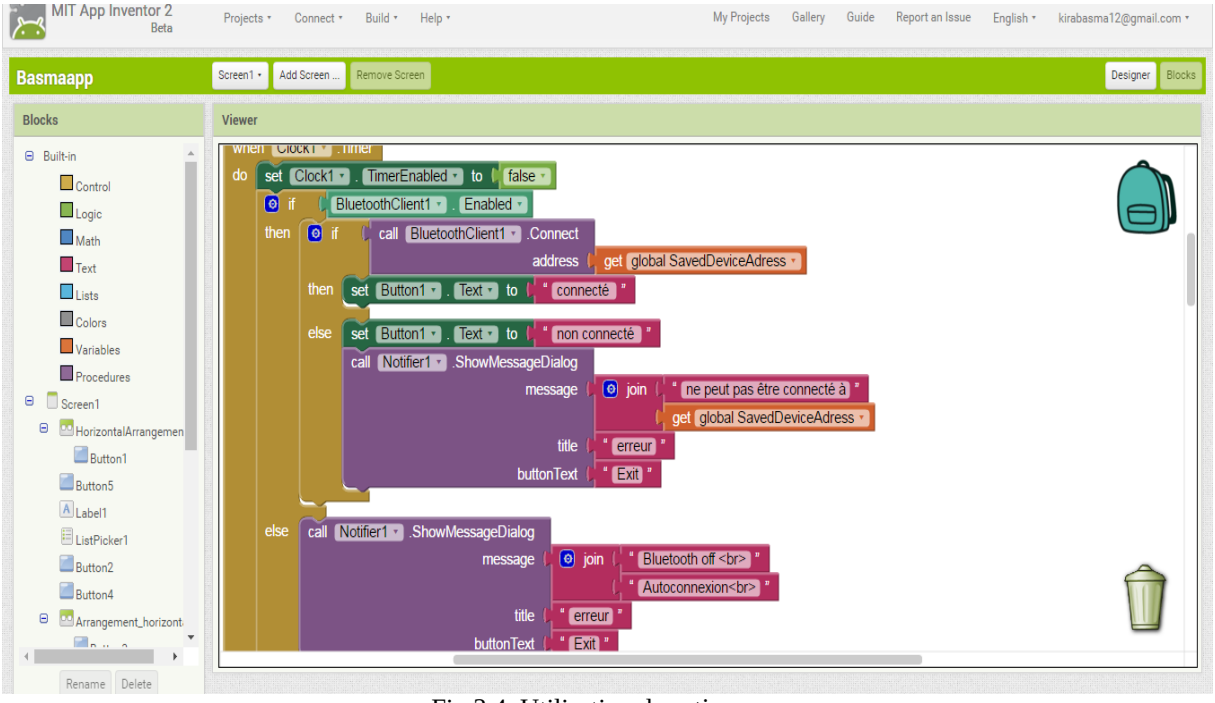

Fig.3.4. Utilisation du « *timer* ».

- Le « *BUTTON1* » permet pour les nouvelles connexions, d'activer le sélecteur de la liste et permet à l'utilisateur de sélectionner un autre périphérique. Lorsqu'une nouvelle connexion est effectuée, l'adresse MAC du module « *ListPicker1* » nouvellement connecté est enregistrée dans le « *TinyDB* » [**26,27]**.

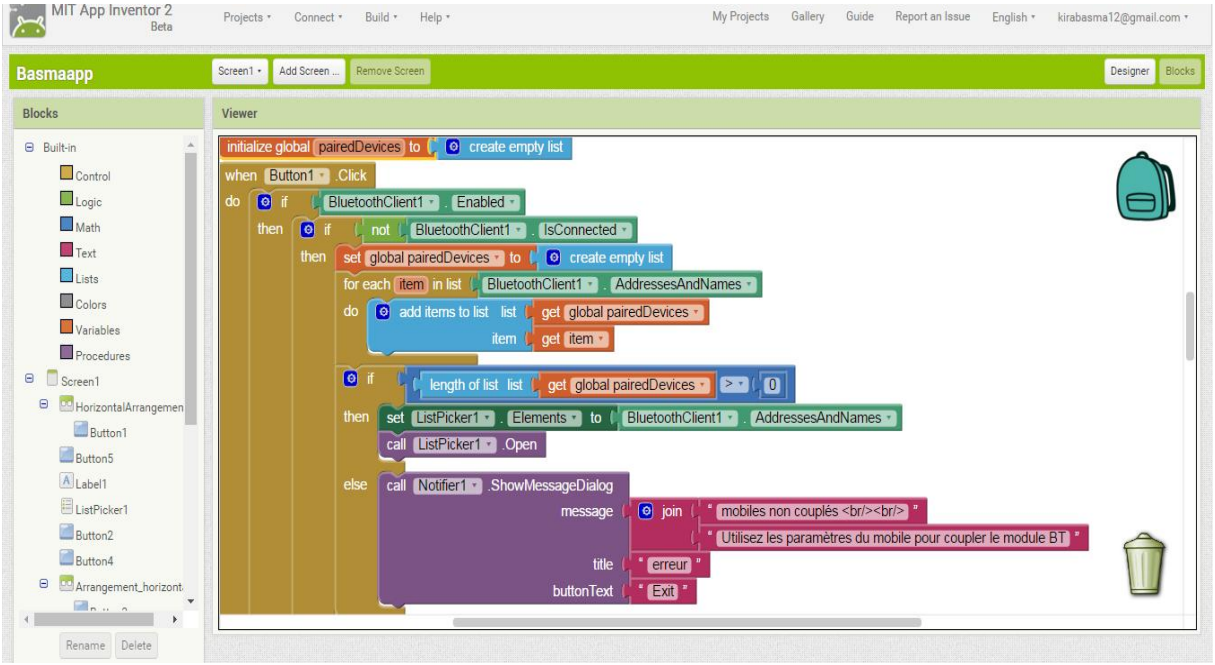

Fig.3.5. Une partie du code pour la sélection d'une nouvelle connexion.

Après l'établissement d'une liaison, nous pouvons effectuer un échange de fichiers par le code illustré dans la figure suivante :

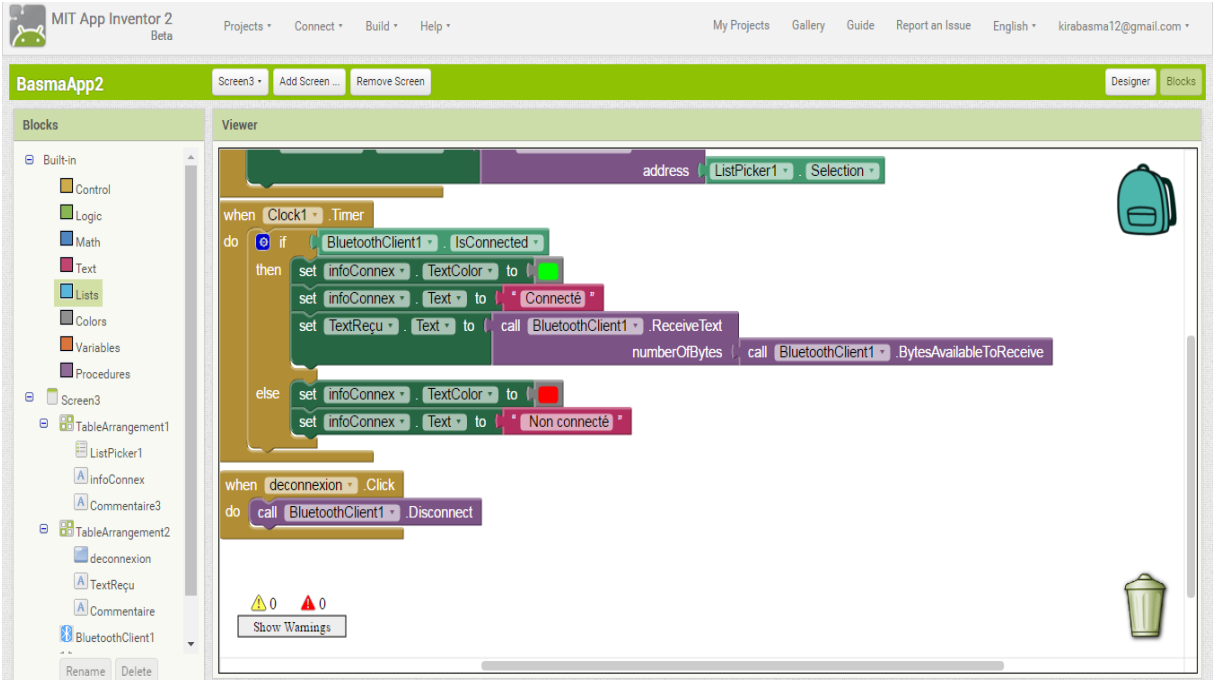

Fig.3.6. Affichage périodique des messages reçus.

3.2. Partie GSM

Bien qu'Android ne soit pas uniquement destiné aux téléphones, la téléphonie reste un composant essentiel de la plate-forme. Il y a encore beaucoup d'applications à inventer ou à perfectionner pour améliorer le quotidien de l'utilisateur, comme celles qui règlent le volume de la sonnerie ou la disponibilité pour la prise d'appels en fonction de la position géographique de l'utilisateur entre sa maison et son lieu de travail !

Notre téléphone, ou notre appareil qui intègre des fonctions de téléphonie, nous offre déjà des applications qui gèrent les appels et les SMS. Pour autant, toutes les applications Android sont égales. Il est ainsi aisé de remplacer ces applications ou d'en ajouter. Par exemple, on peut concevoir un système qui filtre les appels en fonction du numéro ou de la personne qui appelle. Nous pouvons aussi créer un répondeur automatique par SMS.

Dans notre travail ; nous avons besoin d'envoyer les données reçus de l'appareil « *Holter ECG* » au mobile distant du médecin. Ce dernier va utiliser l'option d'alarme vocale « *TextToSpeech* » pour alerter le médecin, et lui donne la possibilité de voir les fichiers reçus, en choisissant ainsi dans le cas d'urgence le message approprié pour répondre.

3.2.1. L'envoi des messages via GSM :

Une liste des contacts va être élaborée après avoir cliqué sur le « Button5 », pour permettre à l'utilisateur de sélectionner le contact désiré (le médecin). Un envoi automatique des fichiers sera établi après la sélection du contact.

