

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et
de la Recherche scientifique

Université Akli Mohand Oulhadj

- Bouira-

Tasdawit Akli Muhed Ulhag - Tubirett -

Faculté des Sciences et des Sciences
Appliquées



وزارة التعليم العالي والبحث العلمي

جامعة أكلي محند أولحاج

- البويرة -

كلية العلوم والعلوم التطبيقية

المرجع: م/م/..... م/م/2022

Mémoire de Master

Présenté au

Département : Génie Électrique

Domaine : Sciences et Technologies

Filière: Electronique

Spécialité: Electronique des Systèmes Embarqués

Réalisé par :

DERRADJI LYES

Et

BENTOUTAH SARAH

Thème

**Etude et réalisation d'une maquette 3D de la FSSA pour
le diagnostic et le suivi**

Soutenu le: 15/09/2022

Devant la commission composée de :

M ^r :	REZKI MOHAMED	M.C.A	Univ.Bouira	Président
M ^d :	LATRECHE SADJIA	M.C.B	Univ. Bouira	Examineur
M ^r :	NOURINE MOURAD	M.C.A	Univ.Bouira	Rapporteur

Année universitaire 2021/2022

Dédicace:

Grâce à l'aide et le succès de Dieu, ce modeste travail a été achevé.

A mes chers parents. Pour tout ce que vous avez fait pour moi, Vous m'avez préparé au monde et vous m'en avez ouvert les Portes, et c'est avec émotion que je vous exprime toute mon affection.

Merci beaucoup mes chers, que Dieu vous garde toujours en bonne santé.

A l'âme de mon grand-père, que Dieu le bénisse au paradis

A mes chers frères « Farouk et Abdou ».

A ma très chère sœur, ma moitié « Hadjer ».

A la personne qui illumine ma vie a mon cher mari "Samir"

et toute ma belle famille.

A mes chères amies, que j'aime trop fort ; « Dhaouia », « Sabrina , Nor, Sarah, Amina , Sorrow et mes chers sœurs UGEL 10 ».

A toute ma famille.

A mon binôme « Derradji Lyes » et toute sa famille.

A notre cher ingénieur notre frère MS khaoui Farid et sa collègue qui nous a aidé à réussir ce modeste travail

A tous ceux dont j'ai oublié de mentionner les noms.

Toute la promo de Electronique des Systèmes Embarquées 2021/2022.

Sarah

Dédicace:

Je dédie ce modeste travail en signe de respect et d'amour :

Je le dédie :

A mes parents qui m'ont soutenu dans la vie et qui m'ont accompagnée

Dans mon parcours.

A ceux qui m'ont soutenu dans les étapes les plus importantes

A toute ma famille, sans exception.

A ma binôme « Sarah Bentoutah » et à toute sa famille

À mes amis : Salim , Samir, Elyes, Idhir, karim, Abdlah, Walid, Houcine, Said.

A tous ceux dont j'ai oublié de mentionner les noms.

Toute la promo de Electronique des Systèmes Embarquées 2021/2022.

Lyes

Remerciements

Nous tenons à remercier Allah le tout puissant qui nous a accordé santé et courage pour mener ce travail jusqu'à son terme.

*Nous tenons à remercier également notre promoteur Mr. **NOURINE Mourad** qui a accepté de nous encadrer et qui nous a toujours guidés dans la réalisation de ce mémoire.*

Sincères remerciement aux membres de jury.

Nous remerciant tous notre enseignants, nous tiens à leur exprimer notre reconnaissance pour avoir accompagné tout au long de notre formation.

Enfin, nous remerciant chaleureusement toute personne ayant contribué de près ou de loin pour que ce travail puisse être réalisé et mené à terme.



Liste des figures:

Figure 1: Logo du système Android	5
Figure 2: Logo du système IOS	5
Figure 3: Logo du système Windows mobile	6
Figure 4: Logo du système BlackBerry	6
Figure 5: Maquette de conception.....	11
Figure 6: Maquette de présentation	11
Figure 7: Maquette de planification urbaine	12
Figure 8: Maquette de paysage	12
Figure 9: Maquette de bâtiment de bureau	13
Figure 10: Maquette de bâtiments résidentiels	13
Figure 11: Maquette d'usine	14
Figure 12: Maquette d'intérieur	14
Figure 13: Logo de la plateforme Android studio	17
Figure 14: Logo de développeur Flutter	18
Figure 15: Fenêtre de l'IDE Android studio	18
Figure 16: Fenêtre de l'IDE Android studio	18
Figure 17: Fenêtre de l'IDE Android studio	19
Figure 18: Fenêtre de la citation du nom de projet	19
Figure 19: La structure du projet.	19
Figure 20: Fenêtre du fichier main.dart	20
Figure 21: Exemple d'un code d'exécution	21
Figure 22: La barre d'outils de Flutter	21
Figure 23: Fenêtre de l'AVD Manager	22
Figure 24: L'Emulateur d'Android.....	22
Figure 25: La barre d'outils d'Android studio.....	22
Figure 26: Fenêtre du résultat final	22
Figure 27: Organigramme de création de notre application.	23
Figure 28: Exemple des bibliothèques.....	24
Figure 29: Exemple d'une classe créée	24
Figure 30: Exemple de la sous-classe créée.....	24
Figure 31: Login Admin.	25
Figure 32: Représentation de la base de données	25
Figure 33 : Plan d'un bloc de La Fssa.....	26
Figure 34: Modélisation du plan.....	26
Figure 35: Logo de langage Dart	28
Figure 36: Dart et flutter	28
Figure 37: Structure de l'utilisation de l'application par l'administrateur et l'agent.	29
Figure 38: Les interfaces de l'application.....	30
Figure 39: Interface d'accueil d'administrateur	31
Figure 40: Interface d'utilisateur Agent (Ajouté des problèmes).....	33
Figure 41: Interface agent et listes des problèmes.	34
Figure 42: Modélisation de la maquette.....	35
Figure 43: Résultat final et vision sur la maquette de la faculté	36

Liste des tableaux:

Tableau I.1: Le fonctionnement de chaque type d'application	12
Tableau I.2 : Comparaison entre les systèmes d'exploitations mobiles.....	16
Tableau I.3 : Historique de l'Android et ses différentes versions.....	19

Liste des abréviations:

ABD	Android Debug Bridge
API	Application Programming Interface
AVD	Appareil Virtuel Android
BIM	Building Information Modeling
CSS	Cascading Style Sheets
DAO	Dessin Assisté par Ordinateur
FSSA	Faculté des Sciences et des Sciences Appliqués
JDE	Environnement de développement Java
Html	Hyper Text Markup Language
IDE	Integrated Development Environment
IFC	Industry Foundation Classes
IOS	InterNetwork Operating System
OS	Operating System
RIM	Research In Motion

Sommaire

Introduction générale :	1
Chapitre I : Généralités sur les applications mobiles et les maquettes	2
I.1. Introduction :	2
I.2 Les applications mobiles :	2
I.2.2. Types d'applications mobiles :	2
I.2.3. Fonctionnalité de chaque type d'application :	4
I.2.4. Caractéristiques des applications mobiles :	4
I.3. Les systèmes d'exploitation mobiles OS :	5
I.3.1. Définition d'un système d'exploitation :	5
I.3.2. Les principaux systèmes d'exploitation mobiles :	5
I.3.3. Comparaison entre les systèmes d'exploitation :	6
I.4. Les outils de développement mobile :	7
I.5. Le système d'exploitation Android :	8
I.5.1. Le choix de la plateforme Android:	8
I.5.2. La naissance d'Android :	8
I.5.3. Historique des versions d'Android :	9
I.5.4. Le kit de développement (SDK) :	10
I.6. Notions sur les maquettes :	10
I.6.1. Définition d'une maquette :	10
I.6.2. Définition de la maquette d'architecture :	10
I.6.3. Critères de classification des maquettes d'architectures :	10
I.6.4. Classification des maquettes d'architectures par utilisation :	11
I.6.5. L'objectif de la maquette 3D :	14
I.7. La maquette numérique BIM :	14
Définition de BIM :	14
I.8. Exploitation des données 3D :	15
I.8.1. Les logiciels de modélisation 3D :	15
I.8.1.1. AutoCad Architecture :	15
I.8.2. Choix d'un logiciel de modélisation :	15
I.9. Conclusion :	16
Chapitre II: outils et logiciels utilisés	17
II.1. Introduction :	17

II.2. Android Studio :	17
II.3. Les étapes principales de création d'application Flutter sur Android studio :	18
II.3.1 : Définition du Flutter :	18
II.3.2. Création de l'application Flutter :	18
II.3.3. L'explication de la structure de projet :	19
II.3.4. Ecriture le code pour l'application :	20
II.3.5. Lancement de l'application :	21
II.5. L'organigramme de création de notre application:	23
II.6. Code de programmation :	24
II.6.1 Exemple d'Activité du notre code :	25
II.7. La modélisation numérique de la maquette :	26
II.7.1. Définition de la modélisation numérique:	26
II.7.2. la modélisation de notre maquette (de la FSSA):	26
II.8. Conclusion :	26
Chapitre III : développement de l'application.	28
III.1. Introduction :	28
III.3. Langages de développement de notre application :	28
III.3.1. Dart :	28
III.3.2. Flutter et Dart :	28
III.4. Organigramme d'utilisation de l'application:	29
III.4. Capture d'écrans des activités de l'application :	30
III.4.1. Les interfaces générales de notre application :	30
III.4.2. L'interface de superviseur (admis) :	31
III.4.4. Interface agent (liste des problèmes) :	34
III.5. Modélisation de la maquette :	34
III.6. Résultat finale et vision sur la maquette de la faculté :	35
III.7. Conclusion :	37
Conclusion générale et perspectives :	38

Introduction générale

Introduction générale

Introduction générale :

Au cours des décennies précédentes, le téléphone n'était qu'un moyen de communication dont la seule mission était de recevoir et d'envoyer des appels et des messages vers et depuis des appareils simples et primitifs.

Mais avec l'avancée de la technologie et des recherches, le simple téléphone a pris un autre aspect en termes de contenu et de forme, car il est devenu un moyen qui rapproche les gens dans le monde, et un outil qui contient de multiples applications utiles utilisées quotidiennement.

La technologie a ajouté au téléphone la touche de programmation, de surveillance, de gestion et de résolution de problèmes à distance.

Notre objectif de cette étude est la mise en œuvre d'une application mobile pour le diagnostic et le suivi des problèmes signalés au niveau de la faculté des sciences et des sciences appliquées de l'Université de Bouira, accompagnée par une maquette 3D de celle-ci. Le but est de proposer une solution permettant de :

- Gérer les problèmes de la faculté d'une manière informatique.
- Faciliter la tâche à l'utilisateur (l'agent de sécurité + superviseur).

Ce rapport de PFE s'articule sur trois chapitres :

- Le premier chapitre contient des généralités sur les applications mobiles, les systèmes d'exploitation et quelques notions sur les maquettes 3D d'architectures et numériques.
- Le deuxième chapitre est consacré principalement sur l'étude du logiciel (l'Android studio).
- Le dernier chapitre est consacré à la réalisation de notre application.

Enfin, nous terminerons par une conclusion générale et proposer quelques perspectives.

