



TRAVAIL PERSONNEL ENCADRÉ

RAPPORT FINAL

IFI-Promotion :23

PROCESSUS DE DEVELOPPEMENT RAPIDE D'APPLICATION MOBILE

Rédiger par :

Abdoul-Fatao OUEDRAOGO

Encadreur :

Dr. Ho Tuong VINH

Table des matières

1	INTRODUCTION	3
2	Analyse du sujet	3
2.1	Définition des termes	3
2.2	Problématique	4
2.3	Objectifs	4
2.4	Travaux à réaliser	4
3	État de l'art	5
3.1	Problèmes posés par le développement de logiciel	5
3.2	Critères de choix d'un logiciel	5
3.3	Etapes de cycles vie d'un logiciel	6
3.4	Processus/méthodes de développement rapide d'application mobile	7
3.4.1	La méthode agile	8
3.4.2	La méthode waterfall ou cascade	9
3.4.3	La méthode en V	10
3.5	Outils de développement rapide d'application mobile	11
3.6	Comparaison	14
4	Solution proposee	15
4.1	Outil a utilisé	16
4.2	Pré-requis	16
5	Implémentation et test	17
5.1	Avant de commencer à coder	17
5.2	Fonctionnement de l'application	20
5.3	Les composants à utiliser	21

5.4	Design de l'application	21
5.5	Codage	22
5.6	Test	24
6	Conclusion et perspective	31
7	Références	32

1 INTRODUCTION

Le développement de logiciel est une activité complexe, qui est loin de se réduire à la programmation. Le développement de logiciels ou les programmes ont une certaine taille, nécessite d'adopter une méthode de développement qui permet d'assister une ou plusieurs étapes du cycle de vie de logiciel. Parmi les méthodes de développement, les approches objet, issues de la programmation objet, sont basées sur une modélisation du domaine d'application. Cette modélisation a pour but de faciliter la communication avec les utilisateurs, de réduire la complexité en proposant des vues à différents niveaux d'abstraction, de guider la construction du logiciel et de documenter les différents choix de conception.

Notre travail consiste a faire un survol sur sur les méthodes/processus de développement rapide d'applications mobiles ,puis adapter un processus selon notre choix et développer une application.

Notre travail est reparti en trois parties :

- L'analyse du sujet
- L'état de l'art
- La Solution proposée

2 Analyse du sujet

Dans cette première partie de notre rapport nous allons faire une analyse détaillée de notre sujet. Pour mieux cerner le travail nous allons d'abord partir d'une définition des mots clés,ensuite définir les objectifs à atteindre et enfin spécifier les travaux a réaliser.

2.1 Définition des termes

Les termes clés de notre sujet sont :

- Méthode/Processus :c'est un ensemble de phénomène actifs organisés dans le temps et produisant un résultat précis et mesurable.
- Application mobile :Une application mobile est un logiciel applicatif développé pour les appareils électronique mobiles tels que les téléphones portables, smartphones, tablettes ou encore certains ordinateurs fonctionnant avec le système d'exploitation Windows phone ou Chrome OS.

2.2 Problématique

Les problèmes aux quels nous nous sommes posés dans le but du développement rapide d'application mobile sont du par :La complexité des langages de programmation Le developpement prend beaucoup de temps beaucoup de demande de la part des clients Le coût élevé des applications

2.3 Objectifs

L'objectif final que nous recherchons à réaliser à travers ce TPE est de faire un survol sur les méthodes/processus de développement rapide d'applications mobiles ,puis adapter un processus selon certains critères à définir en fonction du choix que nous aurions fait.

2.4 Travaux à realiser

- Les travaux que nous aurons à réaliser au cour de travail sont entre autres :
- Définir l'état de l'art des techniques et outils de développement rapide d'application mobile

- Faire une synthèse des avantages et inconvénients des méthodes les plus pertinentes de l'état de l'art.
- Proposer un processus de développement rapide d'application mobile.
- Appliquer le processus proposé pour le développement d'une application mobile.

3 État de l'art

3.1 Problèmes posés par le développement de logiciel

En 1979, une étude du gouvernement américain, basée sur neuf projets logiciels, fait apparaître les symptômes suivants : beaucoup de logiciels ne sont pas livrés, pas utilisés ou abandonnés. Plus précisément, le coût se répartit de la façon suivante :

Logiciels	Coût
Payés, jamais livrés	3.2 M
Livrés, jamais utilisés	2.0 M
Abandonnés ou recommencés	1.3 M
Utilisés après modification	0.2 M
Utilisés en l'état	0.1 M

3.2 Critères de choix d'un logiciel

Un logiciel de qualité doit être :

- Fiable (conforme aux spécifications),
- Robuste (ne plante pas),
- Efficace (espace mémoire, temps d'exécution),

- Convivial (facile et agréable à utiliser),
- Documenté (accompagné d'un manuel utilisateur, d'un tutoriel ou d'un cours).

3.3 Etapes de cycles vie d'un logiciel

- **Analyse des besoins**

L'étape d'analyse et définition des besoins consiste à déterminer les attentes des futurs utilisateurs, par exemple avec un cahier des charges. Il faut décrire à la fois le système et l'environnement dans lequel le système sera exécuté. Cette étape comprend également une étude de faisabilité de la part des experts. C'est sur la base de cette étape que le contrat est signé.

- **Analyse et conception**

L'étape d'analyse et conception consiste à analyser, spécifier et effectuer les choix de conception du système. Cette étape comporte plusieurs sous-étapes

- **Mise en œuvre**

L'étape de mise en œuvre consiste à programmer le logiciel, en suivant les choix effectués lors de l'analyse et la conception.

- **Validation**

La validation consiste à s'assurer que le programme est de qualité. Il existe plusieurs techniques de validation :

- analyse statique (typage, conventions de programmation, détection d'erreurs pouvant survenir à l'exécution) ;
- preuve formelle (coûteuse, peu utilisée) ;
- revue de code (efficace) ;
- tests. Les tests constituent la principale méthode de validation. On distingue les tests unitaires, les tests d'intégration, les tests système et les tests

d'acceptation.

— **Évolution et maintenance**

L'étape d'évolution et de maintenance consiste à effectuer des modifications du logiciel après sa livraison. On distingue plusieurs types de maintenance :

- maintenance corrective (correction de défauts),
- maintenance évolutive (ajout de nouvelles fonctionnalités),
- maintenance adaptative (portage sur une nouvelle plate-forme).