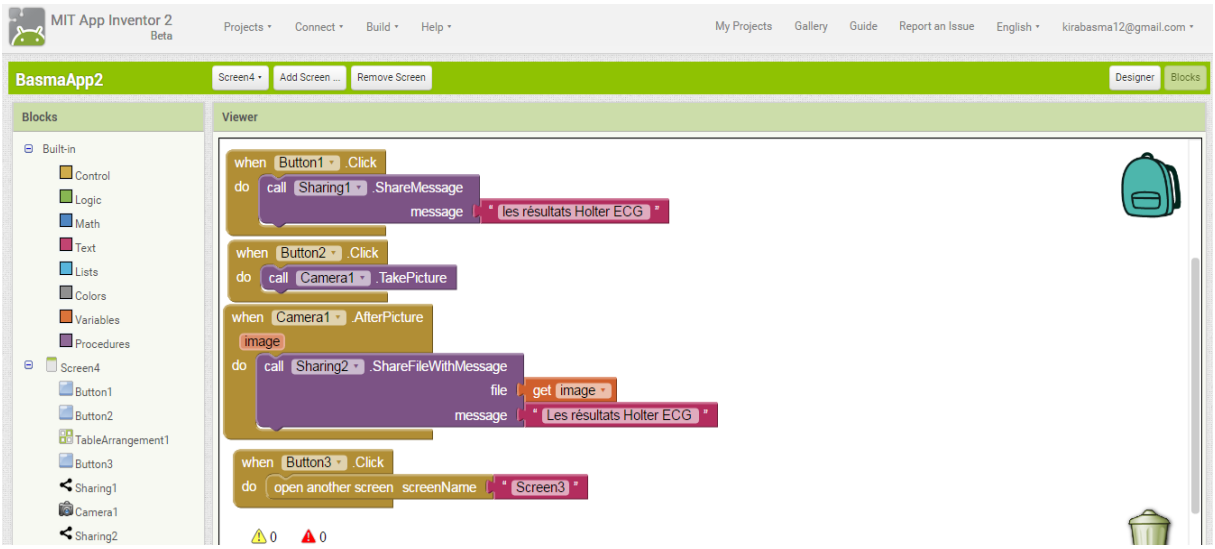

Fig.3.7. Code pour partage des fichiers via GSM.

A la réception des fichiers, le mobile va utiliser le « TextToSpeech » pour alerter le médecin.

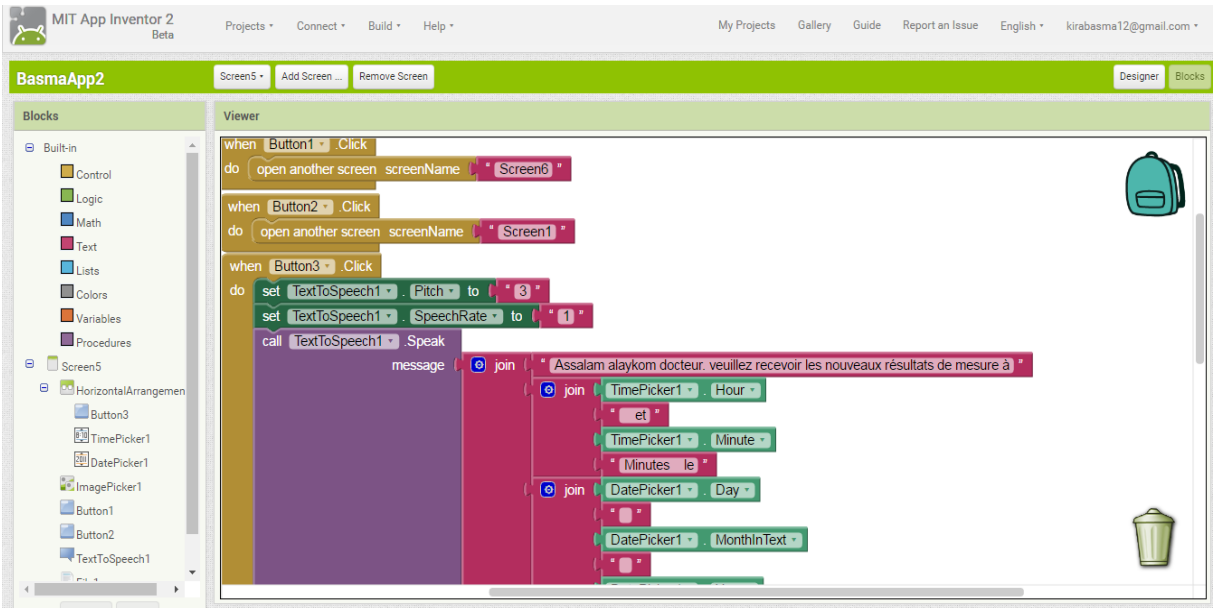

Fig.3.8. Code d'alarme vocale.

Dans le cas d'urgences, le médecin peut choisir un message et l'envoyer immédiatement au patient (ou bien un de ses proches).

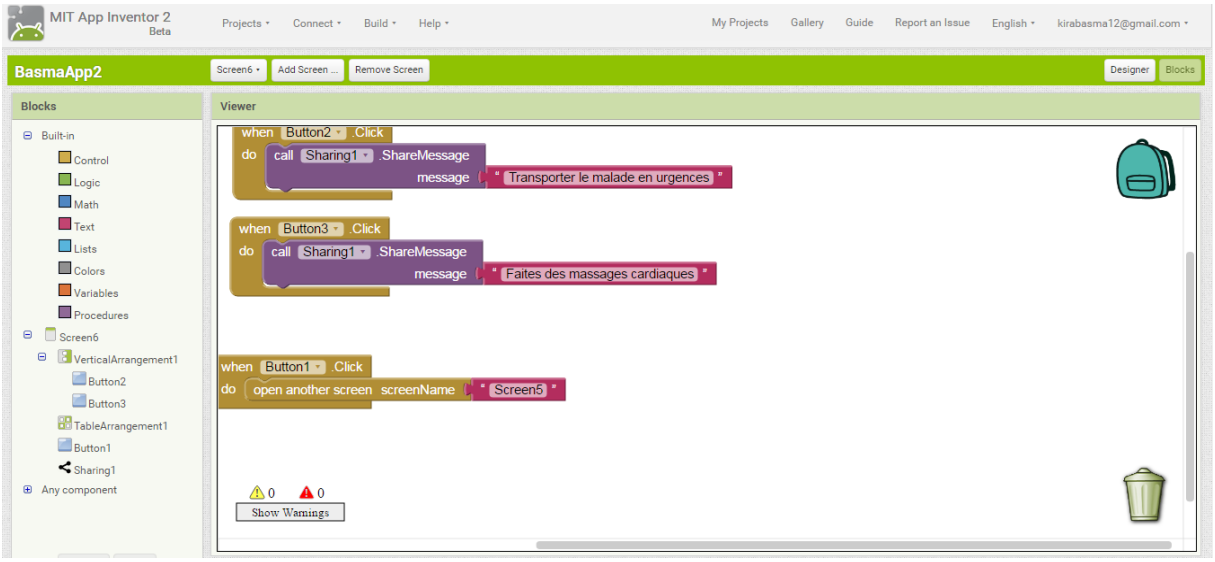

Fig.3.9. Code pour le choix du message par le médecin.

3.3. Autres améliorations

Grâce à la gestion multiple d'écrans dans MIT App Inventor, on peut facilement ajouter des pages de « *Conseils de santé* » ou bien de « *Premiers secours* » …etc. En cliquant sur un bouton nous pouvons accéder à un autre écran, soit le précédent ou bien le suivant. Le code suivant représente un exemple utilisé lors de nos améliorations apportées au projet.

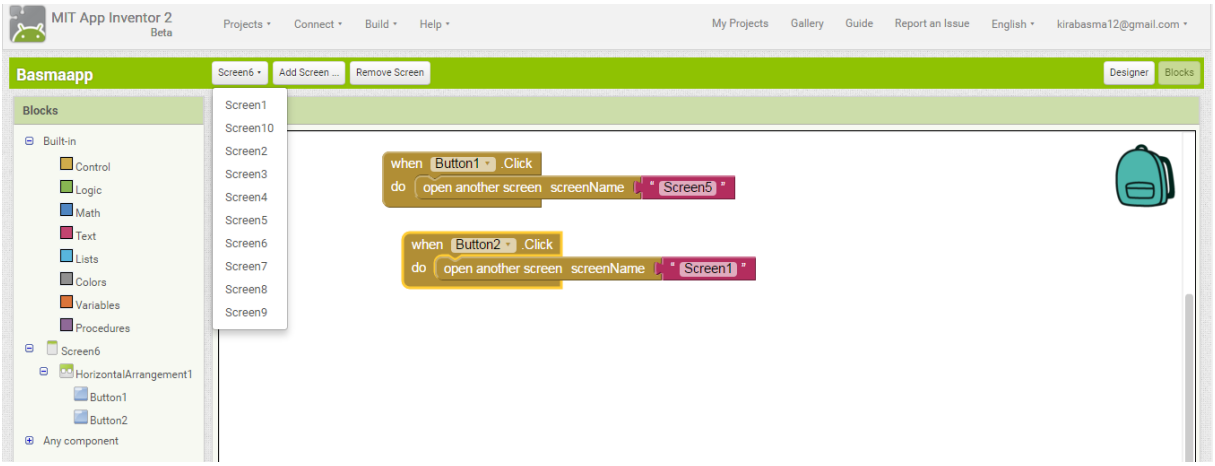

Fig.3.10. Code pour passer d'un écran à un autre.