Chapitre I :

Généralités sur les applications mobiles et les maquettes

Chapitre I : Généralités sur les applications mobiles et les maquettes

I.1. Introduction :

Actuellement on trouve des applications mobiles de différentes fonctions, formes et pour différentes utilisations ; ces applications contiennent des bases de données qui sont organisées de façon à être facilement accessibles, administrées et mises à jour [1].

Dans ce chapitre on va présenter la notion d'application mobile, ses typologies, ses caractéristiques, ainsi que les principaux systèmes d'exploitation, les outils de développements mobiles précisément l'Android studio, et enfin, quelques notions sur les maquettes.

I.2 Les applications mobiles :

I.2.1. Définition d'une application mobile :

Une application mobile est un logiciel applicatif transportable et autonome, développé pour être installé sur un appareil électronique mobile. Elle est identifiée par un ou plusieurs programmes téléchargeables de façon gratuite ou payante depuis un magasin d'applications "Application Store", permettant d'accéder à un contenu homogène et exécutable à partir du système d'exploitation. Les applications mobiles permettent en général un accès plus pratique, rapide et efficace à des sites en version mobile ou web [1].

Les applications mobiles sont des applications embarquées qui doivent fonctionner sur des plateformes mobiles telles que les Smartphones et les tablettes. L'application doit être téléchargée par l'utilisateur pour s'exécuter dans le système d'exploitation du téléphone, tout comme un logiciel ordinaire installé sur un ordinateur. En raison du langage informatique utilisé, les applications mobiles ne peuvent pas être transférées d'un système d'exploitation à un autre :

- En téléchargeant une application Android, on ne peut l'utiliser que sur un téléphone Android. Reflétant le succès commercial et technique de l'iPhone, la plupart des applications mobiles ont été créées à l'origine pour l'iPhone. Cependant, les applications Android ont fortement augmenté depuis 2010/2011 d'après la référence [2].

I.2.2. Types d'applications mobiles :

Selon la littérature scientifique, il existe plusieurs types d'applications mobiles, parmi eux on peut citer :

Chapitre I: Généralités sur les applications mobiles et les maquettes

a. Natives : les applications mobiles sont dites natives lorsqu'elles utilisent les outils de développements de leurs systèmes d'exploitation (les plateformes et langages Android studio, Java, ou Kotlin pour Android, et Xcode, Swift ou Objective-C pour IOS) [3].

Il s'agit d'une application conçue pour les systèmes d'exploitation les plus fiables des smartphones, en référant la langue spécifique de chaque système d'exploitation. Ce mode d'application n'est accessible que sur la plateforme d'application selon sa spécialité et sa formule [3].

b. Web App : elle ne s'agit pas d'une technologie mobile, mais d'une version mobile d'un site web. Les applications web sont directement encapsulées dans un navigateur web, donc elle n'est pas disponible sur les stores (outil de téléchargement/ plateforme) et elle n'a pas besoin d'être installée sur le smartphone pour fonctionner, ce qui procure un avantage en termes de stockage. L'inconvénient, c'est qu'elle nécessite internet pour fonctionner (son contenu est donc inaccessible hors ligne), bien que des technologies telles que les progressive Web App tentent d'y remédier.

Toute application conçue à l'aide de Html (Hyper Text Markup Language) et CSS (Cascading Style Sheets) qui est plus facile à utiliser sur le navigateur Internet d'un smartphone est appelée une application Web. Quelle que soit la marque du smartphone, nous pouvons accéder à l'application Web via son navigateur, nous n'avons donc pas besoin de la télécharger [2].

Exemple de quelques applications web :

On peut citer quelques exemples des applications web telles que :

- Certains services Google comme Google Maps, Gmail ou le moteur de recherche.
- Certaines offres internet comme Amazon (la boutique en ligne mais aussi des services comme Amazon vidéo et Amazon music) et eBay en font aussi partie [4].

c. Hybrides : une application mobile hybride est une application dont le code source permet de générer deux applications : l'une pour Android et l'autre pour IOS. Elle s'appuie pour cela sur des technologies « Web » telles que le JavaScript, HTML, ou encore CSS. Le contenu d'une application hybride s'exécute dans un « conteneur » natif, une webView (c'est-à-dire une page internet), pour pouvoir être utilisée comme une application native. Il s'agit donc, comme toute application web, d'un code « interprété » qui nécessite d'être traduit par la machine à chaque exécution, ce qui réduit la performance de l'application [3].

Chapitre I: Généralités sur les applications mobiles et les maquettes

I.2.3. Fonctionnalité de chaque type d'application :

Type d'application	Web	Native	Hybride
Accès aux fonctionnalités natives (GPS, etc.)	Non (accès limité)	Oui	Oui
Téléchargement depuis les mobiles	Non	Oui	Oui
Codage une seule fois et exécution sur plusieurs plateformes	Oui	Non	Oui
Accès hors ligne	Non	Oui	Oui
Portabilité du code	Oui	Non	Oui
Coût de développement et de maintenance	Réduit	Elevé	Moyen

Tableau I.1 : La fonctionnalité de chaque type d'application [1].

Remarque :

D'après les données marquées sur le tableau (I.1), on constate que les applications Hybrides possèdent toutes les fonctionnalités (accès aux fonctionnalités natives, téléchargement depuis les mobiles, le codage, l'accès hors ligne, le coût de développement...) par rapport aux deux autres types d'applications (web et native). Et donc le type hybride est notre choix pour la réalisation de notre application.

I.2.4. Caractéristiques des applications mobiles :

- Une application mobile se caractérise par sa rapidité d'utilisation.
- Les applications natives dépendent des capacités du téléphone. De plus, en utilisant les composants graphiques du téléphone (boutons, barres de titre, onglets, etc.), les utilisateurs peuvent plus facilement appréhender l'interface et ainsi s'y adapter.
- Une application offre plus de possibilités : accéder aux photos, aux calendriers, aux contacts, réagir aux mouvements du téléphone, lire les codes-barres : sont autant de fonctionnalités que l'application peut offrir.
- Le contenu d'une application mobile est adapté aux spécificités de chaque smartphone.
- Les applications mobiles peuvent être disponibles hors ligne, et être toujours accessible (plus besoin d'attendre l'ouverture de la page) [2].

I.3. Les systèmes d'exploitation mobiles OS :

I.3.1. Définition d'un système d'exploitation :

Un système d'exploitation ou logiciel système, ou encore Operating System (OS) est un logiciel qui contient un ensemble de programmes chargé de relier les ressources matérielles et logicielles d'un appareil mobile, et permet de gérer toutes les autres applications (démarrage, l'arrêt, allocation de ressources mémoire) et la communication avec les supports physiques (clavier, souris, surface tactile, écran, disque dur, lecteur de carte mémoire,...), donc il se concentre sur la gestion des connexions de différents types d'interface.

Les principaux systèmes d'exploitation connus commercialement sont : Android, Ios, Windows mobile, BlackBerry [1].

I.3.2. Les principaux systèmes d'exploitation mobiles :

- A. Android (Google) :** C'est un système d'exploitation qui est fondé sur le noyau Linux et développé par des informaticiens sponsorisé par Google, il est utilisé pour développer les smartphones, les tablettes, les montres digitales, ...etc. Android est un OS open source et son code source est librement accessible et gratuit pour les constructeurs d'appareils souhaitant l'utiliser (contrairement au système Apple ou Microsoft) [1].



Figure1: Logo du système Android.

- A. IOS (iPhone) :** IOS (InterNetwork Operating System) est un **système** d'exploitation mobile créé en 1976 et développé par Apple pour ses appareils mobiles : iPhone, iPad et iPod Touch (23.4% des parts sur le marché d'après des informations collectées par Kantar World Panel au premier trimestre 2021) [6]. Apple a publié la première version de son système d'exploitation en 2008 nommé iPhone OS (Ios 1) [7].

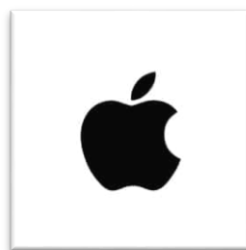


Figure 2: Logo du système IOS [8].

B. Windows Mobile (Microsoft) :Windows mobile est le système d’exploitation développé par Microsoft en 2010 pour ses appareils mobiles avec ou sans écran tactile : smartphone (Lumia, Asha,...), tablettes (Microsoft surface),...etc [7].

Il est le système d’exploitation le plus utilisé dans le monde dans le domaine des ordinateurs de bureau et portables à environ de 77% à 87,8% d'après la référence [9].



Figure 3: Logo du système Windows mobile [10].

C. BlackBerry OS : est une plateforme de communication, elle a été développée par RIM (Research In Motion). RIM fournit des solutions qui intègrent du matériel, des logiciels et des services, permettant un accès simple et transparent aux informations importantes telles que les messages électroniques, les messages téléphoniques, les SMS et les applications Internet et Intranet [7].



Figure 4: Logo du système blackBerry [11].

I.3.3. Comparaison entre les systèmes d’exploitation :

	IOS	BlackBerry	Windows mobile	Android
Langage de programmation	Objective-c	Java	C,C++	Java
	Integré à Xcode	Gratuit	Gratuit	Gratuit
Disponibilité de l’environnement de développement	Xcode	JDE	Visuel studio, eMbeddb VC++	Eclipse, Android studio
Multiplateforme	iPhone, iPod	BlackBerry	Windows mobile,	Android

Chapitre I: Généralités sur les applications mobiles et les maquettes

de déploiement	touch, iPad	sulement	windows CE	seulement
Cout d'outils de développement	Gratuit	Gratuit	Gratuit	Gratuit
Magasin en ligne	App Store	App World	Windows market place	AndroidMarket
Open source	Non	Oui	Non	Oui
Constructeur	Apple	RIM	Microsoft	Google

Tableau I.2 : Comparaison entre les systèmes d'exploitations mobile [2].

Remarque :

Sur la base des données répertoriées dans le tableau (I.2) nous constatons que chaque système d'exploitation a ses propres avantages, tels que les langages de programmation, les plateformes de déploiement, magasins en ligne, et constructeurs. Nous notons également que tous les systèmes d'exploitation ne sont pas open source sauf Android et BlackBerry, et pour cette raison nous avons choisi de créer notre application en utilisant le système d'exploitation Android.

I.4. Les outils de développement mobile :

Chaque plateforme possède ses propres outils de développement (langages, logiciels...), et le langage utilisé pour développer les applications mobiles dépend du système sous lequel l'application sera exécutée [1].

Nous allons présenter dans ce qui suit les outils de développements pour les trois principales plateformes mobiles : Android, IOS et Windows mobile :

- a) **Android** : pour développer une application mobile sur ce système, les 5 meilleurs outils de développement d'applications suivants sont nécessaires :
- Eclipse : est l'un des outils de développement d'applications Android les plus connus et les plus utilisés.
 - Android studio
 - Gestionnaire AVD : « Appareil Virtuel Android », c'est l'émulateur qui sera utiliser pour installer et exécuter une application
 - SDK Android : « Software Development Kit » ou bien le kit de développement logiciel, aussi appeler trousse de développement logiciel. C'est un ensemble d'outils logiciels destinés aux développeurs pour faciliter le développement d'une application sur une plateforme donnée [9].

