



REPUBLIQUE DU BENIN



**MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE (MESRS)**

**ECOLE NATIONALE D'ECONOMIE APPLIQUEE ET DE
MANAGEMENT (ENEAM)**

MEMOIRE DE FIN DE FORMATION

Filière : Informatique de Gestion III

Option : DTS/AIP

THEME

Mise en place d'une application de
Numérisation du processus
d'enregistrement des
mensurations des clients

Présenté par : HOUSINOU Abed-Négo F. et LEWHE Habib J-E

Sous la direction de :

Maître de stage :

M. Fabrice KIKI

Développeur mobile à
acumen

M. Karoll POFAGI

Designer UI/UI

Maître de rapport :

Maurice COMLAN

Enseignant à l'ENEAM

ANNEE ACADEMIQUE : 2018 -2019

DEDICACES

A nos très chers parents

Aucun mot ne pourra exprimer nos sentiments et notre gratitude envers vous

A nos frères et sœurs

Nous vous souhaitons beaucoup de bonheur et de réussite

A tous ceux que nous aimons

Nous dédions le fruit de notre travail.

REMERCIEMENTS

Au terme de ce travail, nous remercions particulièrement Mme Rosaline D. WOROU HOUNDEKON, Professeur Agrégée des Sciences de Gestion, Directrice de l'ENEAM, et M. Théophile K. DAGBA, PhD., Maître de Conférences, Directeur adjoint, Chargé des affaires académiques de l'ENEAM, pour leur implication dans le cours mon apprentissage scolaire.

Nous tenons à remercier sincèrement monsieur Maurice COMLAN, qui a accepté de nous encadrer et qui s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, pour sa générosité, ses conseils qui ont été bien utiles, et pour la grande patience dont il a su faire preuve malgré ses charges professionnelles.

Nous exprimons également notre gratitude aux membres du jury qui m'ont honoré par leur présence et leur acceptation d'évaluer mon travail. Nous n'oublions surtout pas nos maitres de stage Mr Karol Eddy POFAGI et Mr Fabrice KIKI pour leur contribution, leur soutien et leur patience. Enfin, son adressée nos plus chaleureux remerciements à tous nos proches et amis, qui nous ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Merci à toutes et à tous.

RESUME

Dans ce mémoire, nous réalisons une application de numérisation du processus de prise de mesures par les couturiers qui améliore le système actuel d'enregistrement des mesures (enregistrement physique des mesures) par la plupart des couturiers au Bénin. Ce projet vise majoritairement à participer à l'intégration des Tics dans le quotidien des couturiers Béninois qui visiblement ne s'en servent pas dans le cadre de leur travail. Cette application permet donc aux couturiers de pouvoir disposer d'une base de données portable contenant les mesures de leurs différents clients, de disposer d'un catalogue numérique pour la publicité de leurs réalisations et de pouvoir également suivre les commandes de leurs clients. Cela permet de jouir dans une certaine mesure des avantages qu'offre la technologie au 21ème siècle et de renforcer les relations entre client et couturier.

Mots-clés : numérisation, catalogue, couturiers, mesures

ABSTRACT

In this memory, we carry out a scanning application of the seams that improves current system of measurements (physical recording of measurements) by most seams in the Benin. This project aims to participate in the Beninese dressing that visibly do not use it in the context of their work. This application allows the fashion designer to be able to have a portable database containing the measures of their different customers, to have a digital catalog for advertising their achievements and can also follow the controls of their clients. This allows to enjoy a certain extent of the benefits of technology in the 21st century and to strengthen relations between client and tailor.

Keywords : scanning, catalog, fashion designer, measurements

LISTE DES FIGURES

Figure 1 : Organigramme de Acumen Network	3
Figure 2 : Diagramme des cas d'utilisation du système.....	13
Figure 3 : Diagramme de classe du système	22
Figure 4 : Diagramme d'objets du système	23
Figure 5 : Diagramme de séquence du cas « créer client ».....	24
Figure 6 : Diagramme de séquence du cas « créer commande »	25
Figure 7 : Diagramme de séquence du cas « publier modèle »	26
Figure 8 : Diagramme d'activité du cas « créer client »	27

LISTE DES TABLEAUX

Tableau 1 : Tableau d'identification des acteurs et des cas d'utilisation.....	12
Tableau 2 : Description textuelle du cas « créer un client ».....	13
Tableau 3 : Description textuelle du cas « créer une commande ».....	15
Tableau 4 : Description textuelle du cas « créer catalogue ».....	16
Tableau 5 : Description textuelle du cas « Publier un modèle »	17
Tableau 6 : Dictionnaire des données	19