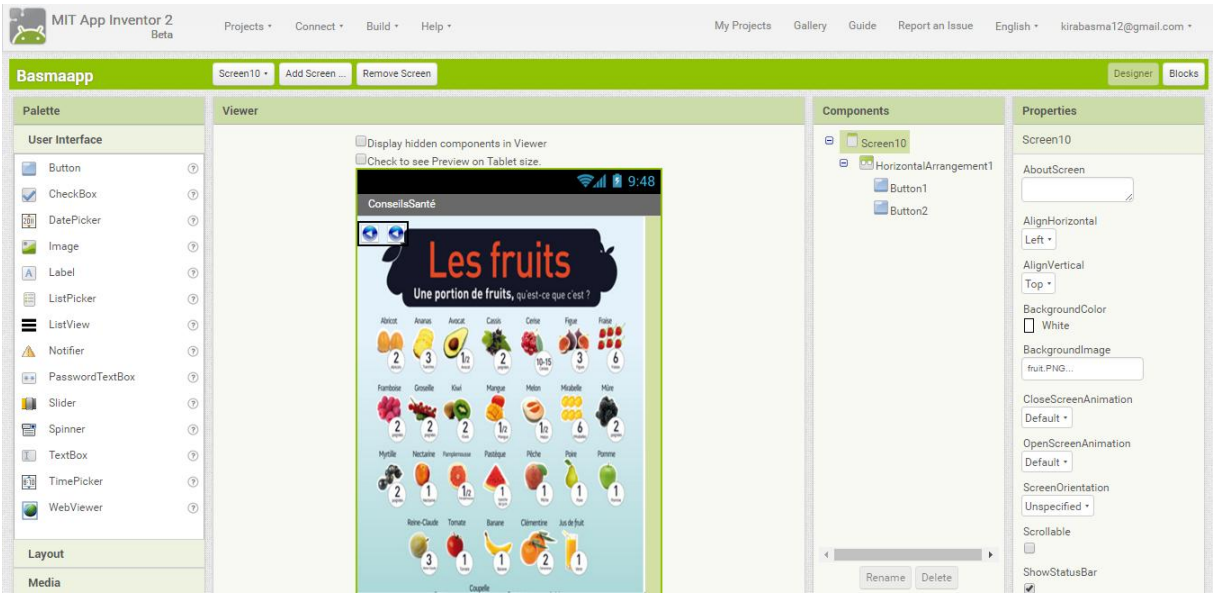

Fig.3.11. Exemple d'un écran de conseils de santé dans la partie « *Designer* ».

3.4. L'application en étape finale

Après l'utilisation des différents codes nécessaires à la réalisation de notre application mobile, nous somme abouti au résultat détaillé dans ce qui suit.

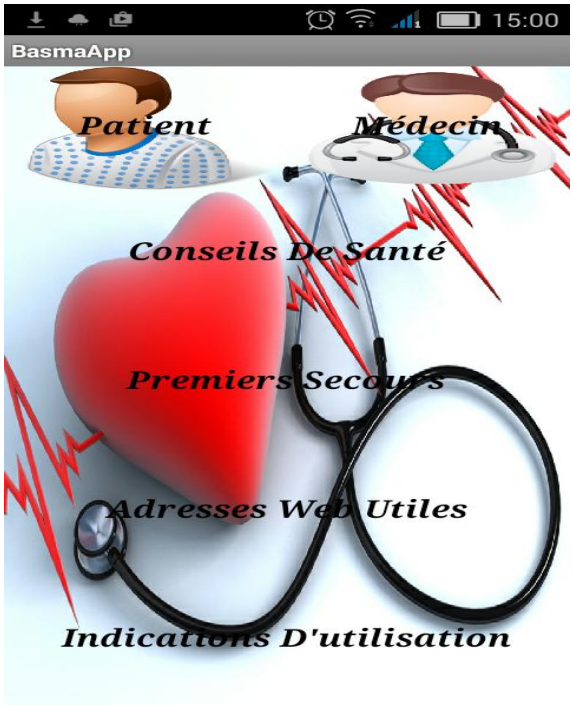

Fig.3.12. Ecran principal de l'application réalisée.

Comme l'indique la figure précédente, l'écran principal contient un bouton nommé « patient » et un autre nommé « médecin » ; ce qui permet la séparation des fonctionnalités offertes par notre application principale, en partageant cette dernière en deux applications secondaires, une appartient au patient et l'autre au médecin.

3.4.1. L'application « *patient* »

L'application « *patient* » est celle responsable de la transmission des données médicales du patient à son médecin. Pour cela elle permet en premier, l'établissement de la liaison Bluetooth entre l'appareil utilisé et le terminal mobile du patient (ou bien l'un de ses proches). Ainsi, l'envoi par SMS ou MMS de ces données au médecin.

Il faut signaler que le partage des informations médicales du patient avec son médecin peut se faire avec d'autre moyens: le courrier électronique (comme Gmail), le Viber, le WhatsApp …etc. Ceci représente l'un des améliorations remarquables de notre application, en offrant des moyens gratuits de transmission des données au médecin.

Fig. 3.13. Ecran principal de l'application « *Patient* ».

L'écran Bluetooth contient un bouton pour accéder à la liste des adresses MAC des appareils Bluetooth à proximité, une étiquète (Label) pour définir l'état de la liaison Bluetooth (non connecté en rouge, et connecté en vert), une autre pour afficher le fichier reçu de la liaison, un bouton pour la déconnexion et un autre pour retourner au écran principal de l'application « *Patient* ».

Fig.3.14. Ecran de la liaison Bluetooth.

Fig.3.15. Liste des appareils à proximité et leurs MAC.

Il faut signaler à ce niveau l'indisponibilité du capteur sous forme d'un « *Holter ECG* » muni d'un adaptateur Bluetooth. Ce qui a ralenti les tests effectués sur notre application. L'alternative trouvée lors des tests était l'utilisation des captures d'images manuelles ou bien des câbles (chose qui a réduit la mobilité désirée).

Les autres options de notre application fonctionnent ordinairement ; elles seront présentées en détail dans ce qui suit.

L'écran contact, offre à l'utilisateur la possibilité de contacter son médecin en lui envoyant un message (SMS, MMS ou bien un courrier électronique… etc.).

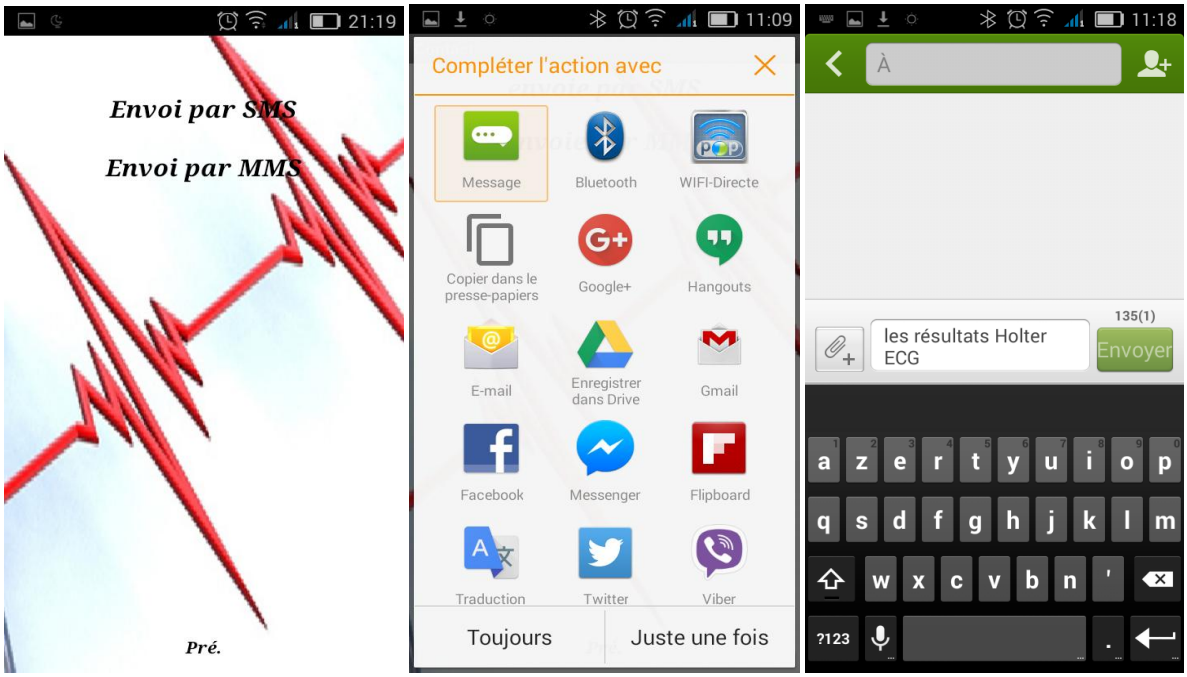

Fig.3.16. Les étapes pour partager les messages avec le médecin.

5.4.1. L'application « *Médecin* »

L'application « *Médecin* » permet à son tour de visualiser les données reçues de la part du patient à l'aide d'un « *image picker* », en utilisant le « *TextToSpeech* » pour informer le médecin de la date et l'heure de réception. Elle permet ainsi de répondre au patient à l'aide du bouton « *Répondre* », en lui envoyant des messages précis. Ces messages sont déjà écrits dans une liste d'une façon simple pour permettre au médecin de choisir une réponse courte et claire ; et l'envoyer rapidement au patient.

Fig.3.17. L'écran principal de l'application « *Médecin* ».

Le médecin après être informer de la date et l'heure de la réception, peut visualiser les fichiers reçus en cliquant sur l'écran. Le composant « *image picker* » va lui donner l'accès aux fichiers désirés.

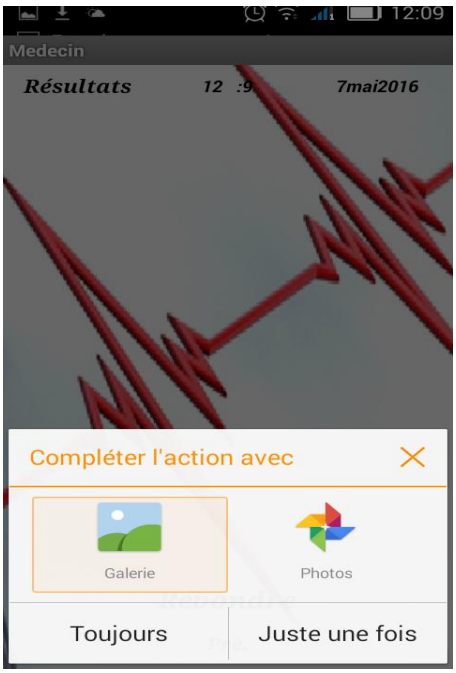

Fig.3.18. Ouverture des fichiers reçus.

Dans le cas d'urgences, et s'il apreçoi une anomalie dans les mesures prises par l' « *Holter ECG* », il peut interagir facilement et en temps réel en choisissant un message préecrit dans la liste offerte (il peut ajouter des expressions s'il ya temps), ainsi l'envoyer au patient imédiatement.