Chapitre I: Généralités sur les applications mobiles et les maquettes

- b) **IOS** : Le langage utilisé pour développer une application dans la plateforme Appl sera l'Objective-c (ou Swift) qui est un langage de programmation à la fois fiable, rapide et efficace qui fournit des retours en temps réel et s'intègre naturellement au code Objective-c. Pour développer une application sur IOS, il faut aussi installer l'environnement de développement Xcode [12].
- c) **Windows Mobile** : Le langage utilisé pour développer des applications dans cette plateforme sont généralement C, C++ accompagné des technologies web (HTML, CSS, JavaScript), et l'environnement de développement utilisé est Microsoft Visual Studio[1].

I.5. Le système d'exploitation Android :

I.5.1. Le choix de la plateforme Android:

Nous avons précédemment décrit grossièrement les différents systèmes d'exploitation utilisés par les applications mobiles.

La section suivante est dédiée à la plateforme sur laquelle nous allons développer notre application dans le cadre de ce PFE. En effet, nous avons choisi le système d'exploitation Android pour les raisons suivantes :

- Android est une plateforme open source puissante, moderne, sécurisée et ouverte. Grâce au code open source et aux **API (Application Programming Interface** ou Interface de Programmation d'Application), les développeurs ont accès pour intégrer, étendre et modifier les composants existants. Les utilisateurs peuvent ajuster l'application en fonction de leurs besoins.
- Android est basé sur le noyau Linux. Par conséquent, il existe plusieurs avantages tels que la grande mémoire, la gestion des processus, le modèle de sécurité, le support de la bibliothèque partagée, etc.
- Le SDK Android fournit des API (interfaces de programmation) complètes permettant aux développeurs de créer des applications riches sur Android [2].

I.5.2. La naissance d'Android :

L'histoire d'Android commence en octobre 2003, où la société Android Inc. est créée. Officiellement, elle développe des logiciels pour mobiles, elle se préparait à sortir un tout nouveau système d'exploitation pour smartphones. En 2005, Google rachète cette entreprise, et sort une première bêta en novembre 2007, avant de lancer la version 1.0 en septembre 2008 avec le HTC Dream. À partir de ce moment-là, le rythme des nouvelles sorties est très élevé : pas moins de 11

Chapitre I: Généralités sur les applications mobiles et les maquettes

versions différentes sont sorties en 3 ans. On remarquera au passage que chaque version d'Android porte le nom d'un dessert [2].

I.5.3. Historique des versions d'Android :

Version Android	Nom	Lancement	Logos
1.0	Apple pie	23 septembre 2008(Dan .M, 2008).	
1.1	Bananas Split	09 février 2009 (Android Developpers, 2009)	
1.5	Cupcake	30 avril 2009 (Xavier.D ,2009)	
1.6	Donut	15 septembre 2009 (Xavier.D ,2009)	
2.0	Eclair	26 Octobre 2009 (Xavier.D ,2009)	
2.2	Froyo	20 mai 2010 (Xavier.D ,2010)	
2.3	Gingerbread	6 Décembre 2010 (Xavier.D ,2010)	
3.0 / 3.1 / 3.2	Honeycomb	22 Février 2011 (Xavier.D ,2011)	
4.0	Ice cream Sandwich	19 Octobre 2011 (Francis.R ,2011)	
4.1/4.2 /4.3	Jelly Bean	9 Juillet 2012 (Android developpers, 2012)	
4.4	Key Lime Pie(KitKat)	3 Septembre 2013	
5.0	Lollipop	15 Octobre 2014	
6.0	Marshmallow	Fin Mai 2015	
7.0	Nougat	10 Mars 2016	
8.0	Oreo	21 Aout 2017	
9.0	Pie	7 Mars 2018	
10	Nom de code Q	3 Septembre 2019	
11	Nom de code R	8 septembre 2020	
12		18 Février 2021	

Tableau I.3 : Historique de l'Android et ses différentes versions.

Chapitre I: Généralités sur les applications mobiles et les maquettes

I.5.4. Le kit de développement (SDK) :

Software Development Kit (SDK), ou bien **un kit de développement logiciel**, aussi appelé **trousse de développement logiciel**, est un ensemble d'outils logiciels destinés aux développeurs, afin de faciliter le développement d'un logiciel sur une plateforme donnée, par exemple (IOS, Android, BlackBerry10, Symbian, Bada, linux, OS X, Microsoft Windows).

Le SDK Android est composé de plusieurs éléments afin d'aider les développeurs de créer et maintenir des applications, parmi ces éléments on trouve :

- des exemples de codes : les bibliothèques Java pour créer des logiciels
- les outils de mise en boîte des logiciels
- AVD : un émulateur de tablettes pour tester les applications
- ADB : un outil de communication avec les vraies tablettes [15].

I.6. Notions sur les maquettes :

I.6.1. Définition d'une maquette :

Selon le dictionnaire «Larousse » une maquette est :

"Une représentation en deux ou trois dimensions, le plus souvent à échelle réduite, ou agrandie mais fidèle dans ses proportions, d'une construction, d'un appareil, d'un décor, d'un objet quelconque. Elle peut aussi être un ensemble de pièces à monter pour obtenir une reproduction en modèle réduit de quelque chose, en particulier d'un avion, d'un véhicule : monter une maquette" [16].

I.6.2. Définition de la maquette d'architecture :

Les modèles de construction sont des représentations 3D à plus petite échelle de bâtiments existants ou de futurs projets de construction. Mais une maquette ne doit pas seulement représenter un bâtiment, mais aussi intégrer son environnement ou futur environnement, comme les routes, les terrains et les infrastructures qui l'entourent. Par ailleurs, la maquette peut également représenter un projet d'urbanisme, et pas seulement un bâtiment, c'est-à-dire un ensemble d'infrastructures telles que plusieurs bâtiments, une route, un accotement, un rond-point, un nouveau parc, un nouveau quartier ou lotissement [17].

I.6.3. Critères de classification des maquettes d'architectures :

Selon la référence [17], la maquette d'architecture peut être classifiée par:

- Sa fonction :
- Sa technologie de fabrication
- Son matériel
- Son temps

- Son utilisation.

Aujourd'hui, de façon générale, les modèles architecturaux peuvent être classifiés en fonction de leurs utilisations ou de leurs fonctions [17].

I.6.4. Classification des maquettes d'architectures par utilisation :

Selon la référence [17] on peut classer les maquettes par utilisation comme suit :

I.6.4.1. Maquette de conception :

Les maquettes conceptuelles sont les modèles utilisés pour analyser les caractéristiques du projet ainsi que la disposition spatiale et l'environnement environnant, ils sont développés au cours des premières étapes de la conception architecturale pour trouver les formes souhaitées, c'est-à-dire des croquis. C'est l'une des premières méthodes de conception.



Figure 5: Maquette de conception [17].

I.6.4.2. Maquette de présentation :

C'est un modèle de démonstration typique qui s'adapte à l'échelle requise par le client. Ce modèle est souvent produit en détail pour fournir aux clients ou aux décideurs un reflet fidèle de la conception architecturale, des points de vue spéciaux peuvent être abordés en montrant les façades détaillées des bâtiments et d'autres éléments du projet.



Figure 6: Maquette de présentation [17].

I.6.4.3. Maquette de planification urbaine :

Le modèle urbain permet au public de comprendre le développement de la ville, y compris l'aménagement du territoire, le paysage urbain, la planification des transports,... etc .De nombreux départements utilisent des modèles d'urbanisme pour promouvoir la régénération urbaine et attirer des investissements et des dons. Il existe actuellement deux modèles d'urbanisme : la partie la plus ancienne aide les gens à comprendre à quoi ressemblait la ville, les nouvelles pièces(les parties mises à jour) permettent aux gens une vision de l'avenir. Les modèles de ces villes miniatures se présentent sous différentes formes et tailles.



Figure 7: maquette de planification urbaine [17].

I.6.4.4. Maquette de paysage :

Les modèles de paysage contiennent diverses caractéristiques spécifiques à la région, tels que les terrains, les développements de bâtiments, différents types de sites (tels que des fermes et des parcs), et d'autres installations telles que des éoliennes et des installations solaires photovoltaïques. Le but du modèle de paysage est d'analyser topographiques, la disposition des bâtiments et d'autres influences environnementales possibles.



Figure 8: Maquette de paysage [17].

I.6.4.5. Maquette de bâtiment de bureaux :

Le modèle d'immeuble de bureaux est créé au niveau de détail élevé demandé par le client pour fournir une représentation réaliste du projet de construction. Généralement elle est utilisée pour des

Chapitre I: Généralités sur les applications mobiles et les maquettes

expositions marketing ou architecturales. Le modèle de bureau donne vie aux conceptions de l'architecte tout en créant diverses configurations d'éclairage, des façades détaillées, en particulier les détails et les textures des fenêtres, et d'autres éléments tels que les personnes, les véhicules et le paysage environnant.



Figure 9: Maquette de bâtiment de bureau [17].

I.6.4.6. Maquette de bâtiments résidentiels :

Les modèles de bâtiments résidentiels sont des outils inestimables dans le commerce de détail pour les approbations de pré-construction, ces modèles peuvent fournir aux acheteurs potentiels de maisons ou aux investisseurs une bonne compréhension de la conception des bâtiments en montrant les tracés des routes, les entrées, les matériaux de construction et les environs. C'est donc un excellent outil de marketing pour les agences immobilières.



Figure 10: Maquette de bâtiments résidentiels [17].

I.6.4.7. Maquette d'usine :

Le modèle d'usine montre comment l'usine fonctionnera une fois construite et comprend diverses installations et processus de traitement des déchets dans une usine de fabrication, etc. Parfois, les modèles d'usine nécessitent des composants personnalisés, tels que des éoliennes et des cheminées, pour représenter les conditions réelles de modèle de paysage.



Figure 11: Maquette d'usine [17].

I.6.4.8. Maquette d'intérieur :

Les modèles d'intérieurs sont réalisés selon différentes fonctions telles que l'industrie, l'entreprise, le bureau et la maison. Une fois le modèle intérieur adapté au modèle du projet architectural, le toit peut être retiré pour indiquer la conception et la décoration de l'espace intérieur, ainsi que le choix du mobilier. Il est largement utilisé sur le marché immobilier pour attirer les investissements d'acheteurs potentiels ou de bureaux de vente.



Figure 12: Maquette d'intérieur [17].

I.6.5. L'objectif de la maquette 3D :

Pour réaliser un projet architectural, qu'il s'agisse d'un bâtiment, d'une route, d'un pont ou d'un plan d'urbanisme, un plan de conception ou un croquis est important. Mais pour rendre le programme plus réaliste, la maquette architecturale est un dispositif indispensable. Il s'agit d'un livre qui présente les types de projets au public en 3D (modélisation 3D).

La modélisation 3D est un outil de visualisation efficace qui permet de monter aux clients et aux parties prenantes la complexité et la portée d'un projet. De plus, la modélisation 3D offre une vue d'ensemble et sert les intérêts à la fois des ingénieurs et du client [18].

I.7. La maquette numérique BIM :

Définition de BIM :

De manière générale, le BIM, Building Information Modeling en anglais, désigne la modélisation des informations du bâtiment. Cet outil est un processus de modélisation de l'ouvrage. Ce processus a comme support important la maquette 3D appliquée au projet architectural.