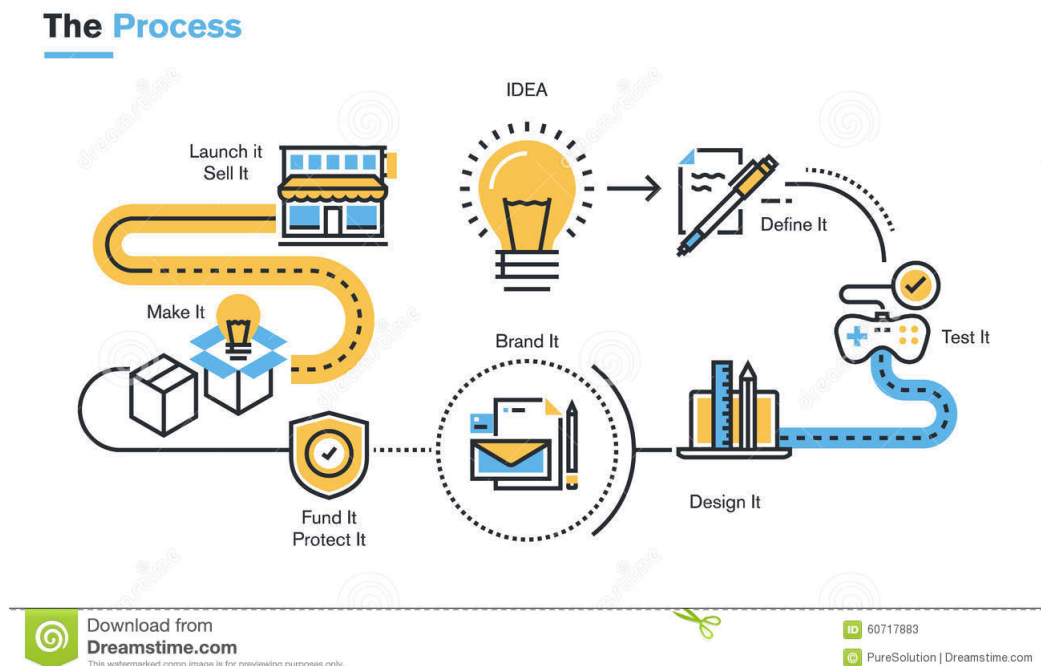


FIGURE 1 – Cycle de vie d'un logiciel

3.4 Processus/méthodes de développement rapide d'application mobile

Les modèles du cycle de vie du logiciel sont des « plans de travail » qui permettent de planifier le développement. Plus le logiciel à développer est complexe

(taille, algorithmes) et critique, plus il est important de bien contrôler le processus de développement et plus les documents qui accompagnent le logiciel doivent être précis et détaillés.

3.4.1 La méthode agile

La méthode agile est une méthode de gestion de projets et de développement. Il est très utilisée par les agences de marketing et les entreprises technologiques. La spécificité de la méthode agile est son approche par itération. Il propose un cycles de développement itératif, incrémental et adaptatif. Il favorise la collaboration entre des équipes pluridisciplinaires et l'implication des clients. L'objectif de la méthode agile est de pouvoir développer les produits plus rapidement à moindre coût, et avec un taux de réussite et de satisfaction plus importante.

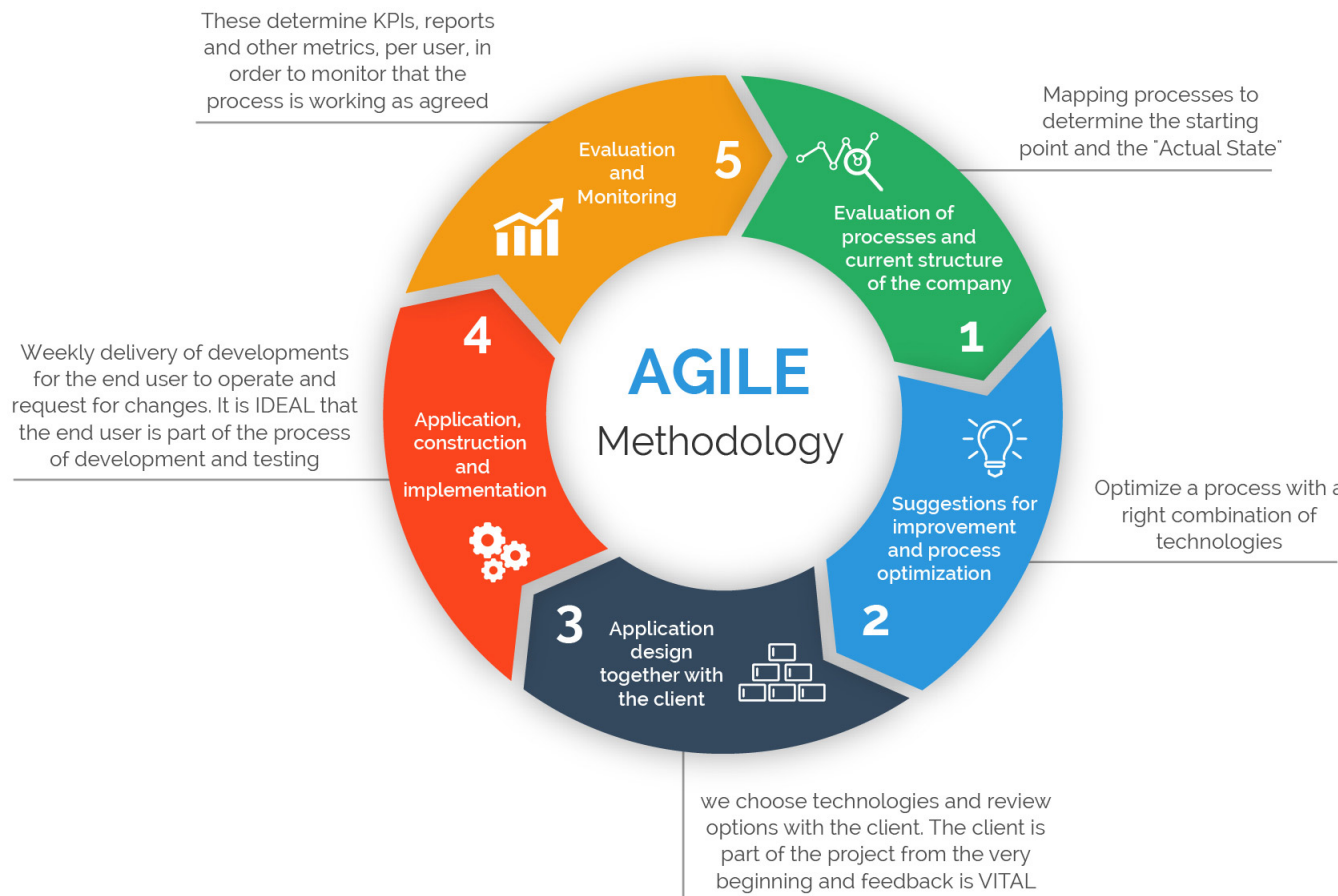


FIGURE 2 – Representation du modele agile

3.4.2 La méthode waterfall ou cascade

Waterfall est une méthode de gestion de projet traditionnelle. C'est la méthode la plus fréquemment utilisée. Elle consiste à suivre une série d'étapes prédéfinies jusqu'à atteindre l'objectif final. Simple, facile à mettre en place, logique et structurée, elle a pour objectif de créer le meilleur produit final possible. L'inconvénient de cette méthode est son manque de flexibilité à cause de son déroulement séquentiel. En effet, la méthode waterfall ne laisse aucune place aux changements et aux imprévus et il n'y a pas d'interaction possible avec les clients ce qui cause

de risques de déception du client lors de la livraison du produit.