Table des matières

DEDICACES.....	i
REMERCIEMENTS	ii
RESUME.....	iii
ABSTRACT	iv
INTRODUCTION.....	1
Chapitre 1 : Présentation du cadre générale de travail	2
I. Présentation du lieu de travail	3
1. Présentation de Acumen Network.....	3
1.1. Historique et mission	3
1.2. Organigramme.....	3
1.3. Déroulement du stage.....	4
II. Etude préliminaire	5
1. Présentation de l'existant.....	5
2. Critique de l'existant.....	6
3. Proposition de solution : une application de numérisation de la prise de mesure par les couturiers	6
a. Objectif principal.....	6
b. Objectifs spécifiques	7
Chapitre 2 : Analyse et Spécification des besoins.....	8
I. Spécification des besoins	9
1. Spécification des acteurs.....	9
2. Spécification fonctionnelle	9
3. Spécification non-fonctionnelle.....	10
4. Description fonctionnelle.....	10
II. Analyse oriente objet	11
1. Les cas d'utilisation	12
1.1. Identification des acteurs et des cas d'utilisation du système.....	12
1.2. Diagramme des cas d'utilisation	13
1.3. Description textuelle de quelques cas d'utilisation.....	13
2. Les Classes	18

2.1.	Dictionnaire des données du système	19
2.2.	Règles de gestion du système	21
2.3.	Diagramme de classes	21
3.	Diagrammes d'objets.....	22
4.	Diagrammes de séquences	23
5.	Diagrammes d'état transitions.....	26
Chapitre 3 : Implémentation du système		28
I. Outils de développement utilisés.....		29
1.	Langages et Framework utilisés	29
1.1.	Flutter.....	29
1.2.	DART comme langage utilisé.....	30
2.	SGBD (Système de Gestion de Base de Données)	31
2.1.	Firebase	31
2.2.	SQLite.....	32
3.	Outils de développement.....	32
3.1.	Modelio.....	32
II. Sécurité de l'application.....		34
III. Travaux réalisés.....		34
CONCLUSION.....		37
PERSPECTIVES		38
BIBLIOGRAPHIE & WEBOGRAPHIE		39
ANNEXE.....		40

INTRODUCTION

L'artisanat occupe une place de choix dans le travail informel au Bénin. En effet, il contribue à hauteur de 13% du PIB du Bénin et occupe la troisième position dans l'économie nationale après l'agriculture et le commerce. Parmi les métiers de l'artisanat, la couture retient particulièrement l'attention. Ce métier est le choix par excellence des jeunes qui, pour diverses raisons, n'arrivent pas à évoluer dans les études. Ce choix se justifie par le fait que ce travail nécessite de la part de l'artisan, les bases de l'écriture et de la lecture. Aussi, ce métier ne nécessite pas une force physique très intense. Mais force est de constater qu'à un moment ou le numérique se répand dans nos vies et dans presque tous les secteurs d'activités que les Tics ne sont pas utilisés dans le métier de la couture au Bénin. Il n'existe aucune proposition application dédiée aux artisans couturiers et adapter au contexte de notre pays. Difficile de se faire connaître et d'accroître leurs visibilité.

C'est dans ce contexte que le présent projet s'inscrit dans la dynamique de concevoir une application ayant pour but de numériser tout le processus d'enregistrement des clients tout en garantissant aux artisans la portabilité de leurs registres, la sauvegarde en ligne, la restauration de leur fichier de même que de nombreuses possibilités pour les autres acteurs à savoir : les clients, les vendeurs de pagne, les stylistes et l'Etat lui-même.

Le présent mémoire structuré en trois (3) chapitres expose le travail d'analyse et d'implémentation pour la mise en place du système. Le premier chapitre aborde la présentation du contexte d'étude à travers la présentation de la structure d'accueil et la présentation du thème. Le deuxième chapitre expose l'analyse et la spécification des besoins effectués pour ledit système. Enfin le troisième chapitre présente l'aspect sécuritaire du système ainsi que les outils utilisés pour sa réalisation.

Chapitre 1 : Présentation du cadre générale de travail

I. Présentation du lieu de travail

1. Présentation de Acumen Network

1.1. Historique et mission

Acumen Network est une entreprise franco-béninoise créée en 2016. Elle est spécialisée dans l'apport d'un soutien holistique en vue d'accompagner les ambitions de transformations technologiques et digitales de ses clients.

De façon plus spécifique, l'entreprise propose des services dans les domaines de l'expertise business et de l'expertise dans la mise en œuvre de projets IT. Elle propose également des formations et des coachings.