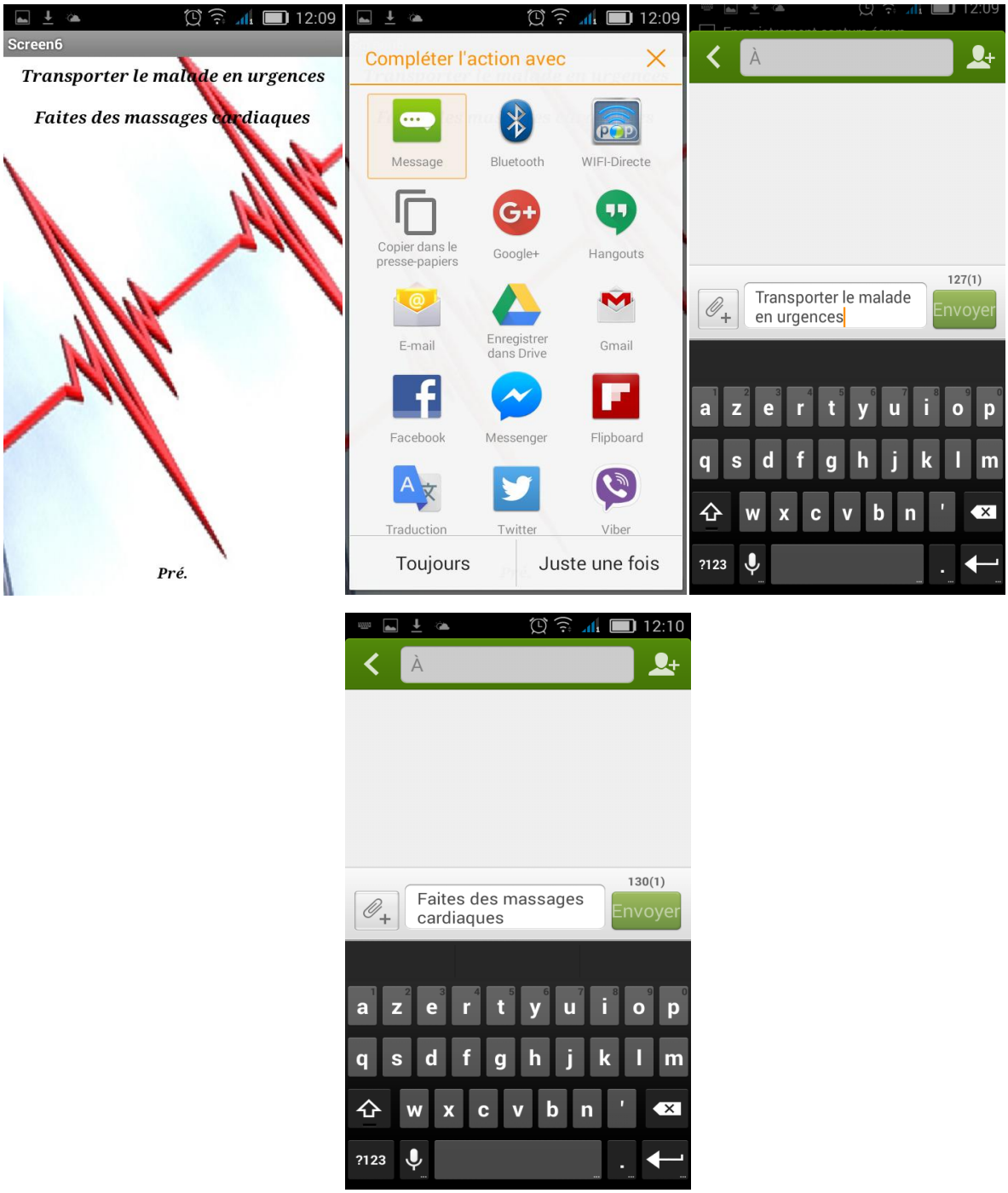

Fig.3.19. Les étapes effectuées par le médecin pour répondre au patient.

5.4.2. Les différentes améliorations

Pour améliorer les performances de notre application, il était nécessaire d'ajouter quelques pages d'utilité quotidienne comme : les pages de conseils de santé, la page des premiers secours et d'indication d'utilisation de notre application.

Le passage entre les différentes pages de conseils se fait par un simple click sur les boutons « *Pré.* » et « *Suiv.* ».

Fig.3.20. Exemple des écrans de conseils de santé.

La plupart des applications sont communicantes. Difficile de s'y tromper, car de toutes les permissions qui sont réclamées, la demande de connexion à Internet est l'une des plus fréquentes. C'est pour cela et pour enrichir encore plus notre application, nous avons utilisé un « *WebViewer* » pour accéder aux liens proposés ayant relation avec le sujet traité par l'application, dont le but principal est de donner des informations utiles aux différents utilisateurs (patients, médecins, et citoyens).

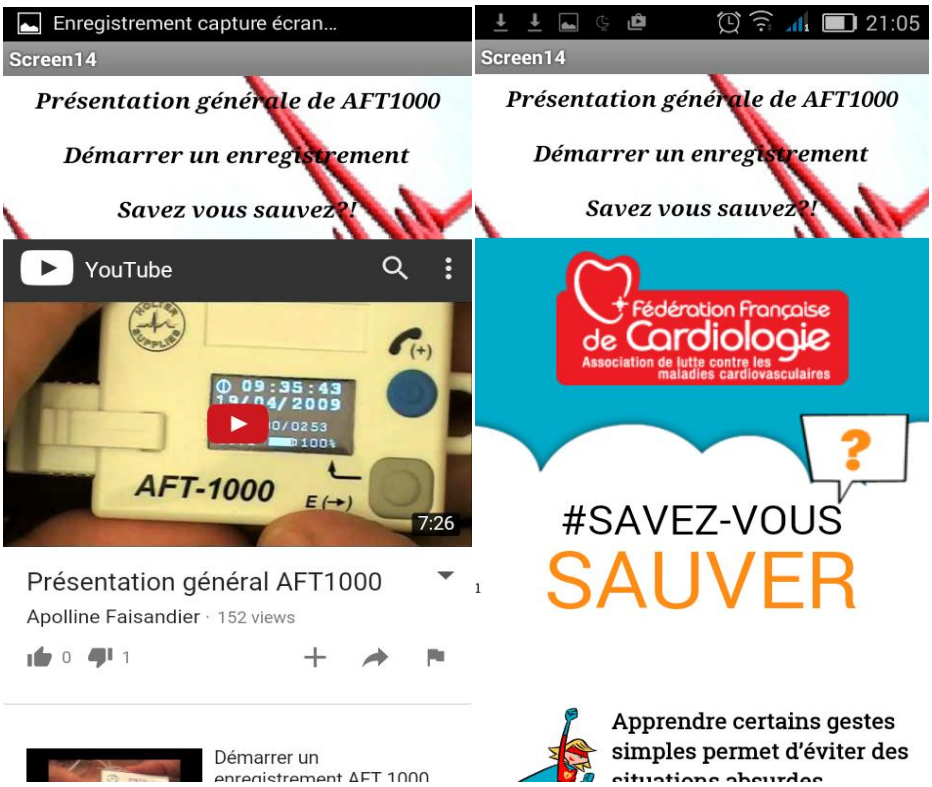

Fig.3.21. Exemples des pages web proposées.

6. Publication de l'application

Le développement d'une application est une étape en soi. Cela prend du temps, requiert des compétences et de la rigueur. Que nous voulions tirer parti de notre travail en vendant notre application ou bien gagner en notoriété en la fournissant gratuitement à la communauté, nous avons dans les deux cas une excellente raison de partager notre application avec les autres utilisateurs !

La publication d'une application n'est que le commencement du cycle de vie de cette dernière car une fois en ligne, nous aurons besoin de la mettre à jour régulièrement, soit pour corriger des bogues notifiés par les utilisateurs, soit pour proposer de nouvelles fonctionnalités ou adapter notre création aux différentes versions du SDK.

« *Google Play Store* » (le marché d'applications Android) permet aux développeurs du monde entier de mettre leurs applications à disposition des utilisateurs d'Android. Ceux-ci pourront ainsi télécharger, gratuitement ou moyennant une rétribution, n'importe quelle application déposée dans cet espace. La publication d'une application Android nécessite la création d'un paquetage (un fichier d'extension .apk) contenant tout le code et les ressources de l'application.

Pour garantir à nos utilisateurs une bonne installation et utilisation de notre application, il parait nécessaire de prendre le temps qu'il faut pour bien tester et vérifier celle-ci, ainsi de passer par toutes les étapes indispensables à la mise en application de notre solution sur le marché d'Android (Annexe IV).

7. Conclusion

Lors de ce dernier chapitre, nous avons détaillé la solution de télésurveillance proposée, en citant ainsi les différentes étapes de la réalisation de notre application « *BasmaApp* ».

Ce chapitre a donné une vue d'ensemble de toutes les opérations effectués sur le code, et testés sur le terminal mobile, et de toutes les difficultés rencontrées le long de la réalisation de cet œuvre en présentant vers la fin l'application en étape finale.

Conclusion générale

Une révolution technologique et numérique a été amorcée déjà depuis un certain nombre d'années dans le domaine de la santé, mais pour autant les nouvelles technologies sont loin de rayonner dans tous les établissements hospitaliers et auprès de tous les acteurs impliqués (patients, personnels soignants et praticiens), l'hôpital du futur progresse chaque année, certes un plus sûrement, mais néanmoins, reconnaissons-le, à vitesse assez lente. Donc, le champ est bien libre et l'horizon peut-être pas aussi bouché qu'il n'y paraît.

La prise en charge et le suivi des patients à domicile seront grandement facilités par le soutien d'appareils innovants de traitement d'informations visuelles et sonores, avec déclenchement et gestion des alarmes. L'autonomie des personnes âgées est également un secteur où l'innovation monte en puissance. La possibilité d'aider à la prévention de la dépendance en mettant place un suivi des troubles liés à la mobilité, et au déroulement des actes du quotidien, est envisagée. Pour les patients atteints de maladie cardiovasculaire, il sera bientôt possible de disposer d'une application mobile qui permettra non seulement de contrôler leur état cardiaque mais de transmettre des alertes en cas de dépistage de rythme ou d'infarctus.

L'objectif de notre travail était la conception et la réalisation d'un système ambulatoire qui aura à sa charge le suivi et la surveillance d'un patient souffrant d'une insuffisance cardiaque chronique et ce, en exploitant les smartphones sous Android.

Une meilleure communication entre les acteurs, l'accès à des informations complètes et une prise de connaissance rapide et aisée des informations sont autant d'aspects qui contribuent à la qualité de la prise en charge de patient. La solution proposée représente un outil qui diminue la distance entre le patient et son médecin cardiologue, en améliorant la qualité des services de santé, un outil qui prendra sa place dans le domaine de cardiologie médicale de demain.

En guise de perspectives, plusieurs travaux pourraient être envisagés pour améliorer ce travail de plus en plus. Parmi ces perspectives, nous parlons également de la géo-localisation du patient à l'aide du fameux GPS. Un des fonctionnalités les plus appréciées sur les plates-formes mobiles modernes, la géo-localisation permet de réaliser des applications innovantes dans le domaine de la télésurveillance des patients et plus particulièrement les personnes âgées.