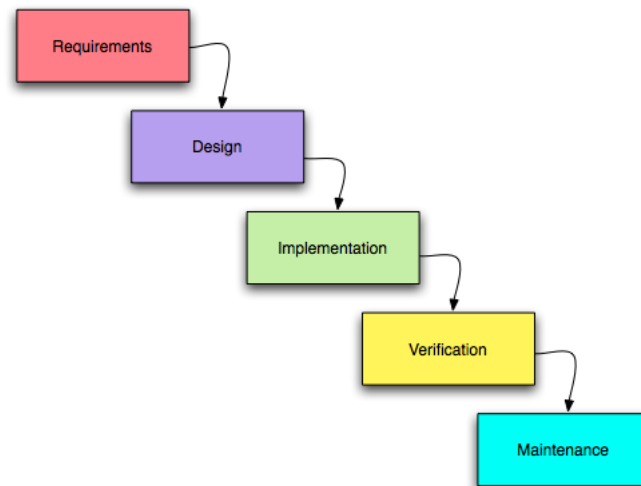


FIGURE 3 – Representation du modele en cascade

3.4.3 La méthode en V

Le modèle du cycle en V a été imaginé pour pallier le problème de réactivité du modèle en cascade. Ce modèle est une amélioration du modèle en cascade qui permet en cas d'anomalie, de limiter un retour aux étapes précédentes. Les phases de la partie montante doivent renvoyer de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés afin d'améliorer le logiciel.

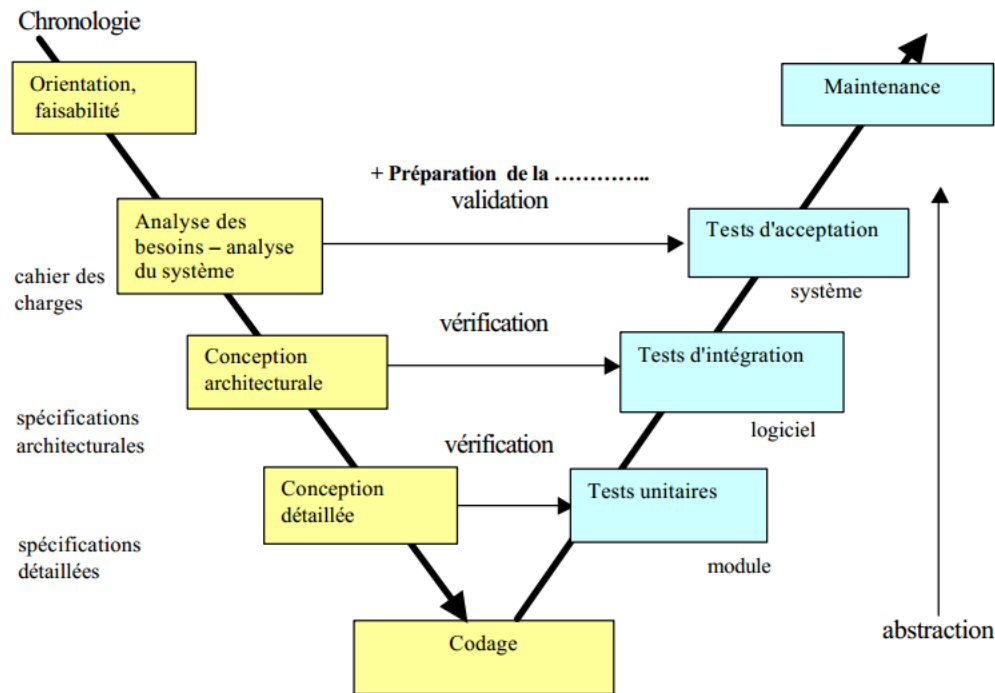


FIGURE 4 – Représentation du modèle en V

3.5 Outils de développement rapide d'application mobile

Le marché des applications mobiles est en plein boom et il devient nécessaire de développer les applications rapidement. Malgré une décennie de progrès, toutefois, il reste trop chronophage, difficile ou onéreux pour beaucoup de créer une application mobile native à partir de zéro. Les outils de développement rapide d'applications mobiles (RMAD) visent à répondre à ce problème. Ils embarquent un environnement de développement intégré WYSIWYG par glisser-déposer pour créer des applications sans avoir à toucher au code natif.

Selon GARTNER, une entreprise américaine de conseil et de recherche dans le domaine des techniques avancées «les outils de développement rapide d'applica-

tions mobiles (RMAD) constituent une classe d'outils de développement hautement productifs qui permettent aux non- programmeurs et dans certains cas, aux développeurs sans expérience mobile de créer des applications mobiles utilisant peu ou sans code.”

Gartner explique ensuite que les outils RMAD contribueront à la démocratisation du développement d'applications. Ces outils permettront aux services informatiques de fournir à l'ensemble de leur organisation une plate-forme permettant aux développeurs de créer des applications mobiles, tout en maintenant le contrôle des données et de la sécurité. Gartner prédit : «D'ici 2022, les citoyens développeurs créeront plus du tiers des applications Web et mobiles.» Il existe plusieurs outils de RMAD dont nous allons présenter quelques uns :

●Google Appmaker

C'est un logiciel permettant de créer des applications sans avoir besoin de coder, grâce à un système de glisser-déposer. Cet outil vise à permettre à tout un chacun, et notamment aux entreprises, de créer des applications mobiles basiques en toute simplicité.

●App Inventor

C'est une plateforme développée par Google Labs qui fournit un environnement de développement totalement visuel. Avec App Inventor, on n'a pas besoin d'écrire ni une seule ligne de code pour créer une application Android. il est gratuit et il suffit juste de le télécharger pour commencer à créer une application Android gratuitement.

●SWIFTIC

Fondé en Israël en 2010, Swiftic revendique la création de plus d'un million d'applications dans le monde, ce qui en fait l'un des grands acteurs du marché des applications. L'outil propose un large éventail de blocs à assembler : gestion de cartes de fidélité, prise de rendez-vous, e-commerce, avis utilisateurs et gestion d'événements, etc. La plupart des exemples mis en avant dans la galerie d'applications concernent des restaurants, des groupes de musique et autres structures organisant des événements.

●AppInstitute

AppInstitute est un générateur d'application mobile pratique doté d'un éditeur drag and drop, qui peut s'adapter à toutes sortes d'activités. Pour générer une application sans développeurs ni connaissances technique.

●Appyet

Appyet est un outil gratuit et idéal pour créer une application native pour Android en 5 minutes top chrono. Facile d'utilisation, il ne nécessite pas de connaissance spécifique en programmation. De plus, vous avez la possibilité de monétiser son application mobile grâce à des bandeaux publicitaires.

●Outsystem

OutSystems est une plate-forme de développement low-code qui permet de développer rapidement des application et à apporter une valeur réelle à votre entreprise.

Il permet de développer visuellement vos applications, en les intégrant facilement à n'importe quel système. Il est possible d'ajouter votre propre code personnalisé lorsque vous en avez besoin.

3.6 Comparaison

Methode	Avantages	Inconvenients
Agile	<ul style="list-style-type: none"> -Utilisé par plus 71/100 des entreprises -Favorise la communication entre les équipes et le clients -Flexibilité de gestion de changement. <p>Approche mixte ,Itérative et incrémentale</p>	<ul style="list-style-type: none"> -Refactoring
Cascade	<ul style="list-style-type: none"> -Simple et facile à mettre en place -le planning est établi à l'avance 	<ul style="list-style-type: none"> -Manque de flexibilité -pas d'interaction possible avec les clients -faible tolérance à l'erreur
V	<ul style="list-style-type: none"> -Evite le chevauchement des étapes -limite les retours aux étapes précédentes 	<ul style="list-style-type: none"> -Difficile de modifier les étapes
Spiral	<ul style="list-style-type: none"> -Risques sont limités <p>-adaptatif</p>	<ul style="list-style-type: none"> -adapté aux grands entreprises et peut engendrer plus des sucôts d'analyse par rapport à la réalisation du logiciel.