Chapitre I: Généralités sur les applications mobiles et les maquettes

La maquette numérique génère les documents nécessaires pour le dépôt du permis de construire (plans, coupes, visualisation 3D,...) [19].

I.8. Exploitation des données 3D :

I.8.1. Les logiciels de modélisation 3D :

Nous allons désormais nous intéresser aux différents logiciels disponibles qui permettent de modéliser des objets en trois dimensions en vue de produire une maquette numérique.

Ils sont assez nombreux sur le marché. C'est pourquoi il est primordial de les tester dans la mesure du possible et d'étudier à chacun leurs avantages et inconvénients [20].

I.8.1.1. Autocad Architecture :

La solution proposée par Autodesk se nomme Autocad Architecture. Elle se présente comme une alternative à Revit. Les utilisateurs habitués à travailler avec Autocad en DAO classique retrouveront leurs marques rapidement grâce à une interface intuitive. La gestion des différents niveaux du bâtiment se fait par élévation [20].

I.8.1.2.Revit :

C'est l'un des logiciels les plus intéressants actuellement proposé sur le marché. Sa bibliothèque de familles d'objets est complète et puissante. La création de nouveaux éléments est possible grâce à un éditeur d'objet inclus. Chaque objet est intelligent. Les propriétés sont paramétrables manuellement et se mettent à jour lorsque des objets liés sont modifiés. Une base de données regroupe l'ensemble des informations et permet de créer des listings ou métrés des éléments [20].

I.8.2. Choix d'un logiciel de modélisation :

La réalisation d'une maquette numérique est relativement chronophage (demande beaucoup de temps) et nécessite plus de temps qu'une simple modélisation. Le logiciel retenu se doit donc d'être performant tout en restant simple d'utilisation.

Nous avons choisi l'Autocad Architecture et Revit pour réaliser notre maquette dans le cadre de notre projet PFE, car ces logiciels sont performant, gratuits et simples d'utilisation, et plus encore : [20].

- Automatisation des plans d'étage, coupes et élévations.
- Dessin rapide des tuyauteries, gaines et circuits grâce aux bibliothèques de pièces.
- Génération automatique d'annotations, de calques, de nomenclatures, de listes et de tableaux.

Chapitre I: Généralités sur les applications mobiles et les maquettes

- Utilisation d'un workflow basé sur des règles permettant de mettre en œuvre précisément les normes du secteur [31].

I.9. Conclusion :

Nous avons rappelé dans ce chapitre les notions de base d'une manière générale, qui sont nécessaire pour la compréhension de la suite de ce manuscrit.

Nous avons aussi justifié notre choix du type hybride de l'application à réaliser sous le système Android. Ce dernier est également choisi parmi les systèmes d'exploitation existant.

Chapitre II : Outils et logiciels utilisés

Chapitre II: outils et logiciels utilisés

II.1. Introduction :

De nos jours, de nombreuses sociétés souffrent de problèmes de sécurité et de contrôle informatique, pour cela ces entreprises cherchent des méthodes et des outils plus adaptés et plus efficaces pour améliorer sa bonne gestion.

La bonne gestion nécessite la présence d'un bon logiciel informatique. Ce logiciel va être représenté dans ce chapitre et que nous allons utiliser pour développer notre application (logiciel Android studio).

II.2. Android Studio :

C'est un environnement de développement (plateforme IDE officielle de Google) des applications mobiles Android. Il présente un éditeur de mise en page visuel, où les développeurs peuvent créer des mises en page composées et les prés visualisés sur n'importe quelle taille d'écran sur différents appareils [23].

- Le logiciel Android studio permet d'offrir :
 - ✓ Un éditeur de source et de ressource.
 - ✓ Des outils de compilation : Gradle (outil de construction pour fournir, automatiser et créer des logiciels meilleurs et plus rapides)
 - ✓ Des outils de test et de mise en points.
- Android studio est un outil de développement qui permet aux développeurs :
 - ✓ de créer des mises en pages complexes.
 - ✓ De réduire la taille de l'application.
 - ✓ D'émulateurs rapides.
 - ✓ Fournir des statistiques en temps réel.
 - ✓ Une construction personnalisée..., etc.



Figure 13: Logo de la plateforme Android studio [24].

Chapitre II: Outils et logiciels utilisés

II.3. Les étapes principales de création d'application Flutter sur Android studio :

II.3.1 : Définition du Flutter : C'est un SDK de développement d'applications mobiles multiplateforme open source créé par Google. Il est très convivial et crée des applications mobiles de haute qualité [25].



Figure 14: Logo de développeur Flutter [26].

II.3.2. Création de l'application Flutter :

Pour créer une application Flutter nous suivons les principales étapes présentées ci-dessous [25] :

Etape 1 : Créer le projet Flutter :

Pour commencer à créer une application Flutter, nous avons d'abord créé un projet Flutter [25]:

- Nous avons ouvert l'IDE Android Studio et on a sélectionné démarrer un nouveau projet Flutter.

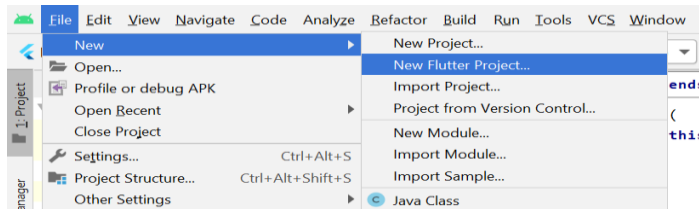


Figure 15: Fenêtre de l'IDE Android studio [25].

Etape 2: Nous avons sélectionné l'Application Flutter (**Flutter Application**) comme type de projet, et cliquons ensuite sur **Next**, comme le montre la Figure(18).

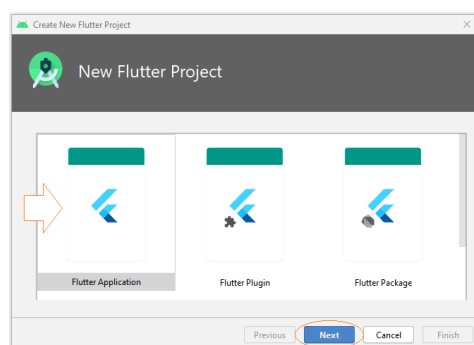


Figure 16: Fenêtre de l'IDE Android studio [25].

Etape 3 : Dans cette étape nous avons vérifié que le **chemin Flutter SDK** spécifie l'emplacement du SDK (**sélectionnez Install SDK...** si le champ de texte est vide).

Chapitre II: Outils et logiciels utilisés

Etape 4 : Dans cette étape, On donne le nom du projet (**Project Name**), par exemple (myapp). Et on clique ensuite sur suivant(**Next**).

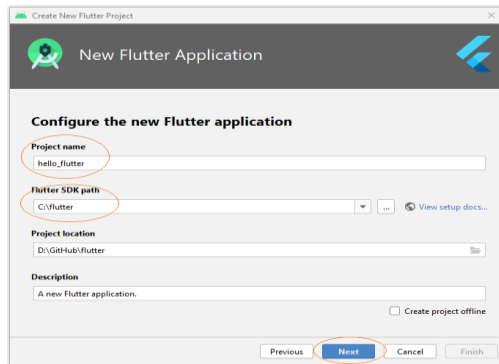


Figure 17: Fenêtre de l'IDE Android studio [25].

Etape 5 : Nous citons le nom de package (**package Name**), et ensuite nous cliquons sur terminer (**Finish**) et Le projet a été créé.

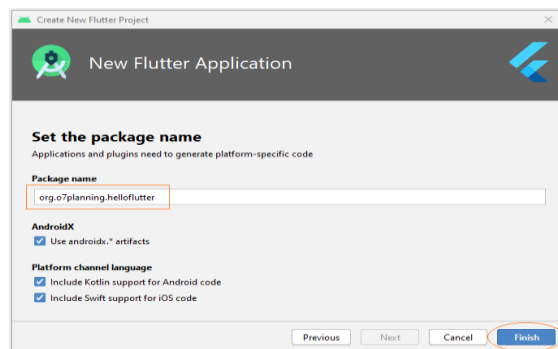


Figure 18: Fenêtre de la citation du nom de projet [25].

La structure du projet qui a été créée, comme illustré à la **Figure (21)**.

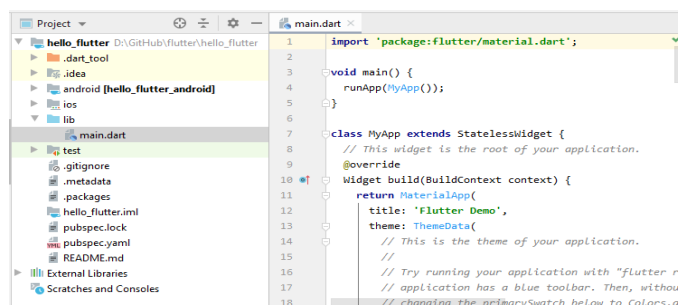


Figure 19: La structure du projet [27].

II.3.3. L'explication de la structure de projet : [26]

android : Le dossier génère automatiquement le code pour l'application d'**Android**.

iso : Le dossier génère automatiquement le code pour l'application d'**ios**.

lib : Le dossier d'accueil contient le code Dart de l'application.

Chapitre II: Outils et logiciels utilisés

lib/main.dart : Le fichier est convoqué pour démarrer (Start) l'application.

test : Le dossier contient les codes **Dart** pour tester l'application.

test/widget_test.dart: Simple code.

.gitignore: **Git version control file**, ce dossier contient la configuration du projet **GIT**.

.metadata : Le dossier est automatiquement généré par l'outil de **Flutter**.

.packages : Automatiquement généré, le fichier contient une liste de dépendances utilisées par le projet.

.iml : Un fichier de projet d'**Android studio**.

Pubspec.lock : Ce fichier doit être ajouté à **GIT Control** pour s'assurer que les membres de votre équipe de développement utilisent les mêmes versions de bibliothèque.

README.md : Le fichier décrit le projet, lequel est écrit selon la structure **Markdown**.

II.3.4. Ecrire le code pour l'application :

Supprimer le contenu du fichier `main.dart` et le remplacer par un nouveau.

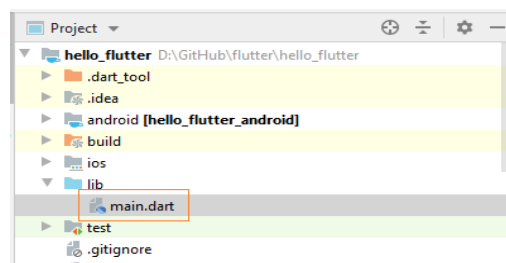
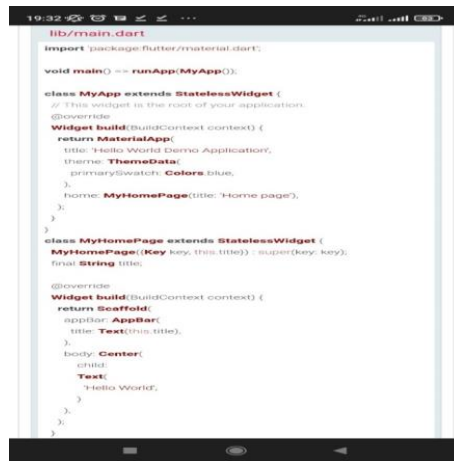


Figure 20: Fenêtre du fichier `main.dart` [25].