Comme autre perspective, nous pouvons penser aussi à concevoir un système plus général, qui regroupe plusieurs paramètres : pression artérielle, température corporelle, EEG, etc.

En conclusion, nous pouvons dire que tous les développeurs sont ainsi sur un même pied d'égalité, qu'ils soient une grande entreprise ou quelques jeunes dans un garage ; chacun peut ajouter de la valeur au déploiement de technologie dans notre vie quotidienne.

Références et bibliographie

Références et bibliographie

[1]Guy Cazuguel, « Technologies et innovations hospitalières N°13 », Technopole de Brest, Janvier 2012.

[2]Guy Vallancien, «Télémedecine, Quinzes villes connectées jeudi », Le télégramme, Le 28 mars 2011.

[3]Agence Régionale de Santé (ARS), Île-de-France, « La télémédecine au secours des urgences », Le 03 juin 2013.

[4]Jean-Bernard Schroeder, **Pierre Leurent**, « Télémédecine 2020, Modèles économiques pour le télé-suivi des maladies chroniques », Livre Blanc, Avril 2013.

[5]Pr Arnaud Lazarus, « Progrès techniques face aux maladies du cœur **», Le** 12 juil 2010.

[6]Pr Jacques Beaune, « Pourquoi docteur », cardiologue et président d'honneur de la FFC, **Savez-vous sauver**, Le 23 juin 2015.

[7]Francis Jaluzot, « Le guide complet Android», Mobile Passion N°8, Septembre 2012.

[8]Nguyen Tien Thinh, « Systèmes d'exploitation pour les mobiles », Institut de la Francophonie pour l'informatique, Juillet 2009**.**

[9]Société Gartner, « Statistiques du marché des systèmes d'exploitation pour mobiles » 2008.

[10]Frédéric Brault, « Hackez Google Android, introduction à la programmation système », l'École Polytechnique, 2009.

[11]Eric Schmidt Mobile, « le rendez-vous international de l'industrie du mobile, Google», World Congress 2010.

[12]Damien Guignard, **Julien Chable et Emmanuel Robles**, « Programmation Android De la conception au déploiement, avec le SDK Google Android 2 », EYROLLES, 2010.

[13]Mark Murphy, **Éric Jacoboni**, et **Arnaud Farine**, « L'art de développement Android », PEARSON, 2009.

[14]Florent Garin, « ANDROID, Développer des applications mobiles pour les Google Phones », 2009.

[15]Zerrouki Fodil, « Conception et réalisation d'une carte d'acquisition ambulatoire de transmission sans fil et de traitement de signaux biomédicaux », Mémoire de Magister, Université de Tizi Ouzou, 2013.

[16]Olivier Le Goaer, « Programmation des applications mobiles avec Android », 2012.

[17]Jean-Francois Lalande, « Développement sous Android »**,** ENSI Bourges, Novembre 2012.

[18]AndroWiiid, Frédéric Espiau et **DakuTenshi**, « Créez des applications pour Android », le site du zero, Le 10 mai 2012.

[19]MERZOUGUI Rachid, « Conception et développement d'applications et services dédiés à la santé sur des terminaux mobiles », Thèse de Doctorat, Université de Tlemcen faculté de technologie**,** Juillet 2011.

[20]Tutorials for App Inventor, « Explore MIT App Inventor », <http://www.appinventor.mit.edu. >**,** Consulté le 3 mars 2016.

[21]Fabien Brissonneau, « Présentation de la formation, Android5 », <http://www.alphorm.com>**,** Consulté le 25 janvier 2016.

[22]Fukuda denshi, « Holter ECG ambulatoire, LTD », <http://www.fukuda.co.jp>**,** consulté le 24 février2016.

[23]Bucky Roberts, « Java / Android Development », <http://www.thenewboston.com>**,** The new boston Forum, Consulté le 25 février 2016.

[24]Android App developers India, « Android Game Development India», <http://www.siliconinfo.com >**,** Consulté le 6mars2016.

[25]Derek Banas, Android Development for Beginners, <http://www.youtube.com>**,** consulté le 17 mars 2016.

[26]Martyn Currey, « Android MIT App Inventor ,Auto Connect To Bluetooth », <http://www.martyncurrey.com>**,** Consulté le 16 avril 2016.

[28]App Inventor, « exemples d'IHM **Android** pour carte Arduino - éduscol STI », <http://www.eduscol.education.fr. >**,** Consulté le 21 avril 2016.

Annexes

Annexe I : Comparaison entre les systèmes d'exploitation de mobiles.

Introduction :

Actuellement, il existe plusieurs SE pour les terminaux mobiles comme Windows Mobile, Palm OS, Symbian, BlackBerry qui sont les systèmes propriétaires. De plus, il ya des plateformes libres ou code source ouvert comme Moblin.org, Ubuntu MID Edition, Android, etc. Cette annexe va nous présenter un petit tableau de comparaison entre les SE de mobiles les plus connus.

Comparaison entre les SE pour mobile :

Mobile plate-forme	**Blackberry**	**Palm OS**	**Symbian**
Société	**RIM**	**Palm**	**Symbian**
	"Standard d'or" pour courriel.	logique, intuitif, simple	Les manufactures peuvent ajouter facilement ses technologies, ses infrastructures à la plateforme
Caractéristiques	- Multitâches. - Bonne intégration avec autres plateformes. - Déploiement facile pour une société. - Gestion facile. - La durée de vie de la batterie plus longue. - Facteur de forme est petit et clavier maniable. - Sécurité haute sans menace d'application coquine, virus, etc.	- Utilisation simple et facile. - Ecran tactile mono tâche. - Navigation facile	- Multitâches. - Ne supporte pas CDMA, - Utilisée seulement pour GSM. - Plusieurs fonctions. - Caractéristique prédominant. - Beaucoup d'applications tierces. - USB v2.0, Bluetooth v2.0, WLAN, IrDA & serial TCP, IPv4, IPv6, MSCHAP v2, - PPP WiFI integer.

Applications principales	- Utilisation facile de quelques applications simples comme : carnet d'adresses, gestion du temps, SMS, téléphone, Calendrier, Bloc-notes, Tâches, Calculatrice, Gestion de mots de passe … etc.	- Calendrier. - Variété d'applications.	- Carnet d'adresses, gestion du temps, SMS, téléphone, Calendrier, Calculatrice… etc.

Mobile plate-forme	**Windows Mobile**	**Ubuntu MID Edition**	**Android**
Société	**Microsoft**	**Ubuntu communauté (Canonical Ltd et Intel)**	**OHA**
Caractéristiques	-Capacité de fonctionnement des logiciels sur « Windows » (seulement Windows), la compatibilité avec tous les logiciels de « Windows ». - multi tâche : robuste. - « smartdial »	- Utilisation simple. - Petite taille/ facteur de forme. - QWERTY clavier - clavier numérique virtuel ou physique. - Un écran tactile de 4 à 7 pouces. - Pouvoir intégrer avec les réseaux mobiles sociaux et les sites web 2.0. - Wi-Fi, 3G, Bluetooth, GPS, WiMAX. - Stockage: 2GB to 8GB Flash ou stockage du disque, 256MB+ /512MB+. - OpenGL 3D. - Gestion	- La plate-forme basée sur Linux 2.6.25 pour ARM. - Supporte le système du fichier FAT32. - Supporte TCP/IP (TCP, UDP, etc.) - Minimum de 128 Mo RAM et 256 Mo de mémoire flash. - 802.11 b / g Wi-Fi. - USB 2.0. - Bluetooth 2.0. - Stockage amovible. - Résolution HVGA. - 16 bits de couleurs. - Ecran tactile. - Clavier QWERTY. - 5 directions de navigation.

		d'utilisation en utilisant le Java, le Flash, l'AJAX ; le HTML, Clutter, Python avec GTK, C/C++ avec GTK et Java. - PDF. - La durée de vie de la batterie est longue.	- Notifications LEDs, Vibration. - GPRS, EDGE, UMTS, HSDPA. - Roaming internationale, SMS, MMS. - Des services pour la téléphonie : l'appel en attente, conférence téléphonique. - USSD. - RIL.
Applications principales	- Calendrier. - Gestion de la tâche - Internet Explorer Mobile…etc.	- Les fonctions de téléphonie ordinaire: contrôle d'appel, etc.	- Les fonctions de téléphonie ordinaire: contrôle d'appel, conférences téléphoniques …etc. - Navigateur Web (supporte le HTML et XHTML) - Réveil, calendrier, Caméra, messagerie instantanée, MMS, Voice Dialer, GoogleSearch.

Conclusion :

Cette annexe a donné une petite comparaison entre quelques SE de mobiles, les plus connus. Chaque type de SE est souvent approprie à quelques modèles concrets des mobiles. Cependant, chaque type de SE a des avantages et des limitations. Il est donc très difficile de choisir la plateforme la plus répondante à l'objectif d'une société ou à une personne.

Annexe II : Exploitation de l'API Bluetooth sur Android Studio

Introduction :

Cette annexe a pour but d'expliquer comment utiliser le Bluetooth dans les applications Android, ceci à travers l'API fournie par Android. Android inclus un support du Bluetooth qui permet aux terminaux mobiles de pouvoir échanger des données entre eux en utilisant cette technologie.

Utilisation de l'API Bluetooth :

La première étape dans l'intégration du Bluetooth dans votre application est de vérifier si le terminal mobile en question possède cette technologie.

Cela est très facile en utilisant l'API Bluetooth et plus particulièrement le BluetoothAdapter.

Voici un petit bout de code qui vérifie la présence du Bluetooth dans un mobile :

```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
if (bluetoothAdapter == null)
   Toast.makeText(TutoBluetoothActivity.this, "Pas de Bluetooth",
                  Toast.LENGTH_SHORT).show();
else
   Toast.makeText(TutoBluetoothActivity.this, "Avec Bluetooth",
                  Toast.LENGTH_SHORT).show();
```

Il ne faut pas oublier la permission d'utilisation du Bluetooth dans le Manifest :

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
```

La deuxième étape est d'activer le Bluetooth :

```
private final static int REQUEST_CODE_ENABLE_BLUETOOTH = 0;
….;
if (!bluetoothAdapter.isEnabled()) {
   Intent enableBlueTooth = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
   startActivityForResult(enableBlueTooth, REQUEST_CODE_ENABLE_BLUETOOTH);
}
```

Tout ce que fait ce code, est de vérifier si le Bluetooth est déjà activé. Sinon il lancer une boite de dialogue pour demander à l'utilisateur d'activer le Bluetooth.

Il suffit de surcharger la méthode onActivityResult pour savoir si l'utilisateur a activé le Bluetooth ou pas.