En analysant le tableau nous remarquons que toute methode à ses avantages et ses inconvenients. Parmi ces methodes, la methode agile est la plus utilisee car elle s'adapte à tout type de projet par sa flexibilité et son approche mixte c'est pour

cela que c'est la methode la plus utilisée dans les entreprises pour les gestions de projet.

A la difference de la vieille methode de developpement qui est la programmation base sur les languages comme C++, Java, plusieurs amelioration ont ete fait pour simplifier cette methode qui est en plus d'etre complique nous prend beaucoup de temps pour le developpement. Pour facilite la tache du developpeur des langauges plus simplifies,des librairies, des frameworks et des outils ont developpes pour repondre a cette attente.

Notre TPE se situe dans le cadre des outils de developpement rapide. Les avantages de l'utilisation de ces outils sont multiples et nous pouvons citer entre autres :

- programmation événementielle,
- La fusion rapide de sources et de data multiples grâce à des technologies d'intégration légères,
- L'amélioration du temps de réponse du hardware et des systèmes d'exploitation,
- La simplification de la conception des moteurs d'exécution grâce aux standards comme le HTML5, JavaScript,
- De meilleures interfaces, conçues plus rapidement grâce à l'utilisation du HTML5 et la multiplication des librairies de composants.

4 Solution proposee

La solution que nous proposons à la suite de ces travaux est une application mobile que nous avons nommé Feedback Course. Cette application va permettre aux étudiants de donner leur appriaciation sur un cours donne et de lui permettre de faire donner des suggestions ou des remarques qui vont permettre aux responsables et aux corps enseignants de savoir les critiques faites par les étudiants.

4.1 Outil a utilisé

Pour réaliser cette application nous avons utilisé App Inventor. App Inventor est un programme en ligne qui apprend aux utilisateurs à créer des programmes pour applications. Il s'ouvre dans votre navigateur Web pour un accès facile. Le programme est open source, ce qui signifie que vous pouvez l'utiliser complètement gratuitement. Tout ce que vous fabriquez avec le programme vous appartient également à conserver et à utiliser à votre guise.

4.2 Pré-requis

L'utilisation de MIT App Inventor est soumise à quelques exigences techniques, mais il ne devrait pas y avoir quelque chose que vous n'avez pas déjà sur votre ordinateur. Les exigences incluent :

- Une connexion internet. Le programme s'exécute dans votre navigateur, vous devez donc être en ligne pour l'utiliser. Une connexion Internet stable est donc nécessaire afin que votre progression enregistre sans problème.
- App Inventor fonctionne mieux avec les dernières versions de Chrome ou de Firefox pour Windows. Il ne fonctionne pas sur Internet Explorer. Pour les utilisateurs de Mac, vous devriez utiliser la dernière version de Safari.
- Assurez-vous que votre ordinateur dispose de la dernière mise à jour Java.
- Disposez-vous d'un téléphone ou tablette Android (facultatif). Vous pouvez tester votre nouvelle application directement sur votre téléphone ou votre tablette sous Android. Si vous ne souhaitez pas configurer votre appareil avec le programme,

vous pouvez utiliser l'émulateur Android inclus directement sur le site Web.

• Il faut avoir un compte Google. App Inventor vous connecte à Google avant de commencer à coder. Le compte permettra d'enregistrer vos différents projets sur Google cloud pendant que vous travaillez. Si vous ne disposez pas de compte Google le programme App Inventor vous guidera à travers les étapes pour créer un compte, puis vous ramène directement sur le site Web pour commencer à coder.

5 Implémentation et test

5.1 Avant de commencer à coder

Avant de vous lancer dans le processus de codage, préparez votre ordinateur à commencer à faciliter les choses plus tard :

- Comme mentionner dans la partie pré-requis, il faut créer un compte Google si vous ne l'avez pas déjà fait.
- Mettre à jour tous les programmes requis.
- Connectez et testez votre tablette ou votre téléphone avec le programme en procédant comme suit :

Connectez votre téléphone ou votre tablette via Wifi.

Vous pouvez utiliser App Inventor sans rien télécharger sur votre ordinateur ! Vous développerez des applications sur notre site Web : ai2.appinventor.mit.edu. Pour effectuer des tests en direct sur votre appareil Android, il suffit d'installer l'application MIT App Inventor Companion sur votre téléphone ou tablette Android. Une fois le compagnon installé, vous pouvez ouvrir des projets dans App Inventor sur le Web, ouvrir le compagnon sur votre appareil et tester vos applications au fur et à mesure de leur création :

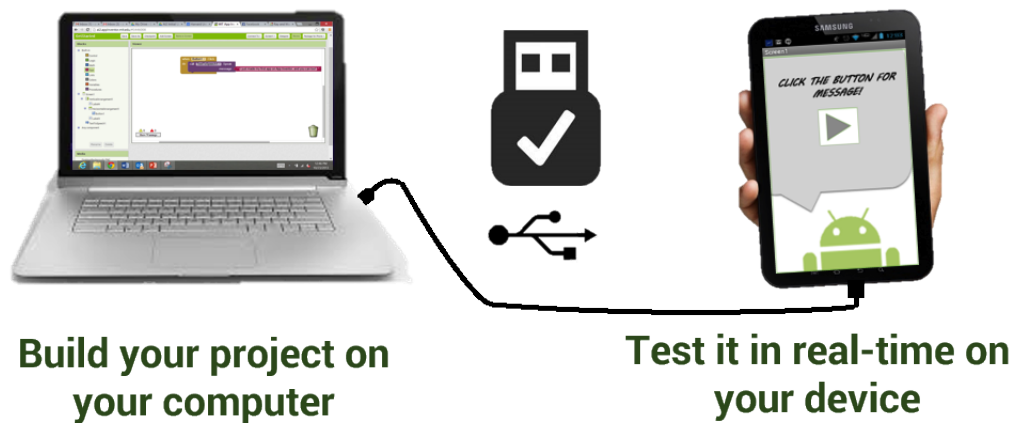


FIGURE 5 – Connection au wifi

Les étapes suivantes vous guideront tout au long du processus :

Etape 1 :Téléchargez et installez l'application MIT AI2 Companion sur votre téléphone.

Une fois le téléchargement terminé, suivez les instructions pour installer l'application Companion sur votre appareil. Vous devez installer le MIT AI2 Companion une seule fois, puis le laisser sur votre téléphone ou votre tablette chaque fois que vous utilisez App Inventor.

Etape 2 :Connectez votre ordinateur et votre appareil au même réseau WiFi

App Inventor vous montrera automatiquement l'application que vous créez, mais uniquement si votre ordinateur (exécutant App Inventor) et votre appareil Android (exécutant Companion) sont connectés au même réseau WiFi.

Etape 3 :Ouvrez un projet App Inventor et connectez-le à votre appareil.

Accédez à App Inventor et ouvrez un projet (ou créez-en un nouveau - utilisez

Projet, Démarrer un nouveau projet et nommez votre projet).