Par exemple on peut le remplacer par le code suivant pour l'exécution d'une application Flutter :



```
lib/main.dart
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Hello World Demo Application',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MyHomePage(title: 'Home page'),
    );
  }
}

class MyHomePage extends StatelessWidget {
  MyHomePage({Key key, this.title}) : super(key: key);
  final String title;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(this.title),
      ),
      body: Center(
        child: Text(
          'Hello World!',
        ),
      ),
    );
  }
}
```

Figure 21: Exemple d'un code d'exécution [27].

II.3.5. Lancement de l'application :

L'application Flutter doit être mise en place dans un appareil d'Android ou d'ios. Par conséquent, au cours du processus de développement, nous avons, comme choix, l'une des options suivantes :

1. Connecter l'appareil Android à un ordinateur et activer le mode développeur (développer mode).
2. Connecter l'appareil Ios à l'ordinateur (comme un iPhone), et activer le mode développeur (développer mode).
3. Exécuter un émulateur Android (Android Emulator).

Au cours de la programmation d'une application Flutter dans Android Studio, la meilleure approche consiste à lancer un émulateur d'Android (Android Emulator).

Dans Android Studio on sélectionne :

- **Tools>AVDManager**

Ou on clique sur l'icône « AVD Manager » sur la barre d'outils :

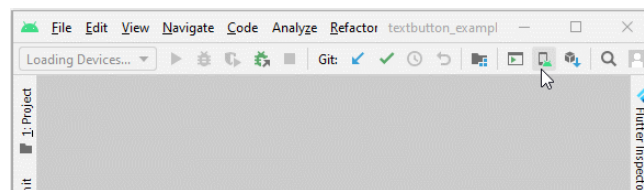


Figure 22: La barre d'outils de Flutter [27].

Chapitre II: Outils et logiciels utilisés

Ensuite on a lancé une application virtuelle sur la liste :

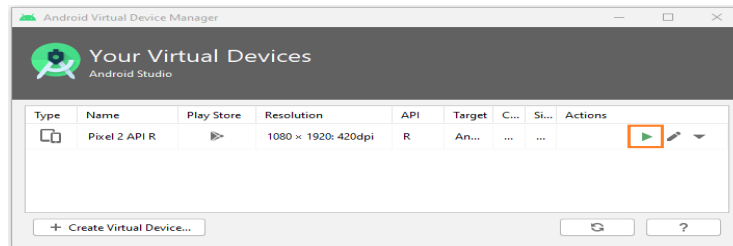


Figure 23: Fenêtre de l'AVD Manager.

Android Emulateur est mis en place et prêt à déployer l'application Flutter :



Figure 24: L'Emulateur d'Android [27].

Dans la barre d'outils d'Android Studio, et pour lancer notre application Flutter on a utilisé l'illustration suivante :

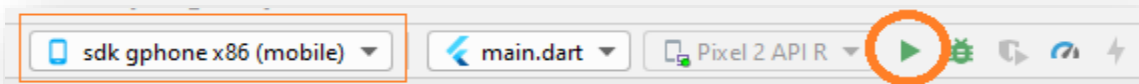


Figure 25: La barre d'outils d'Android studio [27].

Enfin on a le résultat final suivant du notre exemple

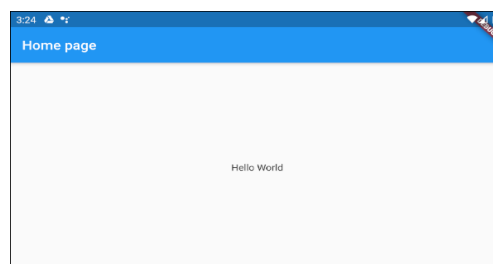


Figure 26: Fenêtre du résultat final [27].

II.5. L'organigramme de création de notre application:

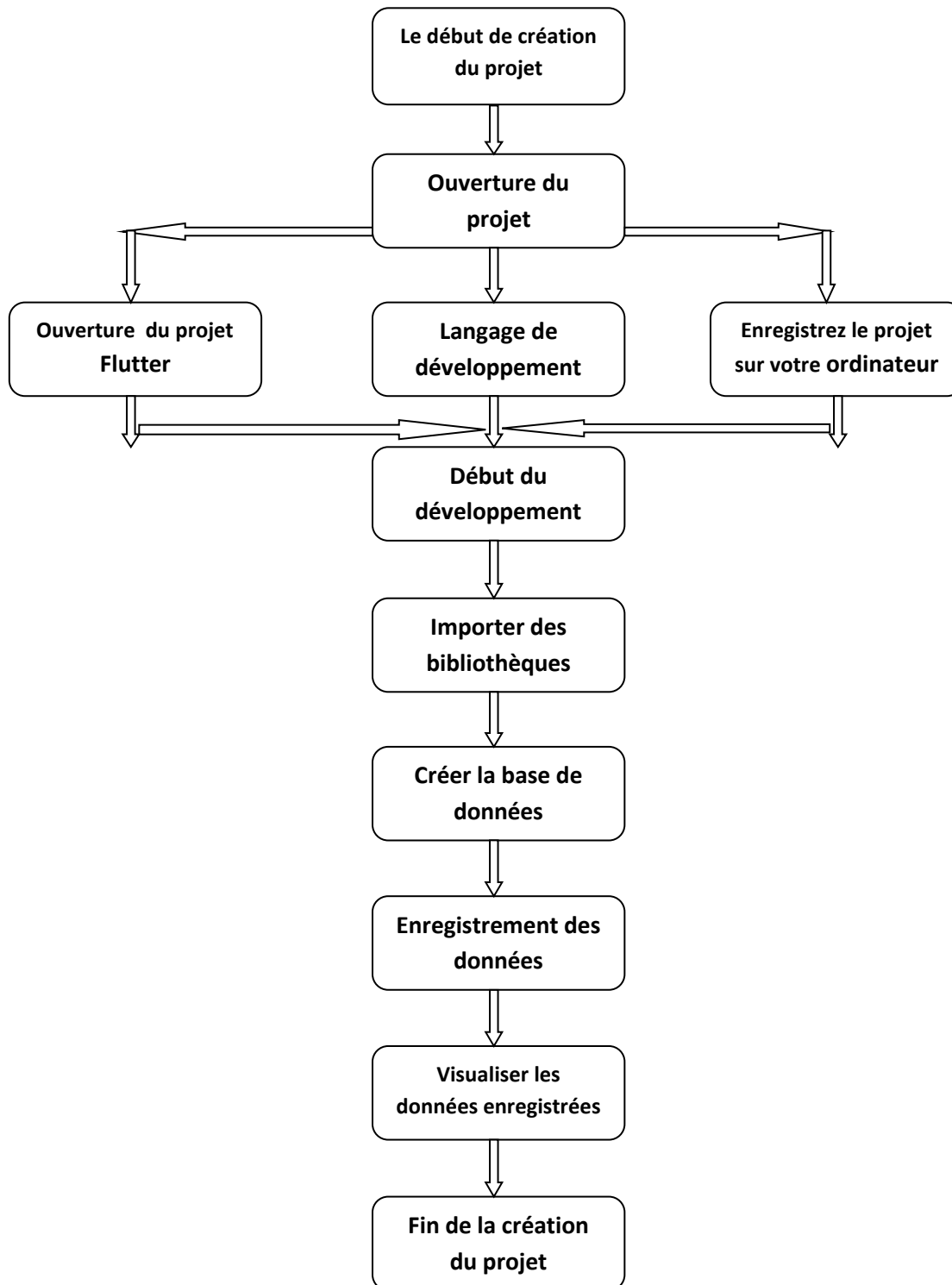


Figure 27: Organigramme de création de notre application.

Chapitre II: Outils et logiciels utilisés

II.6. Code de programmation :

Pour créer une application mobile, il faudrait un code de programmation, dont les formats du programme le choix revenue au développeur est dans les figures ci-dessous (Figure 29 et 30 et 31)

On donne notre format du travail par trois étapes :

Etape 1 : La partie des bibliothèques importées

Cette partie sert à importer les bibliothèques nécessaires au développement de l'action requise, sachant que les bibliothèques permettent d'interagir directement avec l'appareil sur lequel tourne le programme. Il contient des fonctions de manipulation de fichiers et de gestion de tâches et contient également des instructions de programmation.

```
import 'package:flutter/material.dart';
import 'package:pages/sqldb.dart';
import 'AdminScreen1.dart';
import 'addProblemScreen.dart';
import 'homeScreen.dart';
```

Figure 28: Exemple des bibliothèques.

Etape 2: Créer la classe, Cette étape permet de nommer l'action nécessaire pour démarrer la programmation de votre activité.

Le nom de la classe (**AddAjent**) comme le montre la **Figure 30**.

```
class AddAjent extends StatefulWidget {
  static const id = 'addAjent';
```

Figure 29: Exemple d'une classe créée.

Etape 3: Lors de la création d'une sous-classe de la classe parente dans cette activité, nous trouvons les conceptions et les rôles de l'activité souhaitée.

Le nom de sous- classe (**_AddAjentState**) comme le montre la **Figure 31**.

```
@override
State<AddAjent> createState() => _AddAjentState();
}

class _AddAjentState extends State<AddAjent> {
```

Figure 30: Exemple de la sous-classe créée.

Ces étapes différentes d'une activité à l'autre en fonction du nombre d'actions nécessaires pour chaque activité.

Chapitre II: Outils et logiciels utilisés

II.7. La modélisation numérique de la maquette :

II.7.1. Définition de la modélisation numérique:

La modélisation numérique est une représentation virtuelle d'un objet technique ou d'un espace. Elle se fait grâce à un logiciel de modélisation numérique appelé modeleur numérique. On peut aussi utiliser des logiciels de C.A.O (logiciels de Conception Assisté par Ordinateur.)

La modélisation numérique permet de représenter un objet technique dans un espace à trois dimensions (3D). [20]

II.7.2. La modélisation de notre maquette (de la FSSA):

Les résultats suivants donner par logiciel modélisation Revit comme montre les figures ci-dessous.(Figure 33 et 34).

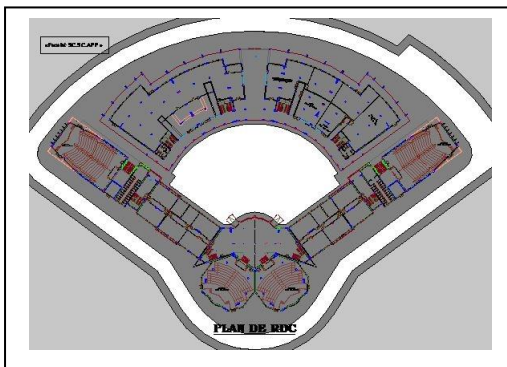


Figure 33 : Plan d'un bloc de La Fssa

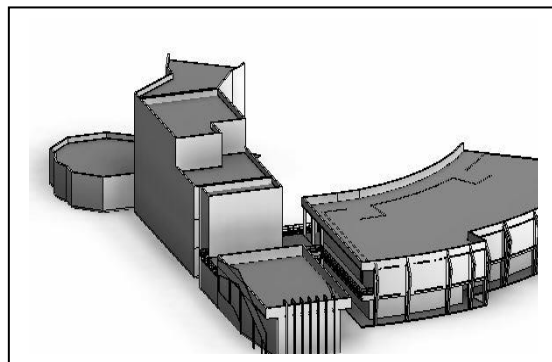


Figure 34: modélisation du plan.