1.2. Organigramme

Le fonctionnement d'Acumen network suit la hiérarchie décrite par l'organigramme ci-dessous :

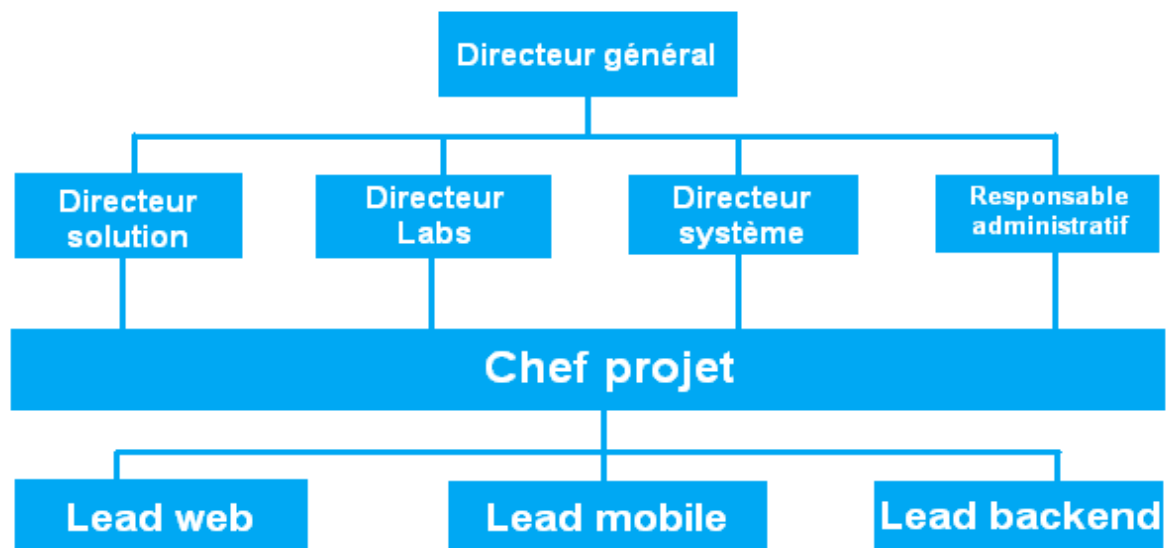


Figure 1 : Organigramme de Acumen Network

1.3. Déroulement du stage

Notre stage à Acumen a débuté le 27 octobre 2019 et a pris fin le 27 janvier 2020. Durant cette période, hormis le développement de notre application, nous avons eu l'opportunité de participer à certains travaux au sein de l'entreprise en réalisant les tâches qui nous étaient confiées. Nous avons aidé à peupler la base de données utilisée pour tester la plateforme digitale Agrosfer. Nous avons également été formé à l'utilisation de quelques technologies et outils tels que le Framework Java Spring pour la création des micro services et des application web, les technologies Java et Flutter pour le développement d'applications mobiles et iOS, le Framework frontend Angular pour le développement web et le logiciel de design (UI/UX design) Figma pour la réalisation des interfaces utilisateurs.

1.4. Problèmes rencontres et proposition de solution

Pendant notre stage nous avons été confrontés à un problème majeur qui est celui de la connexion internet qui n'était pas très stable provoquant le crash de certains tests et des difficultés à télécharger des dépendances ou tout autres outils nécessaire au bon déroulement de notre travail. De plus cela rendait impossible l'utilisation de certains outils comme Figma qui nécessite constamment la connexion.

L'instabilité de la connexion étant due aux retards de paiement des factures d'électricité et au fournisseur d'accès internet de la structure, nous proposons comme solution de :

Payer une recharge supplémentaire pour ne pas être surpris par l'épuisement de la recharge en cours ;

Opter pour un fournisseur d'accès internet plus fiable et offrant un meilleur service après-vente.