Puis choisissez "Connect" et "AI Companion" dans le menu supérieur du navigateur AI2 :

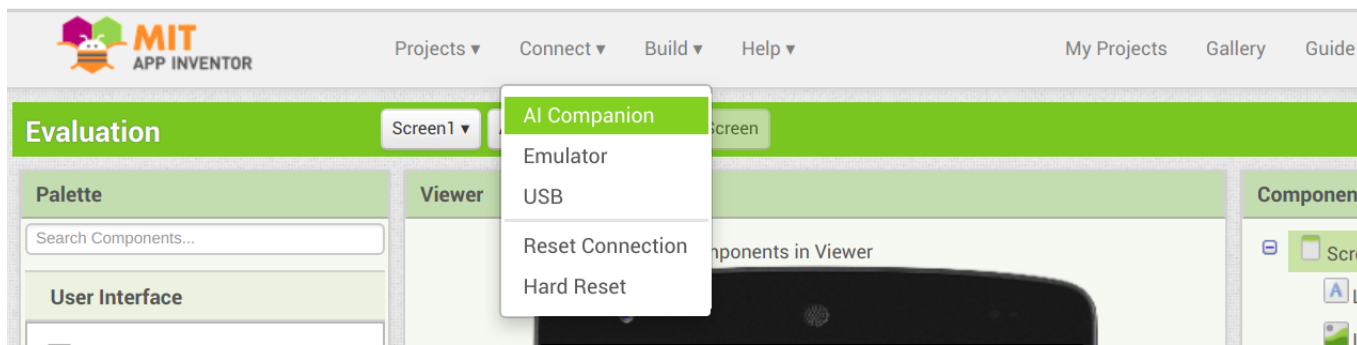


FIGURE 6 – Connection

Une boîte de dialogue avec un code QR apparaîtra sur l'écran de votre PC. Sur votre appareil, lancez l'application MIT App Companion comme n'importe quelle autre application. Cliquez ensuite sur le bouton «Scan QR code» sur Companion et scannez le code dans la fenêtre App Inventor :

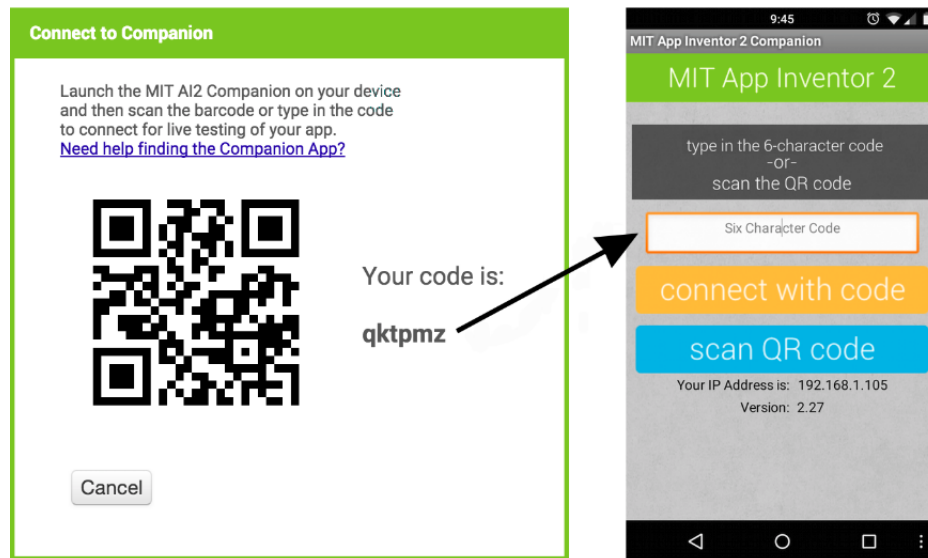


FIGURE 7 – Scan Qr code via le telephone

Dans quelques secondes, vous verrez l'application que vous construisez sur votre appareil. Il se mettra à jour au mesure au fure et à mesure que vous modifiez votre conception et vos blocs, une fonctionnalité appelée «test en direct».

5.2 Fonctionnement de l'application

Le processus est simple mais crée des résultats compliqués. Tout ce que vous avez à faire est de glisser-déposer des boîtes.

Chaque case place un certain élément dans votre programme d'application. En connectant les boîtes, vous pouvez ajouter plus d'éléments et en modifier des parties. Pensez-y comme à un casse-tête avec une image d'une maison. Chaque pièce représente à elle seule une partie de l'image finale. Un tas de morceaux ensemble va créer la maison alors que d'autres créent l'arrière-plan. Lorsque toutes les pièces sont placées ensemble, vous voyez l'image complète. Si une pièce n'est pas à la place, l'image finale semble déformée.

5.3 Les composants à utiliser

Une fois le design nous allons identifier les différents composants à utiliser. Dans le cas de notre application les composants à utiliser sont :

- Label : pour des champs de texte,
- Button : permet de faire des actions(soumettre ,supprimer,retour)lorsqu'on click sur le button,
- Textbox : permet de saisir des informations comme le nom ,email, etc...,
- ListPicker : permet de créer des listes,
- Notifier1 : permet d'afficher des notifications,
- TinyDB1 : permet d'enregistrer des informations,
- FusiontablesControl1 : une base de données en ligne qui servira de stockage de nos données.

5.4 Design de l'application

Avant de passer au codage vous devriez faire un design de votre application sur un bout de papier qui va donner l'apparence de l'application à développer. Une fois le design terminé nous allons l'appliquer sur App Inventor dans la partie design en plaçant les différents composants par le principe de glisser-déposer.