II.8. Conclusion :

Dans ce chapitre est donne l'ensemble des outils et logiciels utilises pour la création de l'application de notre projet. Sous le système d'exploitation Android Studio, les étapes de création sous Flutter sont détaillées et présentés.

Les résultats obtenus sont aussi présentés et argumentés.

Chapitre III: Développement de l'application et la maquette.

Chapitre III : développement de l'application.

III.1. Introduction :

Dans ce qui suit, nous allons introduire d'une part les différents langages utilisés pour l'implémentation de notre application, et d'autre part, nous présenterons le fonctionnement de celle-ci et des différentes interfaces annoncées.

III.3. Langages de développement de notre application :

III.3.1. Dart :

Dart est un langage de programmation qui optimise sur plusieurs plateformes (multiplateforme). Il est développé par Google et est utilisé pour créer des applications langage orienté objet avec une syntaxe de type C++ [28].

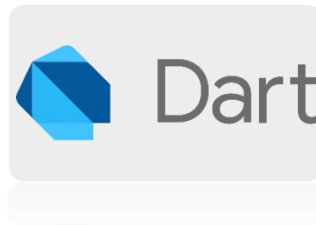


Figure 35: Logo de langage Dart [26].

III.3.2. Flutter et Dart :

Flutter (kit de développement de Google), est l'outil qui nous permet de créer les applications, combiné au langage de programmation Dart. Flutter et Dart tous deux sont créés par Google pour nous aider à créer des applications pour iOS et Android [29].



Figure 36: Dart et flutter [30].

III.4. Organigramme d'utilisation de l'application:

La structure d'utilisation ci-dessous présente les différents utilitaires de l'application en suivant les options proposées pour découvrir et diagnostiquer les problèmes existants et bien les suivre:

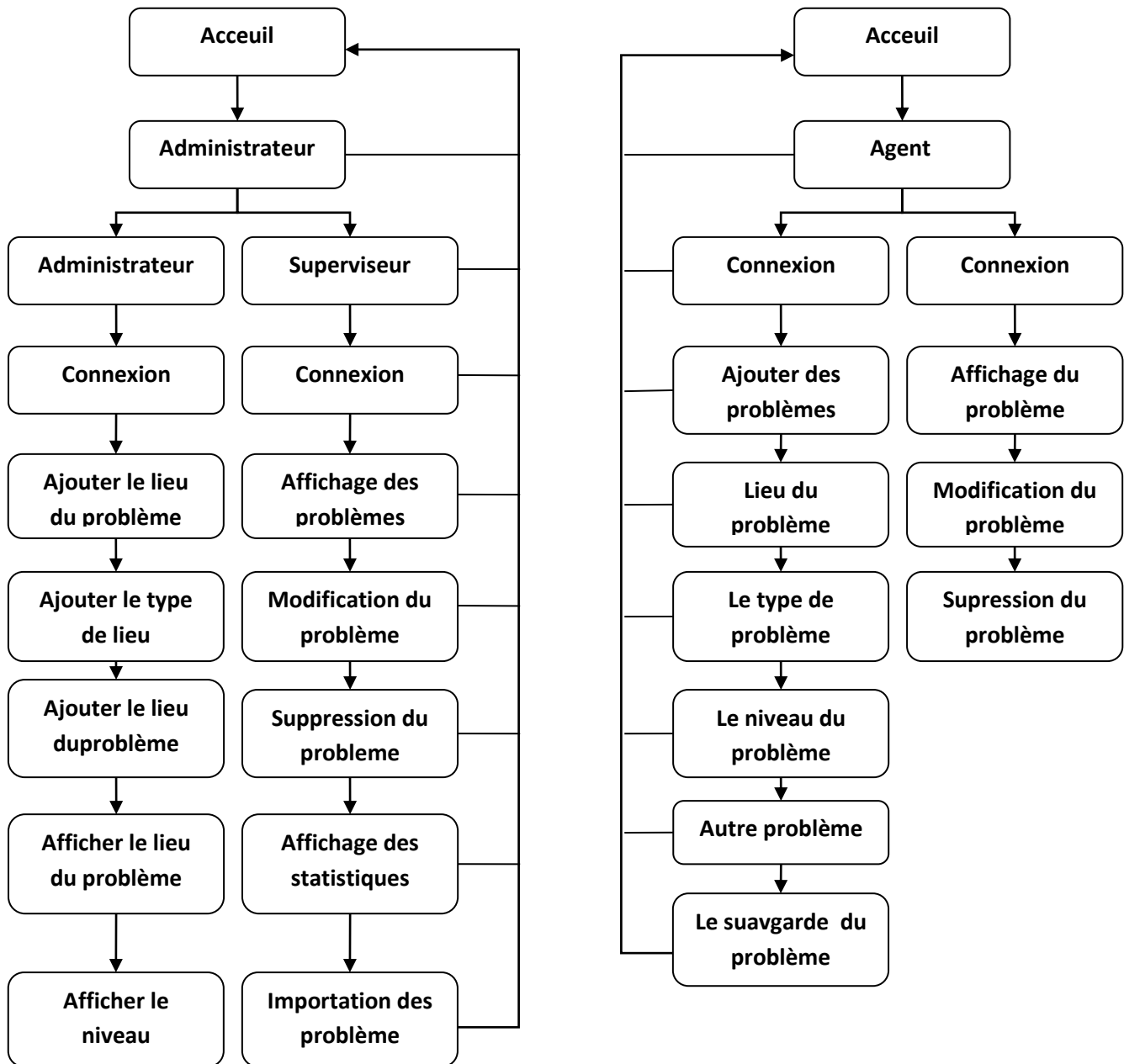


Figure 37: Structure de l'utilisation de l'application par l'administrateur et l'agent.

Remarque

L'application est utilisée par deux personnes, le superviseur (Administrateur) et l'agent.

Le superviseur: Diagnostiquer et suivre les problèmes.

Agent : Pour découvrir le problème.

III.4. Capture d'écrans des activités de l'application :

Dans ce qui suit nous présenterons les différentes interfaces de notre application en 2 parties (première partie pour une seule salle du département, et la deuxième pour le département complet) suivi par une maquette 3D de notre département en citant les détails de chaque imprime écran.

III.4.1. Les interfaces générales de notre application :

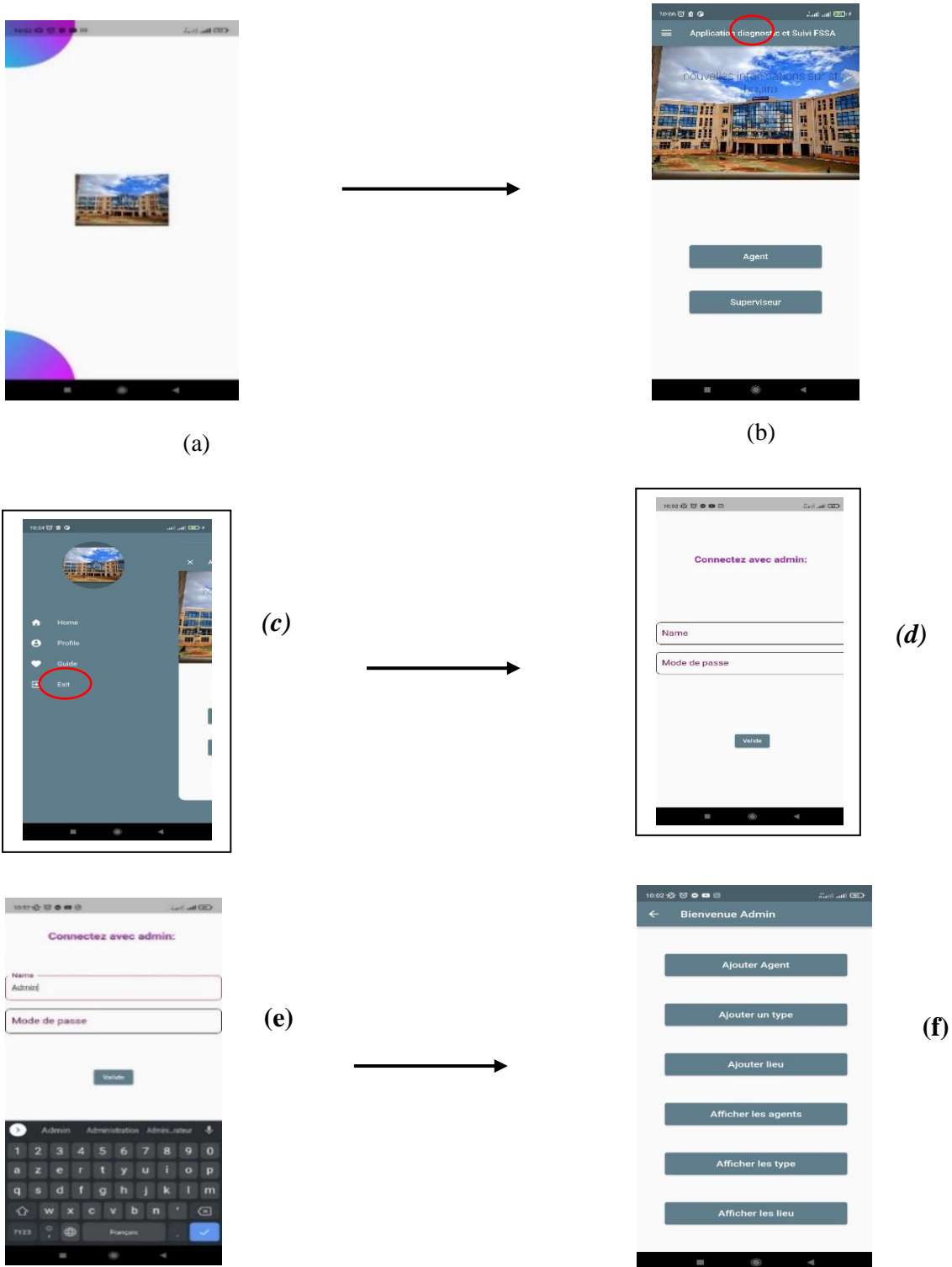


Figure 38: Les interfaces de l'application.

Chapitre III : Développement de l'application et de la maquette

- La figure (a) illustre la page d'accueil de l'application.
- La figure (c) illustre la page d'outil de l'application, et si nous cliquons sur le bouton "Home" il nous donne la page de superviseur / l'administrateur figures (d,e) .
- La figure (f) montre la page de choix qui s'affiche directement après la connexion du superviseur avec le nom et le mot de passe qui sont strictement nécessaires.