```
      @Override
      protected void onActivityResult(int requestCode, int resultCode, Intent data) {
             super.onActivityResult(requestCode, resultCode, data);
             if (requestCode != REQUEST_CODE_ENABLE_BLUETOOTH)
                 return;
```

```
            if (resultCode == RESULT_OK) {
                // L'utilisation a activé le bluetooth
            } else {
                // L'utilisation n'a pas activé le bluetooth
            }
        }
```

Nous avons besoins de la permission suivante :

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

Le bout de code, qui nous permet d'activer le Bluetooth sans l'avis de l'utilisateur :

```
if (!bluetoothAdapter.isEnabled()) {
   bluetoothAdapter.enable();
}
```

Obtenir la liste des terminaux déjà connus :

Nous pouvons obtenir la liste des terminaux déjà connus et ainsi la stocker pour une utilisation ultérieure.

```
private Set<BluetoothDevice> devices;
…
devices = bluetoothAdapter.getBondedDevices();
for (BluetoothDevice blueDevice : devices) {
Toast.makeText(TutoBluetoothActivity.this,"Device = " + blueDevice.getName(),
 Toast.LENGTH_SHORT).show();
}
```

Recherche de nouveaux périphériques :

Créer un Broadcast receiver qui sera avertit lors de la détection d'un nouveau terminal :

```
private final BroadcastReceiver bluetoothReceiver = new BroadcastReceiver() {
public void onReceive(Context context, Intent intent) {
String action = intent.getAction();
if (BluetoothDevice.ACTION_FOUND.equals(action)) {
BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
Toast.makeText(TutoBluetoothActivity.this, "New Device = " + device.getName(),
Toast.LENGTH_SHORT).show();
  }
}
};
```

Enregistrer notre broadcast :

```
IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
registerReceiver(bluetoothReceiver, filter);
```

Lancer le scan grâce à la méthode startDiscovery :

```
bluetoothAdapter.startDiscovery();
```

Attention, il ne faut surtout pas oublier que dans le « *onDestroy* » il faut :

- Arrêter la découverte des nouveaux périphériques.
- Désabonner notre BroadcastReceiver.

Avec :

```
@Override
protected void onDestroy() {
  super.onDestroy();
  bluetoothAdapter.cancelDiscovery();
  unregisterReceiver(bluetoothReceiver);
}
```

Pour que notre terminal mobile puisse être trouvé et lié avec d'autres terminaux, il faut le rendre visible. Pour cela il suffit d'utiliser le code suivant :

```
Intent discoverableIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION, 300);
startActivity(discoverableIntent);
```

Etablir une connexion :

Pour établir une connexion et faire échanger des données, il faut un coté serveur (celui qui reçoit la connexion) et un autre client (celui qui envoie la connexion).

Code du « *Serveur* » :

```
private class AcceptThread extends Thread {
   private final BluetoothServerSocket mmServerSocket;

   public AcceptThread() {
        BluetoothServerSocket tmp = null;
        try {
             tmp = mBluetoothAdapter.listenUsingRfcommWithServiceRecord(NAME,
MY_UUID);
        } catch (IOException e) { }
        mmServerSocket = tmp;
   }

   public void run() {
        BluetoothSocket socket = null;
        while (true) {
            try {
                socket = mmServerSocket.accept();
            } catch (IOException e) {
```

```
                break;
            }

            if (socket != null) {
                manageConnectedSocket(socket);
                mmServerSocket.close();


 break;
            }
        }
    }

    public void cancel() {
        try {
            mmServerSocket.close();
        } catch (IOException e) { }
    }
}
```

Code du «Client» :

```
private class ConnectThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final BluetoothDevice mmDevice;

    public ConnectThread(BluetoothDevice device) {
        BluetoothSocket tmp = null;
        mmDevice = device;
        try {
            tmp = device.createRfcommSocketToServiceRecord(MY_UUID);
        } catch (IOException e) { }
        mmSocket = tmp;
    }

    public void run() {
        mBluetoothAdapter.cancelDiscovery();
        try {
            mmSocket.connect();
        } catch (IOException connectException) {
            try {
                mmSocket.close();
            } catch (IOException closeException) { }
            return;
        }
         manageConnectedSocket(mmSocket);
    }

    public void cancel() {
        try {
            mmSocket.close();
        } catch (IOException e) { }
    }