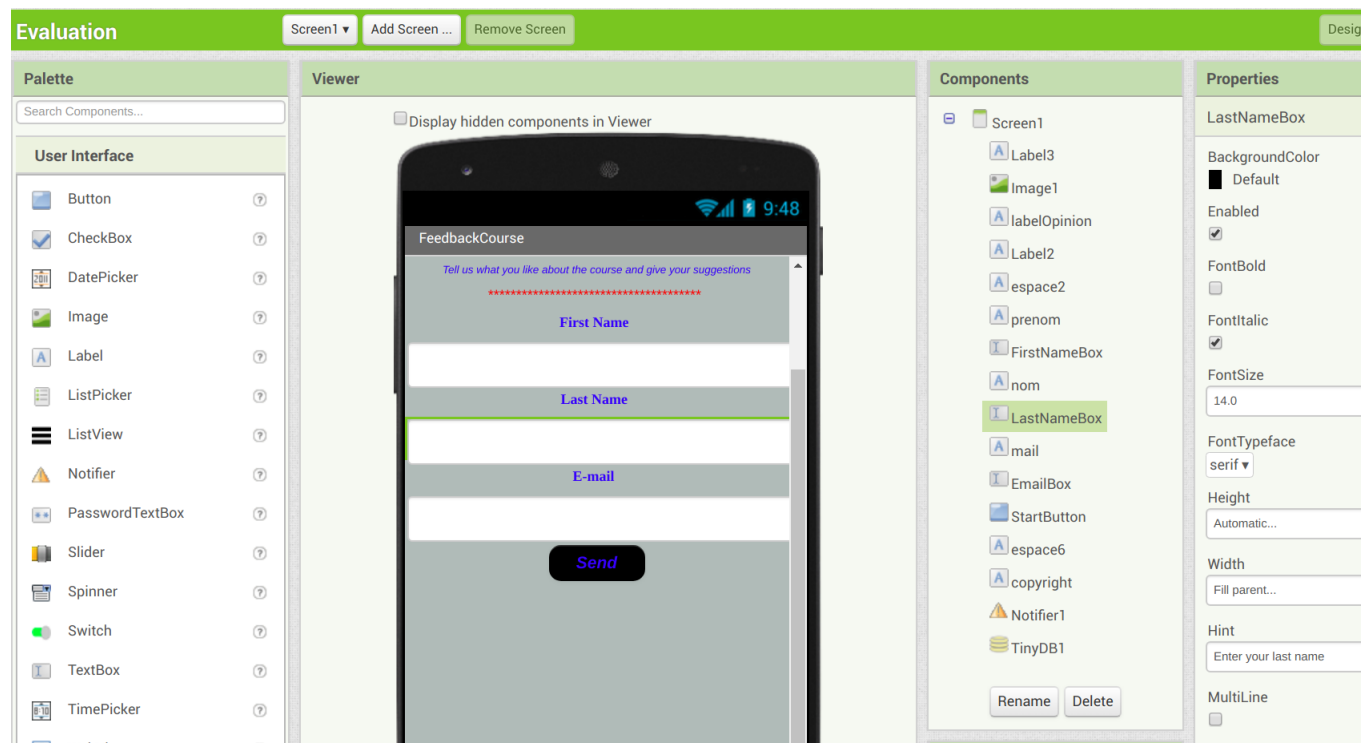


FIGURE 8 – Design de l'application

5.5 Codage

Nous sommes maintenant à la partie la plus attendue qui est le code. Sur l'interface de App Inventor se trouve une partie appeler block ou va se dérouler le codage du comportement de notre application. Pour comprendre, si nous observons la première interface de l'application ci-dessous, nous avons des Text-box(nom,prenom,email) et un bouton send.

FeedbackCourse

INTERNATIONAL FRANCOPHONE INSTITUTE

INSTITUT FRANCOPHONE INTERNATIONAL

FeedbackCourse

Tell us what you like about the course and give your suggestions

First Name

Last Name

E-mail

Send

FIGURE 9 – Interface de l'application

Au click du bouton nous allons recuperer les informations saisi par l'utilisateur puis le soumettre ; mais avant nous realiser des controls qui va obliger l'utilisateur à remplir toute les champs si non l'envoi est refusé et affiche une notification pour indiquer de completer les autres champs. Si tous les informations sont saisis et que l'email ne respecte pas la norme l'envoi est automatique refusé aussi. On traduit tout cela par le code ci-dessous :

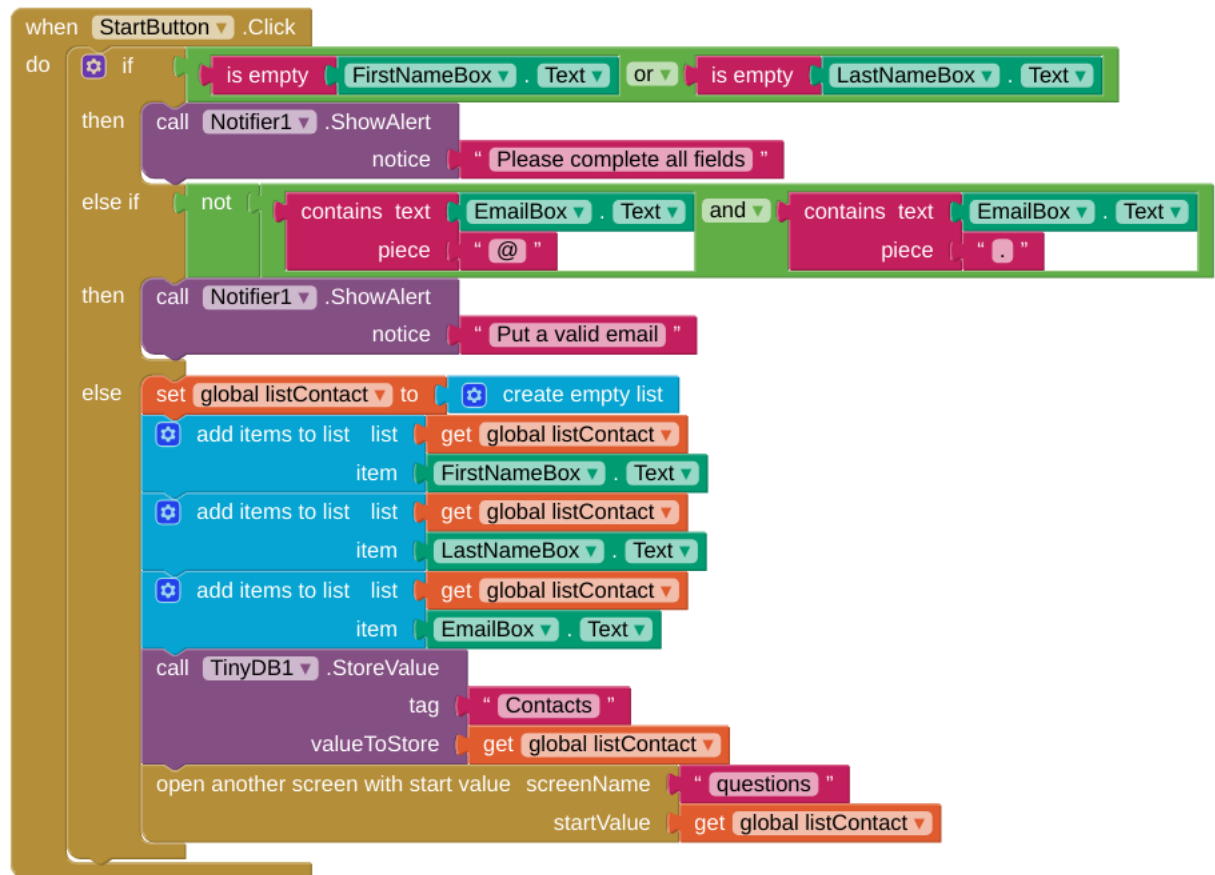


FIGURE 10 – Code

5.6 Test

Après avoir réaliser tous les différents étapes de la partie block nous allons réaliser le test de notre application en l'installant sur notre téléphone portable. Notons que pour l'étape de l'installation nous deux manière de proceder :

-La 1ère option consiste à installer l'application via un Qr code qui est valable uniquement pour 2h c'est à qu'en installant l'application via le Qr code apres 2h l'application ne sera plus accessible à moins de le réinstaller à nouveau.

Dans Build, App(provide Qr code .apk) nous generons le Qr code.