III.4.2. L'interface de superviseur (admis) :

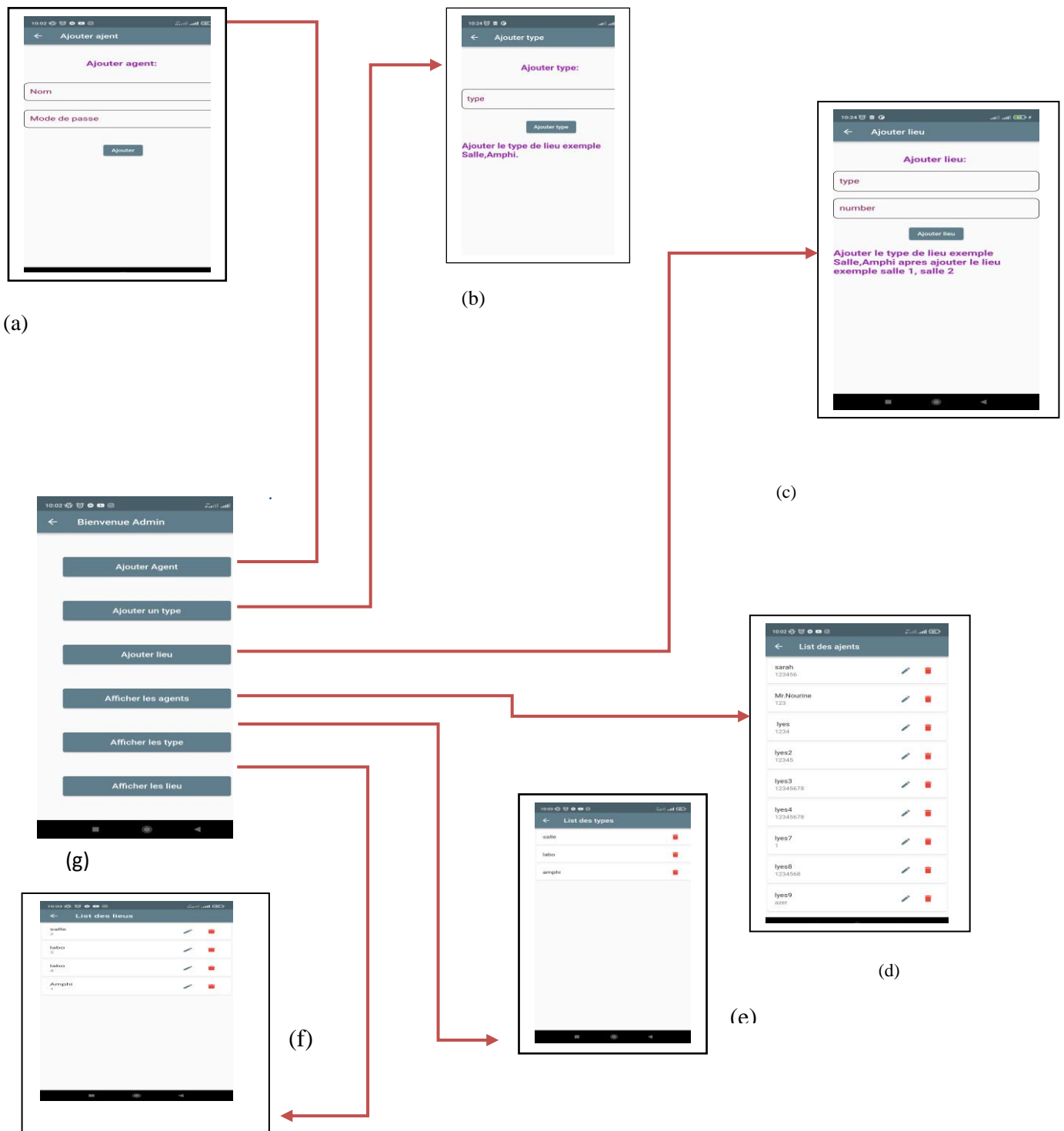


Figure 39: Interface d'accueil d'administrateur.

Chapitre III : Développement de l'application et de la maquette

Les résultats obtenus sont :

- (a) La page qui permet d'ajouter un agent
- (b) La page qui permet d'ajouter un type
- (c) La page qui permet d'ajouter un lieu
- (d) La page qui permet d'afficher les agents
- (e) La page qui permet d'afficher les types
- (f) La page qui permet d'afficher les lieux
- (g) Interface d'accueil d'administrateur.

III.4.3. L'interface d'agent (Ajouter des problèmes) :

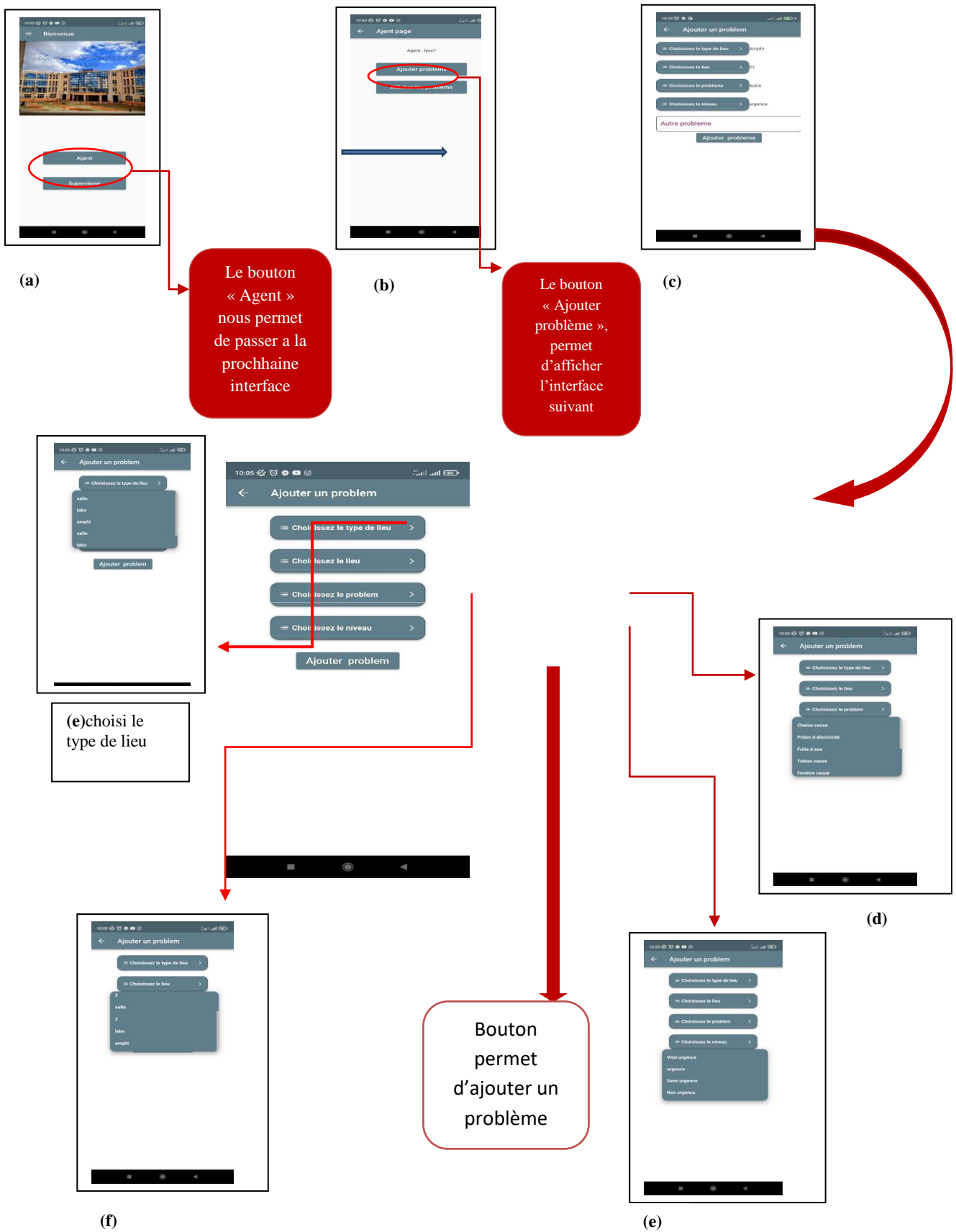


Figure 40: Interface d'utilisateur Agent (Ajouté des problèmes).

Chapitre III : Développement de l'application et de la maquette

Les résultats obtenus sont :

- (a) : Page d'accueil.
- (b) : Ajouter un probleme constaté.
- (c) : Ajouter un problème constaté.
- (d) : Choisir le problème observé.
- (e) : Choisir le niveau d'urgence.
- (f) : Choisir le lieu.
- (g) : Choisir le type de lieu.

III.4.4. Interface agent (liste des problèmes) :

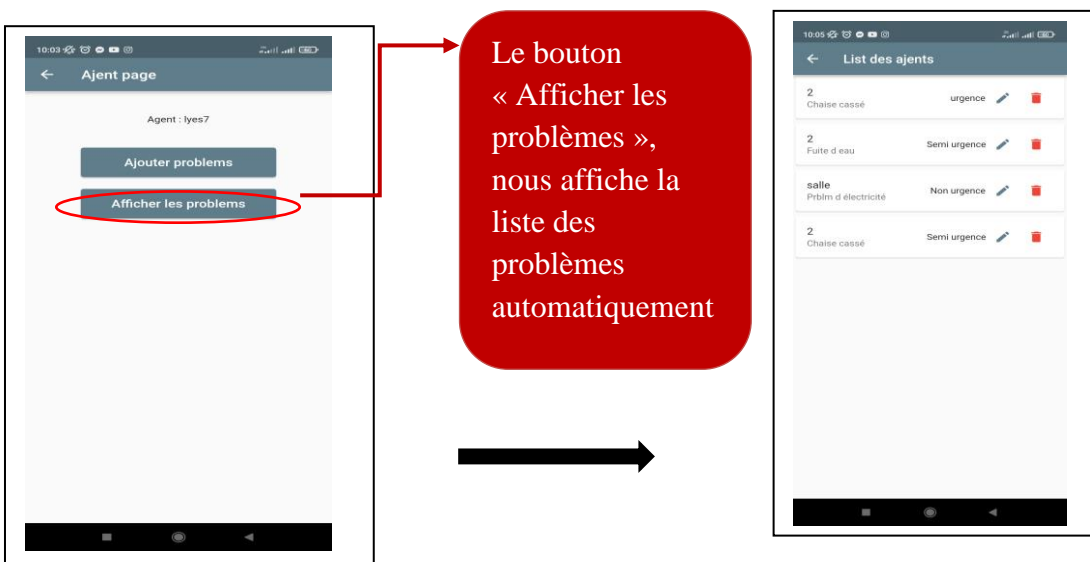
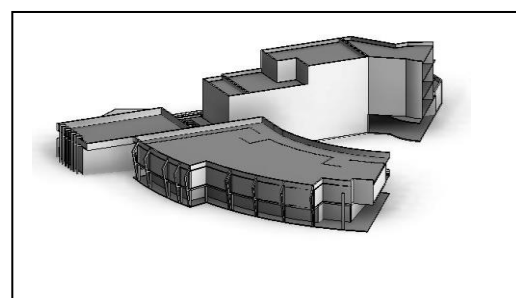
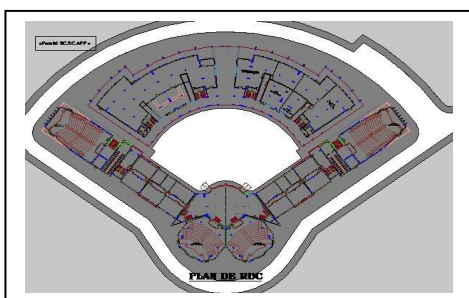


Figure 41: Interface agent et listes des problèmes.