}
```

Une fois le serveur et le client connectés, chacun va pouvoir échanger avec l'autre au travers de son objet « *BluetoothSocket* ». Comme pour toute communication par les sockets, les méthodes et techniques sont les mêmes.

Conclusion :

Cette annexe à donner les grandes lignes du travail de l'arrière plan et de la gestion de Bluetooth, en présentant le code utilisé sur « *Android Studio* ».

Ce qu'il faut prendre en considération, c'est que la connectivité Bluetooth doit être prise comme de véritable opportunité pour créer des applications communicantes. Sachez cependant que l'activation de ces fonctionnalités est consommatrice d'énergie et que l'autonomie de l'utilisateur n'en sera que réduite.

Annexe III : Création d'une nouvelle application « *HelloWorld* » sur Android Studio.

Introduction :

Android Studio est l'IDE officiel de la plateforme Android. Après avoir installé l'Android Studio, et le JDK (Java Developpement Kit), nous pouvons créer notre première application. Cette annexe va nous déterminé les étapes nécessaires à la création d'un premier projet sous l'Android Studio.

Création du projet :

1. Installation de JDK à partir du site officiel d'Oracle et de l'IDE « Android Studio », à partir de son site officiel de la plateforme Android (developper.Android.com).

Fig. AIII.1. Ecran principal d'Android Studio.

2. Création d'un nouveau projet en choisissant le nom de l'application :

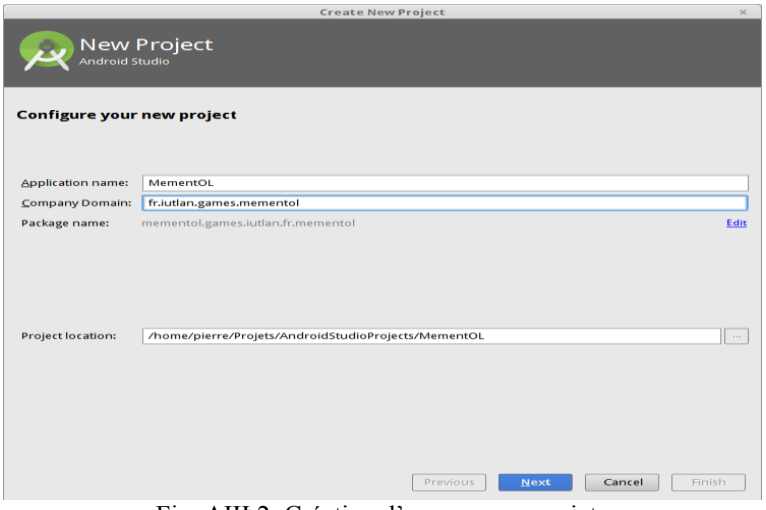

Fig. AIII.2. Création d'un nouveau projet.

3. Il faut ensuite choisir la catégorie des terminaux mobiles désirée et aussi la version adéquate :

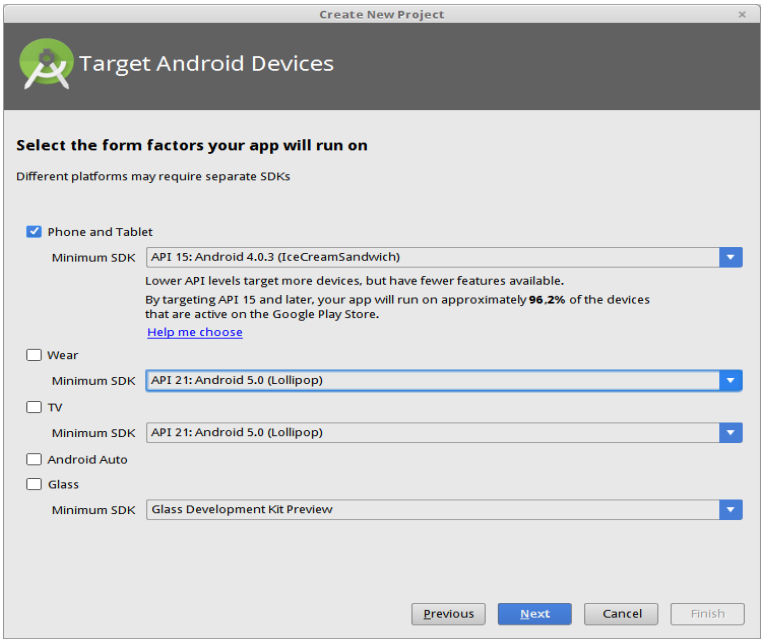

Fig. AIII.3. Choix de terminal mobile et de version.

4. Et puis commencez une nouvelle activité « *Blank Activity* » :

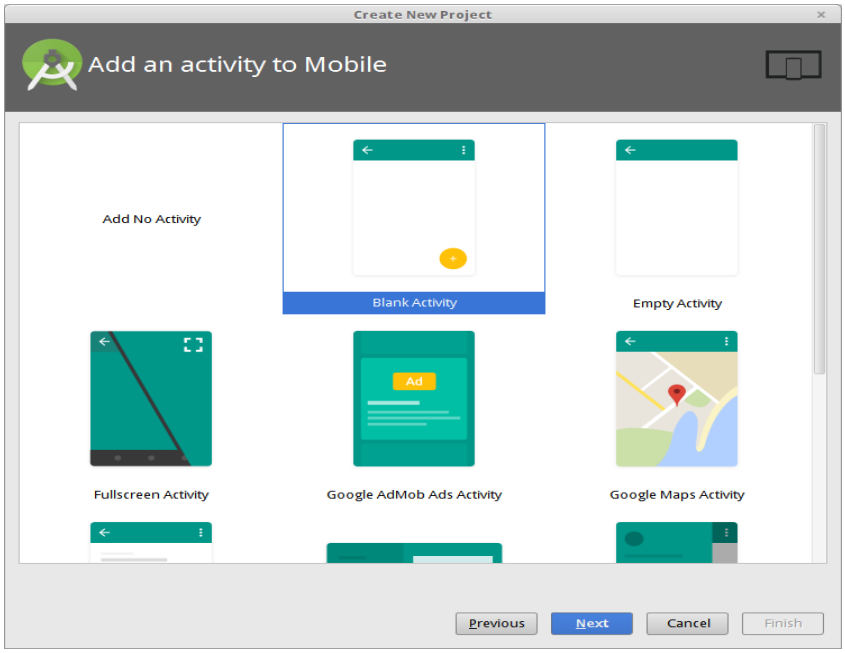

Fig. AIII.4. Création d'une nouvelle activité.

5. L'assistant demande ensuite plusieurs informations :
   - Nom de l'application, ex : HellorHorde.

- Nom de la classe principale : MainActivity.
- Nom du layout de la classe principale : activity_main.
- Nom du layout du menu principal : menu_main.
- Tout peut être renommé ultérieurement, voir refactor/rename.
- Le package du logiciel a été défini dans le premier écran.

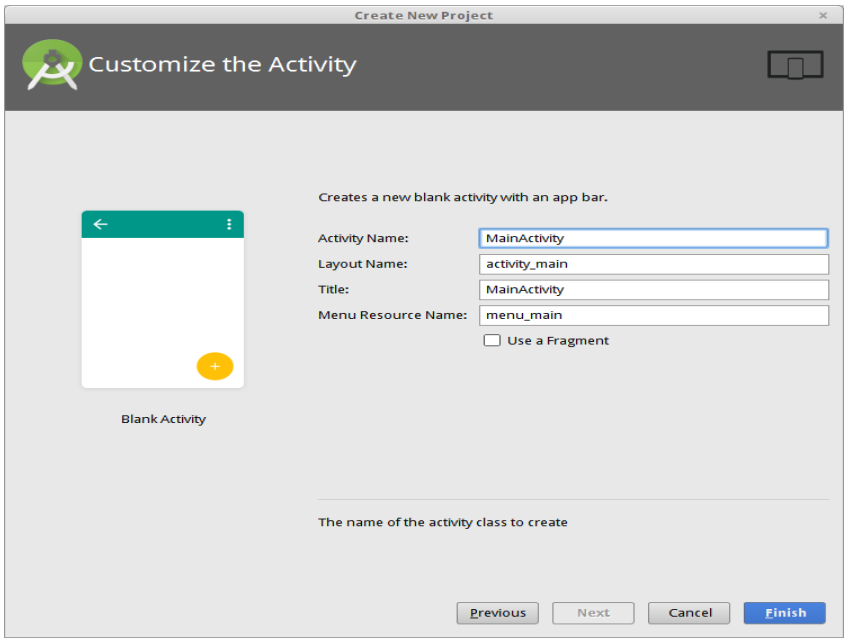

Fig. AIII.5. Nomination de la nouvelle activité.

6. L'assistant a créé de nombreux éléments visibles dans la colonne de gauche de l'IDE :

- manifests : description de l'application.
- java : les sources, rangées par paquetage.
- res : ressources = fichiers XML et images de l'interface, il y a des sous-dossiers :
  - drawable : images, icônes utilisés dans l'interface.
  - layout : interfaces (disposition des vues sur les écrans).
  - menu : menus contextuels ou d'application.
  - mipmap : images, icônes utilisés dans l'interface.
  - values : valeurs de configuration, textes . . . etc.
- Gradle scripts : c'est l'outil de compilation du projet.

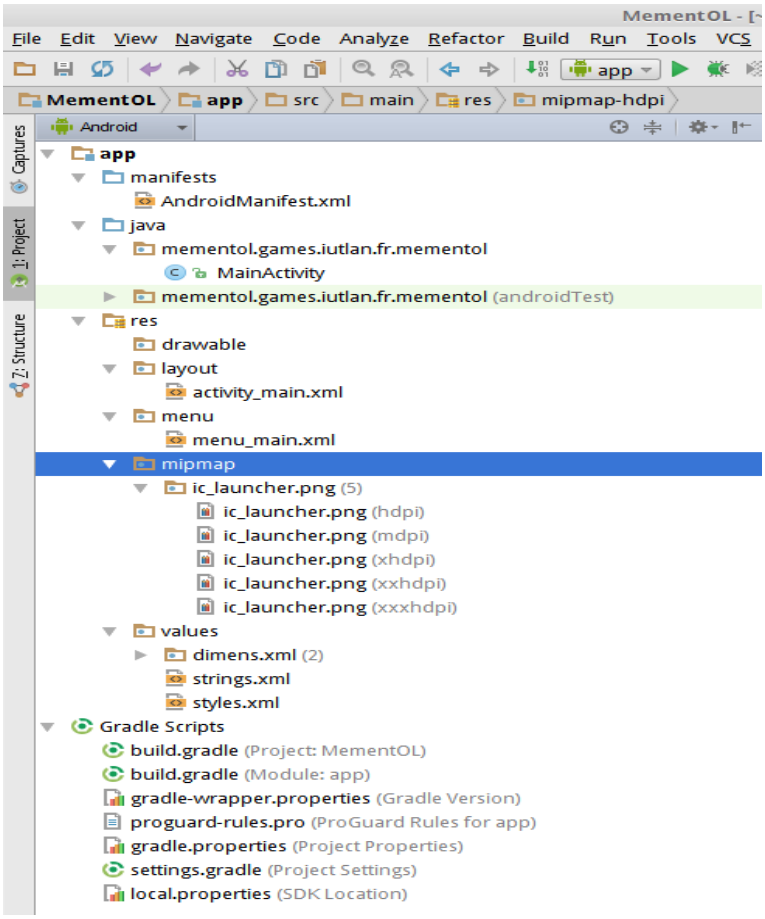

Fig. AIII.6. Les éléments créés par l'assistant.

7. Editeurs spécifiques :

Studio fournit des éditeurs spécialisés pour les fichiers XML, par exemple :

- Formulaires pour :
  - res/values/strings.xml : textes de l'interface.
- Éditeurs graphiques pour :
  - res/layout/*.xml : disposition des contrôles sur l'interface.

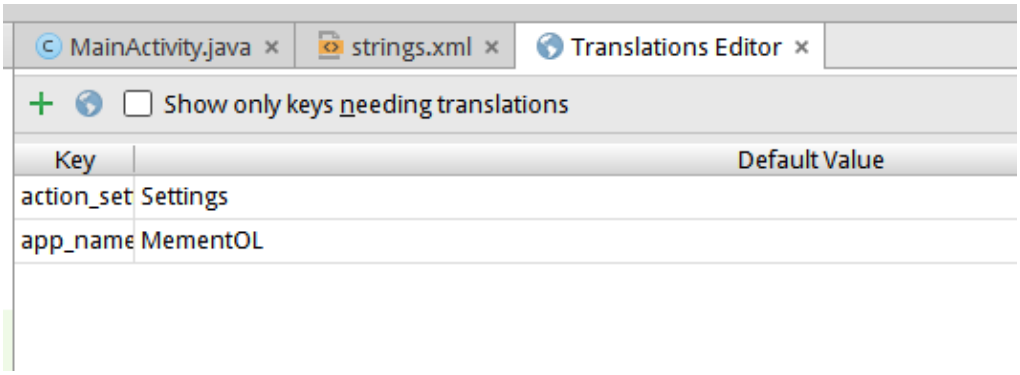

Fig. AIII.7. Editeurs spécifique

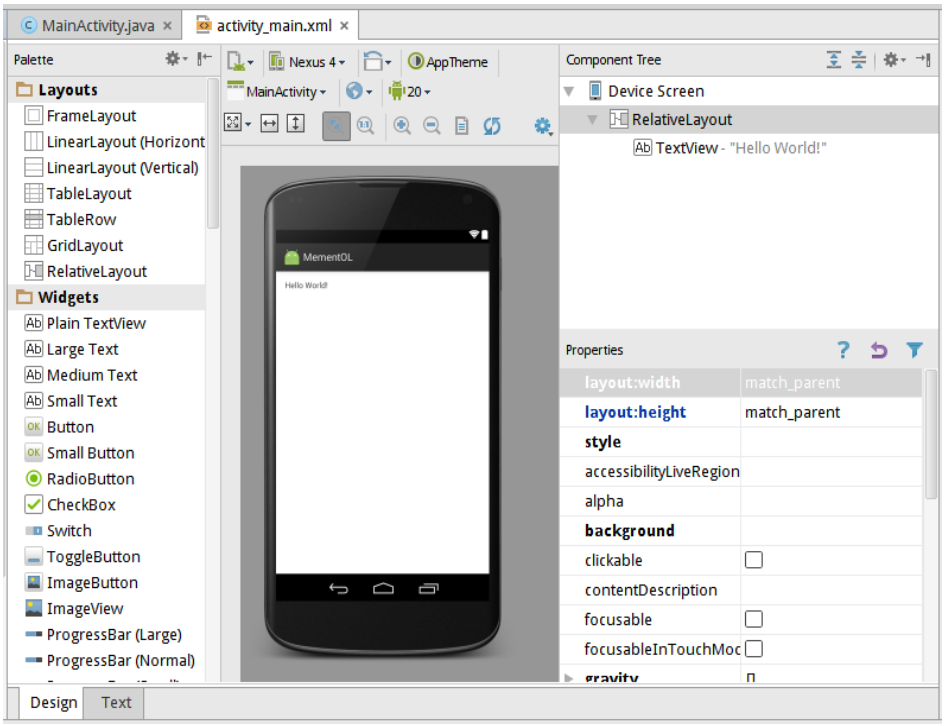

Fig. AIII.8. Affichage de l'activité sur l'IDE.

8. Source XML sous-jacent :

Ces éditeurs sont plus confortables que le XML brut, mais ne permettent pas de tout faire. Dans certains cas, il faut éditer la source XML directement :