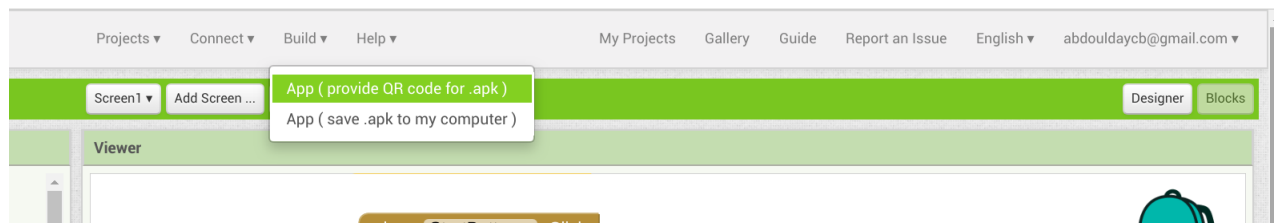


FIGURE 11 – Generation du Qr code

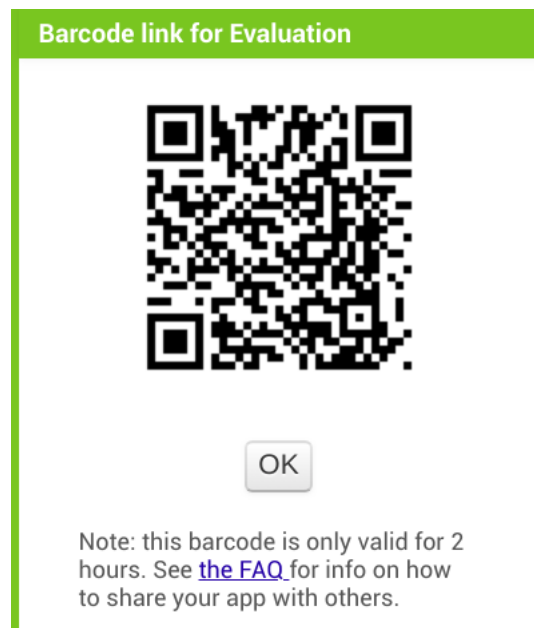


FIGURE 12 – Qr code

-La 2ème option consiste à installer l'application de façon définitive. Dans Build, App(save .apk to my computer) pour télécharger le fichier .apk

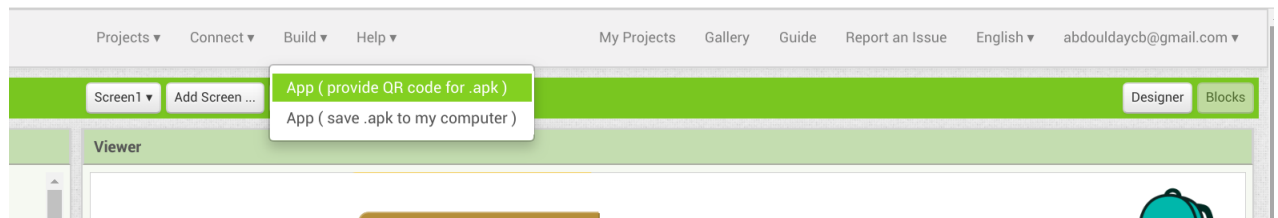
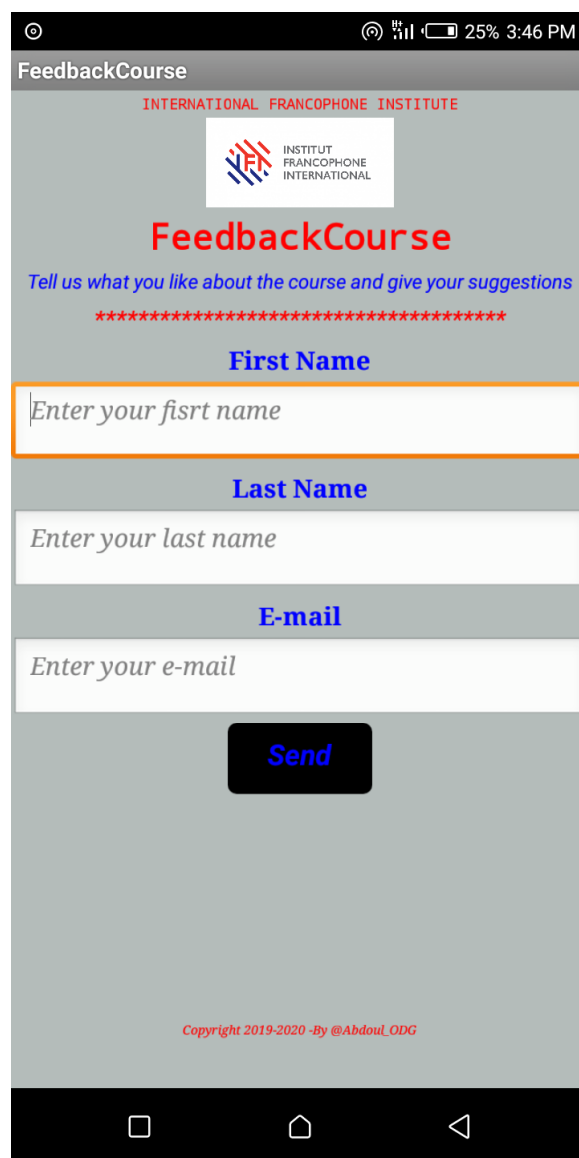


FIGURE 13 – Telechargement de l'application

Maintenant que notre application est installer nous passons au test. Une fois l'application lancee nous obtenons la figure ci-dessous.



FeedbackCourse

INTERNATIONAL FRANCOPHONE INSTITUTE

INSTITUT FRANCOPHONE INTERNATIONAL

FeedbackCourse

Tell us what you like about the course and give your suggestions

First Name

Enter your first name

Last Name

Enter your last name

E-mail

Enter your e-mail

Send

Copyright 2019-2020 -By @Abdoul_ODG

FIGURE 14 – Première interface de l'application

Nous nous renseignons sur tous les champs puis cliquons sur **send** ; nous serons dirigés sur une autre page (figure ci-dessous)

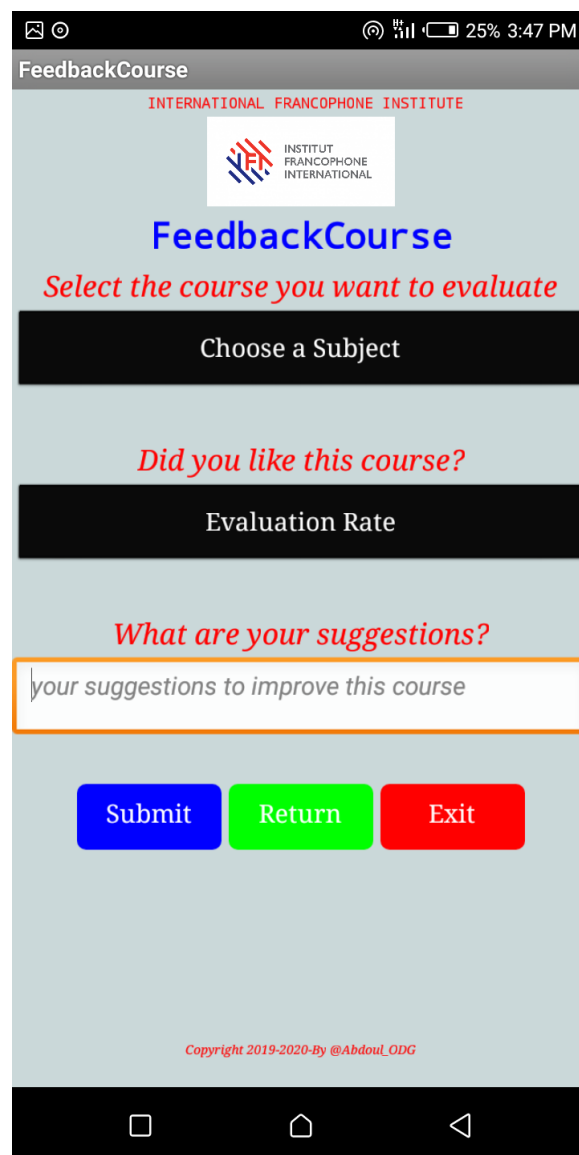


FIGURE 15 – Second interface de l'application

Dans cette partie aussi nous allons choisir la matiere que nous voulons evaluer ,ensuite nous donnons notre appriciation puis donner des suggestions ou des re-marques et enfin nous cliquons sur **submit** pour envoyer.