III.5. Modélisation de la maquette :

Pour la modélisation de notre maquette on a utilisé l'Autocad comme un logiciel de conception assistée par ordinateur, qui nous a permet de créer notre maquette en 3D facilement.

Nous vous montrons les étapes de modélisation de la maquette de notre faculté ci-dessous :



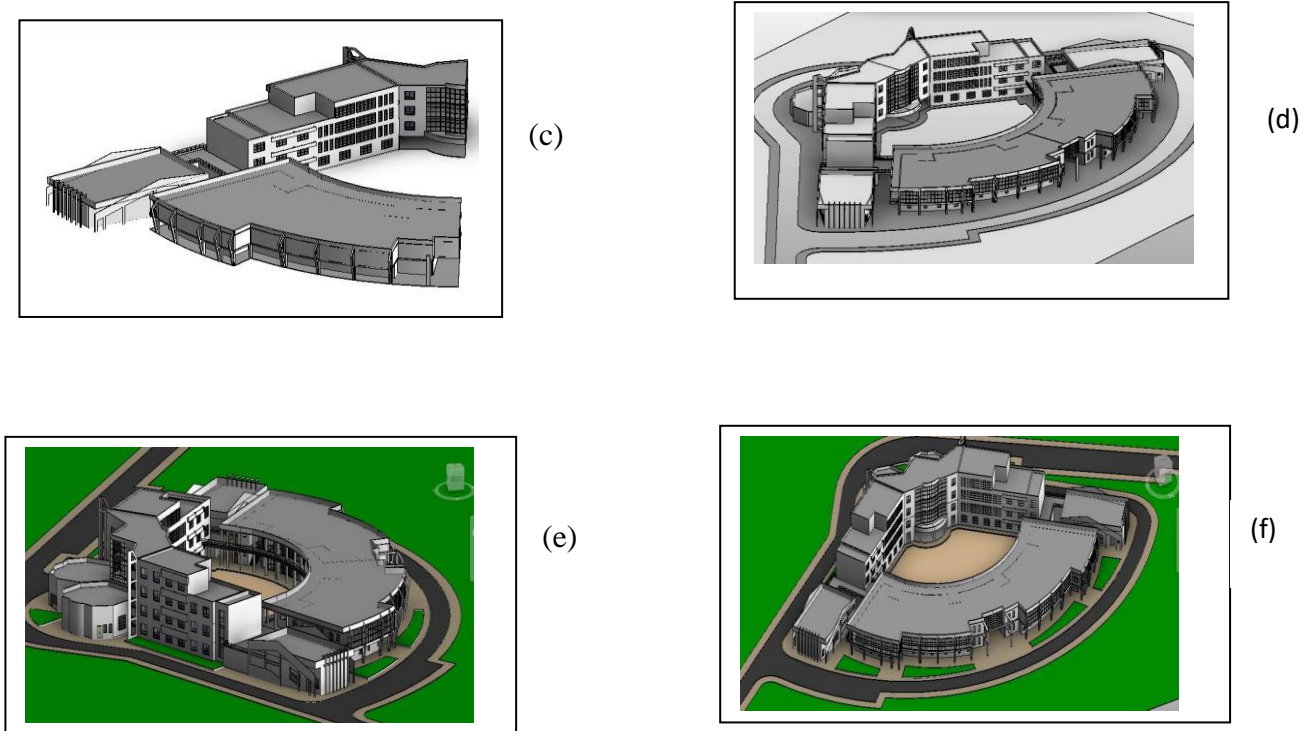


Figure 42: Modélisation de la maquette.

Les résultats obtenus sont :

(a) : Vu en plan d'un bloc de la faculté.

(b) : *Modélisation du plan en 3D.*

(c), (d) : *L'ajoute des fenêtres et des étages et des endroits extérieurs.*

(e), (f) : Coloration des façades et des endroits.

III.6. Résultat final et vision sur la maquette de la faculté :



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

Figure 43: Résultat final et vision sur la maquette de la faculté

Les résultats obtenus sont :

- (a) , (b): vision sur les façades externe de la maquette.
- (c) , (d) : vision interne de la maquette (la cour de la faculté).
- (e), (f) : vision d'en haut de la faculté.
- (g), (h): Vision de la FSSA.

III.7. Conclusion :

Les résultats préliminaires obtenus par la mise en œuvre de l'application proposée sont présentés dans ce dernier chapitre. La maquette 3D réalisée sous le logiciel Autocad est aussi présentée. Cependant, nous signalons que seule une partie de la FSSA est donnée ici en 3D, car la contrainte du facteur temps nous a fait défaut, et par conséquent, cette partie fera l'objet d'une perspective de ce PFE.

Conclusion générale et perspectives

Conclusion générale et perspectives

Conclusion générale et perspectives :

Ce projet nous a permis de concevoir, développer et intégrer une application mobile sous la plateforme Android qui a fortement occupé le marché des terminaux mobiles.

La réalisation de ce travail est organisée en trois principales étapes :

- Nous avons commencé par une étude préalable de notre projet sur les différents systèmes d'exploitation et spécialement le système Android en étudiant son architecture, ses versions et ses outils de développements. Ensuite nous avons pris une étude générale sur la maquette.

- Puis en à passé à la conception de notre application, nous avons utilisé le langage de programmation Flutter pour la partie l'application et l'environnement de développement Android Studio pour la réalisation, et le langage Autocad et le Revit pour la maquette.

- Enfin, nous avons réussi à réaliser un outil simple et efficace qui aide l'utilisateur a suivre d'une façon informatique les problèmes de la faculté des sciences et sciences appliqués de la wilaya de Bouira.

L'outil réalisé est très satisfaisant. Néanmoins, plusieurs perspectives restent envisageables, comme mettre un lien entre l'application mobile de la FSSA et sa maquette 3D ...

Cependant, nous avons rencontré un certain nombre d'obstacles qui ont empêché une amélioration complète de notre application, Parmi les obstacles rencontrés:

- Le modèle 3D pour visualisation uniquement sous Revit, le modèle n'est pas modifiable.
- La communication sans rapport avec Revit et Android Studio pour transfère les fichiers modèles directement vers l'espace de développement mobile (programmation).
- Il n'y a aucune relation et plate-forme pour toute communication entre les deux logiciels de développement.

De là, nous constatons que nous ne pouvons pas le faire pour les raisons mentionnées ci-dessus, et nous laissons des parenthèses pour les étudiants qui choisissent de poursuivre cette étude.

Références bibliographique :

- [1] **BENYOUCEF, L et BOUAMARA, N. Conception et réalisation d'une application mobile multiplateforme pour la gestion des ventes. Thèse de doctorat. Université Mouloud Mammeri. Tizi Ouzou. 2022**
- [2] **BEKKAYA, H et MEZIANE, S. développement d'une application de géolocalisation des médecins de la wilaya de Tlemcen sous Android. Projet de fin d'étude master II. Université de Tlemcen. 2017**
- [3] **<http://www.inflexsys.com> Consulté le 25 mars 2022.**
- [4] **<http://www.ionos.fr>. Publier le 07 mars 2019. Consulter le 20 Mai 2022.**
- [5] **<http://fr.dreamstime.com>. Consulter le 23 mai 2022.**
- [6] **<http://www.story.backMarket.fr>. Publier février 2021. Consulter le 23 Mai 2022**
- [7] **<http://www.edition-ellipses.fr>. Consulter le 18 Mai 2022.**
- [8] **<http://www.vectorstock.com>. Consulter le 18 Mai 2022.**
- [9] **<http://www.lojiciel.com>. Consulter le 18 mai 2022**
- [10] **<http://www.microsoft.fandom.com> Consulter mai 2022**
- [11] **<http://www.seekpng.com>. Consulter le 20 mai 2022**
- [12] **<http://www.Apple.com>. Consulter le 19 mai 2022**
- [13] **<http://www.dspace.univ-tlemcen.dz>. Consulter le 20 mai 2022**
- [14] **<http://www-igm.univ-mlv.fr> consulté Consulter le 23 mai 2022.**
- [15] **<http://www.DeveloperTools.com>. Publier le 19 aout 2012. Consulter le 19 Mai 2022.**
- [16] **<http://www.larousse.fr>. Consulter le 23 mai 2022**
- [17] **<http://maquettesarchitecture.fr> Consulté juin 2022**
- [18] **<http://www.faire-construire-maison> Consulté juin 2022**

[19] MAESTRI Annalisa, premiers pas en BIM. L'essentiel en 100 pages, paris : Edition Afnort Eyrolles, P.6, Ouvrage.,2017

[20] RENAUD, M. Création automatique d'une maquette numérique 3D d'un bâtiment à partir d'un relevé scanner laser 3D, P [22-24], MÉMOIRE, CNAM (CONSERVATOIRE NATIONAL DES ARTS ET METIERS), ÉCOLE SUPÉRIEURE DES GÉOMÈTRES ET TOPOGRAPHES ,2016

[21] <http://definir-tech.com> Consulté juin 2022

[22] Programmation mobile avec Android, pierre Narzic, 2022

[23] <http://developer.android.com> Consulté juin 2022

[24] <http://www.greenspector.com> Consulté juin 2022

[25] <http://Fr.Acervolima.com>consulté Consulter juin 2022

[26] www.redbubble.com Consulté juin 2022

[27] <http://www.Devstory.net> Consulté juin 2022

[28] <http://dart.dev> Consulté juin 2022

[29] <http://blog.back4app.com> Consulté juin 2022

[30] www.dataykarot.net Consulté juin 2022

[31] <http://www.arkance-systems.be> Consulté juin 2022

[32] wikipedia

المخلص

الهدف من مشروع نهاية الدراسة هو الرصد و التشخيص عن طريق تطبيقنا للعديد من الصعوبات التقنية للكلية العلوم و العلوم التطبيقية.

نقترح اقتراح تطبيق رقمي مضمن على هاتف ذكي استنادا إلى اندرو ويد ستيديو و الأوتوكاد و الريفيت في مخطوطة مشروع نهاية الدراسة.

وهي تتمثل في تسجيل جميع المشكلات التي يلاحظها الوكيل الفني ، ثم يتم إرسالها إلى المسؤول للمتابعة والتحكم ، أو حتى لاحتياجات الإحصاءات متوسطة المدى.

Résumé

Le suivi et le diagnostic par moyens informatiques des multiples difficultés techniques de la FSSA est l'objectif de ce projet de fin d'études (PFE).

La création d'une application numérique embarquée sur un smartphone, basée sur Android Studio, Autocad et le Revit, est proposée sur ce manuscrit de PFE. Elle consiste à enregistrer l'ensemble des problèmes observés par un Agent technique, puis transmise à un administrateur pour le suivi et le contrôle, ou même pour les besoins de statistiques à moyen terme.

Abstract

The monitoring and diagnosis by computer means of the multiple technical difficulties of the FSSA is the objective of this end-of-study project.

The creation of a digital application embedded on a smartphone, based on Android Studio, Autocad and Revit, is proposed on this end-of-study project manuscript. It consists in recording all the problems observed by a Technical Agent, then transmitted to an administrator for follow-up and control, or even for the needs of medium-term statistics.