```
<RelativeLayout
      xmlns:android=
            "http://schemas.android.com/apk/res/android"
      android:layout_width="match_parent"
      android:layout_height="match_parent" >
     <TextView
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:text="@string/hello_world" />
</RelativeLayout>
```

9. Reconstruction du projet :

- Automatique :
  - Ex: modifier le fichier res/values/strings.xml ou une source Java.
  - Gradle compile automatiquement le projet.
- Manuelle, parfois nécessaire quand on modifie certaines ressources :

- Sélectionner le projet et choisir menu Build/Clean...etc.

10. Première exécution :

Le SDK Android permet de :

- Installer l'application sur un vrai téléphone mobile connecté par USB.
- Simuler l'application sur une tablette virtuelle *AVD*.

L'assistant de création d'AVD demande :

- Modèle de tablette ou téléphone à simuler.
- Version du système qu'il doit contenir.
- Orientation et densité de l'écran.
- Options de simulation :
  - Snapshot : mémorise l'état de la machine d'un lancement à l'autre, mais exclut UseHost GPU.
  - Use Host GPU : accélère les dessins 2D et 3D à l'aide de la carte graphique du PC.
- Options avancées :
  - RAM : mémoire à allouer, mais est limitée par votre PC.
  - Internal storage : capacité du flash interne.
  - SD Card : capacité de la carte SD simulée supplémentaire (optionnelle).

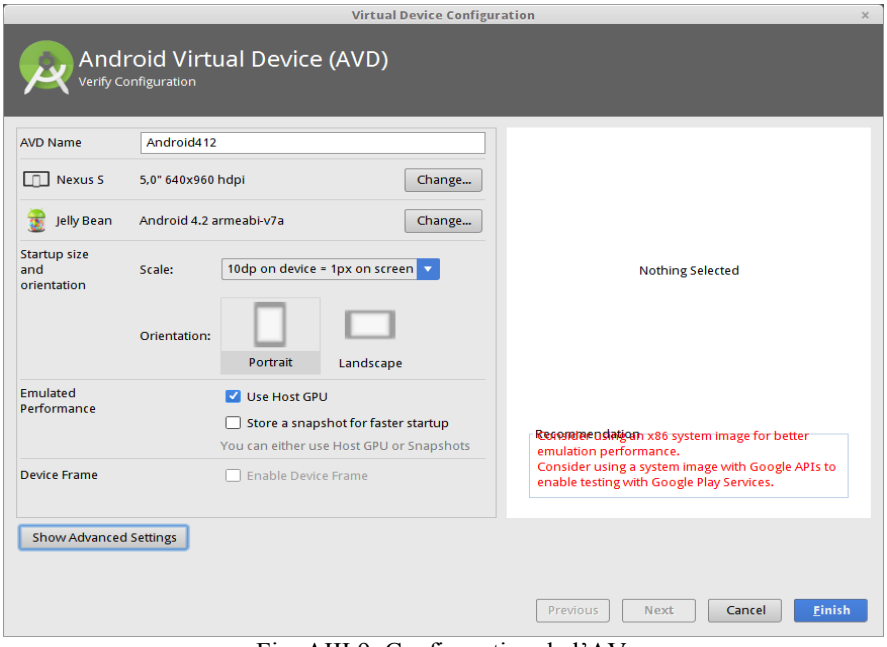

Fig. AIII.9. Configuration de l'AV

11. Lancement d'une application :

Bouton vert pour exécuter, bleu pour déboguer.

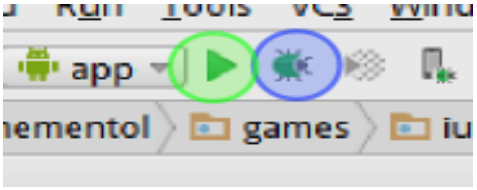

Fig. AIII.10. Boutons d'exécution et de débogage.

Résultat sur l'AVD :

Fig. AIII.11. L'affichage de la première application sur l'écran de l'AVD.

Conclusion :

Cette annexe a donné une description détaillée de la création d'une application sur « *Android Studio* ». Elle a expliqué ainsi les différentes étapes qu'un développeur doit passer pour aboutir à un résultat simple (l'affichage d'un simple message sur l'écran).

Annexe IV : Publication d'une application sur Google Play Store.

Introduction:

La publication d'une application sur le Play Store doit être un événement bien préparé ; nous devons tester, en autres, les performances de celle-ci, ainsi que sa compatibilité avec les différents modèles de téléphones (différents écrans) et les différentes versions d'Android. Cette annexe va nous expliqué d'une façon générale les étapes nécessaires à la publication d'une application.

Les étapes pour publication d'une application sur Google Play Store :

1. Choix de version :

Que ce soit pour des raisons marketing, pour notifier l'ajout de corrections et/ou de nouvelles fonctionnalités ou pour suivre l'évolution de l'application par l'équipe de développement, nous devons spécifier la version de notre application. Un numéro de version d'une application permet notamment de faire le lien avec de potentielles dépendances vers d'autres briques logicielles et de permettre de gérer la compatibilité avec les différentes versions de la plate-forme Android.

Chaque application que nous publions doit avoir un code et un nom de version. Nous pouvons les définir dans le concepteur dans le panneau Propriétés pour le composant « *Screen1* ».

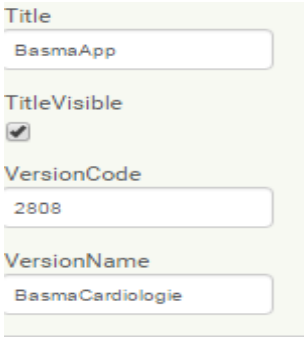

Fig. AIV.1. Nom et code de version.

2. Téléchargement du fichier « *.apk* » :

Une fois nous terminons notre application, nous pouvons télécharger le fichier .apk Android comme suit :

- Accéder à l'écran d'accueil de l'application que nous voulons télécharger dans l'App Inventor.
- cliquer sur « *Build*».
- cliquez sur «*App save .apk to my computer*» cela nous invite à enregistrer l'application quelque part.

Fig. AIV.2. Les étapes de téléchargement du fichier.apk.

Une fois le .apk est téléchargé, nous somme prêt à commencer le processus de publication.

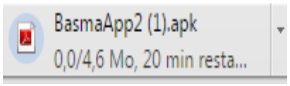

Fig. AIV.3. Le fichier.apk.

3. Publication de l'application :

Nous pouvons maintenant aller à «*Google Play Publishing Home* » et suivre les instructions pour publier notre application sur Google Play.

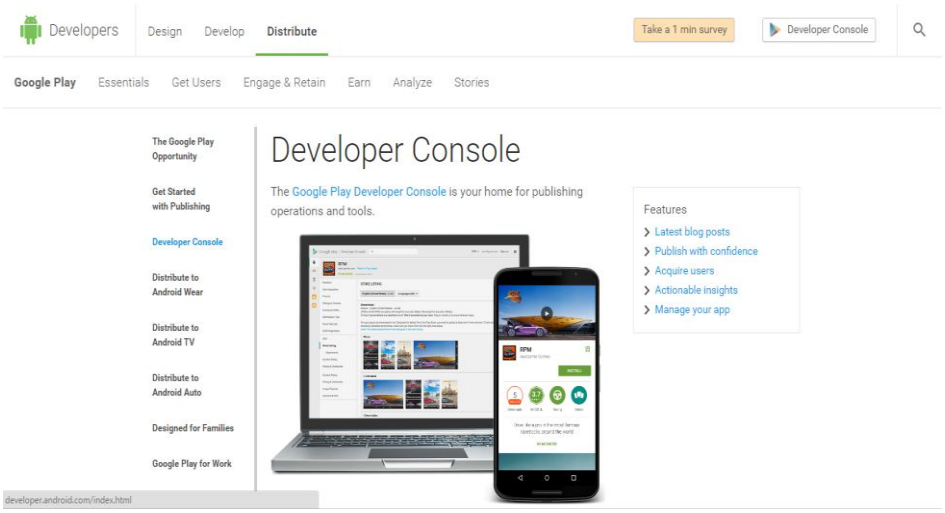

Fig. AIV.4. La page de publication dans « *Google Play* ».

4. Configurer des tests alpha et bêta :

Grâce à la Console développeur de Google Play, nous pouvons sélectionner des groupes d'utilisateurs qui recevront différentes versions de test bêta de notre application.

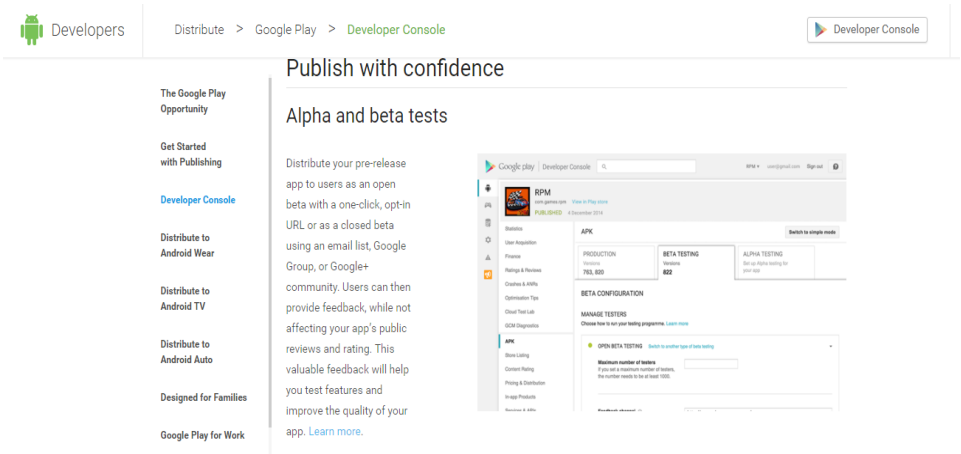

Fig. AIV.5. Tester l'application.

À l'aide des outils de tests alpha et bêta disponibles dans notre compte de développeur, nous pouvons tester différentes versions de notre application.

Pour effectuer un test, il faut d'abord :

- Choisir le mode de test :

- Soit fermé (pour une liste déterminée d'utilisateurs ou testeurs à l'aide de leurs adresses e-mail).
- Ou bien ouvert (pour un nombre maximal d'utilisateurs pour tester votre application bêta sans avoir besoin de spécifier des adresses e-mail ni de créer un groupe Google).

- Ajouter un fichier APK au test. Pour cela nous devons:
  - Sélectionner une application.
  - Dans le menu de gauche, cliquer sur APK > Tests bêta ou Tests alpha.
  - Cliquer sur Importer.
  - Sélectionner le fichier APK.
  - Cliquer sur Publier.
- Recueillir des commentaires :
  - Une fois que les testeurs alpha et bêta ont accepté de participer et qu'ils ont installé notre application à l'aide du lien fourni, ils passent automatiquement à la version de test en quelques minutes.
  - Étant donné que les testeurs ne peuvent pas laisser d'avis publics relatifs aux applications au stade alpha ou bêta sur Google Play, il est préférable d'inclure un mode d'envoi de commentaires spécifique ou d'indiquer aux utilisateurs comment ils peuvent envoyer leurs commentaires (par e-mail, via un site Web ou sur un forum de messages, par exemple).

Conclusion :

La publication reste une activité à part entière, il faut : répondre aux utilisateurs, corriger les bogues, ajouter de nouvelles fonctionnalités, etc. Tout comme pour un site Internet, il faudra aussi apporter un soin particulier au référencement de notre application : est-ce que les utilisateurs trouvent notre application sur le Google Play Store? Utilisons-nous les bons mots pour que notre application apparaisse en premier lors d'une recherche sur le Play Store?

La publication d'une application n'est qu'un début, la suite dépendra de nous !