FIGURE 16 – Choix d'une matière

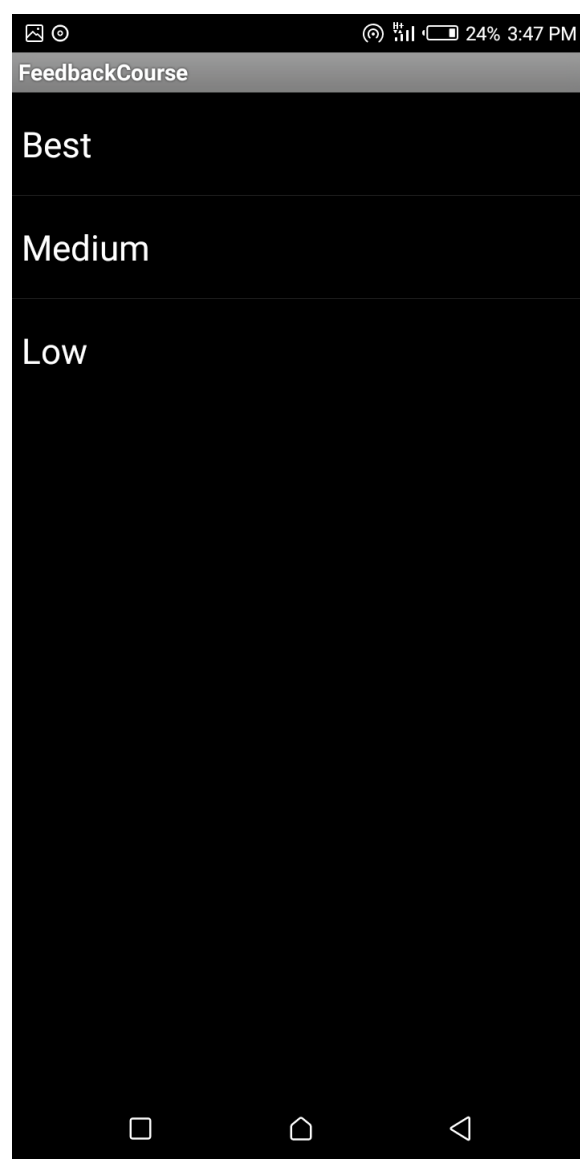


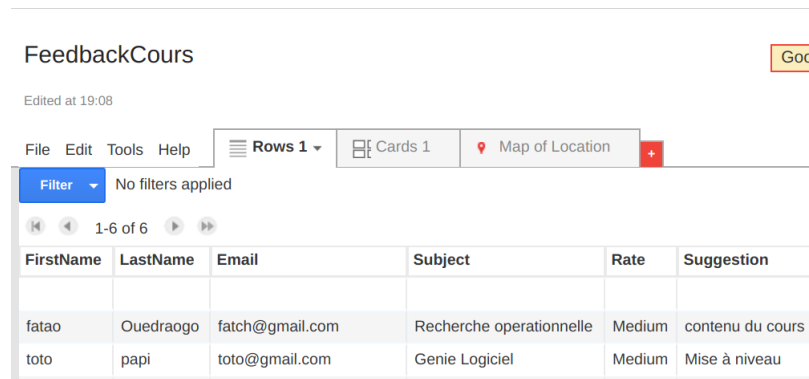
FIGURE 17 – Evaluation

The screenshot shows the FeedbackCourse mobile application interface. At the top, the status bar displays signal strength, Wi-Fi, and battery level at 24% at 3:48 PM. The app header includes the title 'FeedbackCourse' and the 'INTERNATIONAL FRANCOPHONE INSTITUTE' logo. The main content area features the text 'FeedbackCourse' in blue, followed by the instruction 'Select the course you want to evaluate' in red. A black button displays 'Choose a Subject: Genie Logiciel'. Below this, the text 'Did you like this course?' is shown in red, followed by a black button with 'Evaluation Rate: Medium'. The next section asks 'What are your suggestions?' in red, with a text input field containing 'Mise à niveau'. At the bottom, there are three buttons: 'Submit' (blue), 'Return' (green), and 'Exit' (red). A copyright notice 'Copyright 2019-2020-By @Abdou_ODG' is visible at the very bottom of the app screen.

FIGURE 18 – Suggestion

Les informations envoyées par l'utilisateur seront enregistrées sur une base de données en ligne que seul le responsable des études dans notre cas pourra voir ces informations.

La figure ci-dessous montre les données envoyées par des utilisateurs.



FeedbackCours

Edited at 19:08

File Edit Tools Help Rows 1 Cards 1 Map of Location

Filter No filters applied

1-6 of 6

FirstName	LastName	Email	Subject	Rate	Suggestion
fatao	Ouedraogo	fatch@gmail.com	Recherche operationnelle	Medium	contenu du cours
toto	papi	toto@gmail.com	Genie Logiciel	Medium	Mise à niveau

FIGURE 19 – base de donnée

6 Conclusion et perspective

Au terme de notre travail nous avons fait d'abord l'analyse du sujet qui nous a permis d'identifier la problématique de notre sujet et de fixer les objectifs et résultats attendus. Ensuite, nous avons fait l'état de l'art qui nous a permis d'avoir une large compréhension de notre sujet. Nous avons ainsi pu étudier certains articles en rapport avec notre sujet. Enfin, nous avons proposé une solution à notre problème.

Dans l'ensemble nous pouvons dire que le travail que nous eu à mener nous a permis d'obtenir des résultats à travers la solution proposée à la seule limite que notre application ne fonctionne pas sur toutes les plate-formes; mais tout cela explique par le fait que nous avons rencontré beaucoup de difficultés pour le choix de l'outil. Nous travaillons donc pour l'amélioration de notre solution.

Perspective

L'outil App Inventor que nous avons utilisé pour développer notre application ne supporte pas pour le moment la plate-forme iOS donc nous prévoyons utiliser un autre outil qui nous permettra de déployer notre application sur toutes les plate-formes.

7 Références

- [1] Development methodologies of mobile applications by Veronika Vesela and Michal Krbecek
- [2] Software Development Processes for Mobile Systems :Is Agile Really Taking Over the Business ? by Luis Corral, Alberto Sillitti, Giancarlo Succi
- [3] Reviews on Agile Methods in Mobile Application Development Process
- [4] 2017 Gartner Market Guide for Rapid Mobile App Development Tools (<https://dzone.com/article/gartner-market-guide-for-rapid-mobile-app-dev>)
- [5] 25 simple tools for building mobileapps (<https://www.infoworld.com/article/3214637/mobile-development/25simple-tools-for-building-mobile-apps-fast.html>)
- <https://www.appstud.com/fr/guides/agence-mobile/app095/>
- [7] <https://www.journaldunet.com/developpeur/expert/68336/les-langages-de-developpement-qui-ont-la-cote-en-2018.shtml>
- [8] <https://www.developpez.com/actu/185087/Quels-sont-les-langages-de-programmation-les-plus-utilises-par-les-developpeurs-Une-analyse-des-evenements-publics-sur-GitHub/>
- [9] <http://bemobee.com/developpement-dapplication-mobile-native-ou-hybride/>
- [10] <https://fr.yeeply.com/blog/quelle-plateforme-choisir-creer-son-application-mobile/>