II. Etude préliminaire

1. Présentation de l'existant

L'accès aux métiers de l'artisanat en général et à celui de la couture en particulier est conditionné par une formation professionnelle reçue par les apprentis dans les ateliers des maîtres artisans. Ces formations sont de plus en plus règlementées par des structures telles que la Fédération Nationale des Artisans du Bénin (FENAB) et le ministère en charge de la formation professionnelle. Elles se soldent pour la plupart par l'obtention d'un diplôme de fin de formation qui atteste de la maîtrise du métier. Tout au long de sa formation le couturier est appelé à prendre les mesures de ses clients pour diverses commandes, une étape très importante pour une parfaite réalisation de son travail afin de garantir la satisfaction de ses clients. Pour ce faire le couturier béninois se sert de son outil de mesure (le mètre) et de son registre de prise de mesure, très souvent un cahier dont il se sert pour enregistrer les mesures de ses clients en fonction de leur commande. C'est en quelque sorte sa base de données physique, elle lui permet de ne plus avoir à reprendre les mesures d'un client déjà enregistré et de faire des mises à jour au besoin. Par ailleurs, toujours dans le cadre de la promotion des Tics en vue de la valorisation des métiers de l'artisanat il existe plusieurs organismes au Bénin et partout dans le monde qui par diverses applications mobiles et web participent à cela. Nous pouvons citer par exemple Metiers.africa une application qui permet de recenser les artisans au Bénin afin de connecter les particuliers et les artisans.

2. Critique de l'existant

Au Bénin le métier de la couture reste encore non informatisé. Cette pratique habituelle qu'est l'utilisation d'un support physique pour l'enregistrement des mesures des clients par les couturiers béninois ne saurait s'avérer efficace compte tenu du nombre important de clients à enregistrer tous les jours. Ce nombre ne cessant de s'accroître quand un cahier se termine, le couturier devra en prendre un autre. Quand il doit mettre à jour les mesures d'un client ou rechercher simplement les mesures, c'est des minutes de recherches dans les différents registres puis c'est des ratures ou l'usage de plusieurs pages pour un même client. Cet état de chose au 21^{ème} siècle ne saurait perdurer avec le développement des solutions numériques.

3. Proposition de solution : **une application de numérisation de la prise de mesure par les couturiers**

Afin d'accompagner les couturiers béninois, et de participer à l'intégration de l'usage des Tics dans le cadre de leur travail la solution que nous proposons consiste en une application mobile multiplateforme de numérisation de la prise des mesures des clients par les couturiers. Cette solution s'inscrit dans le cadre de pouvoir développer une application dédiée aux couturiers et également adaptée au contexte béninois.

a. Objectif principal

Le présent projet a donc pour objectif d'intégrer l'usage des TIC dans le quotidien des couturiers du Bénin. Cet objectif peut être décliné en plusieurs objectifs spécifiques que l'on peut encore qualifier de résultats attendus.

b. Objectifs spécifiques

Comme objectifs spécifiques nous avons :

- ❖ La mise en place d'une application multiplateforme pour :
 - L'inscription des artisans couturiers par type et par spécialité ;
 - L'enregistrement en ligne des mesures des clients ;
 - La localisation des couturiers par un api de localisation intégrer à l'application
 - L'enregistrement des commandes des clients ;
- ❖ L'intégration de l'usage de l'application dans le quotidien des couturiers ;

Chapitre 2 : Analyse et Spécification des besoins

I. Spécification des besoins

1. Spécification des acteurs

Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système dans le but de réaliser une plus-value. Les acteurs en interaction avec notre système sont :

- Le couturier

Cet usager accède à l'information complète du système afin de gérer ses clients.

- L'administrateur

Il est le déclencheur de toute action dans le système. Il se charge de gérer la partie de la base de données. Il tient à jour les données des couturiers, caractéristique des mesures, infos sur les clients des couturiers.

2. Spécification fonctionnelle

Le système à mettre en place devra permettre au couturier de réaliser un certain nombre d'actions à savoir :

- ✓ L'enregistrement en ligne des mesures de leurs clients
- ✓ La création des commandes sur demande des clients
- ✓ La création d'un catalogue comportant ses plus belles réalisations
- ✓ La publication de ses modèles au besoin
- ✓ L'administration du système

3. Spécification non-fonctionnelle

A part les besoins fondamentaux, notre système doit répondre aux critères

suivants :

LA SÉCURITÉ ET L'INTÉGRITÉ DES DONNÉES : l'accès à l'application n'est permis qu'après une phase d'authentification sécurisée. Les informations d'authentification doivent être confidentielles. L'application doit garantir l'intégrité, la cohérence et la persistance des données.

LA RAPIDITÉ DE TRAITEMENT : En effet, vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.

LA PERFORMANCE : Notre logiciel doit être avant tout performant c'est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des usagers d'une manière optimale.

LA CONVIVIALITÉ : Le futur logiciel doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.

4. Description fonctionnelle

Le système à mettre en place doit être performant et sécurisé. Il doit permettre entre autres de :

- Créer un client

Cette fonctionnalité permet au couturier de récupérer grâce à un formulaire des informations de base sur un client ainsi que toutes les mesures possibles pouvant être utile au couturier.

- Créer une commande

Cette fonctionnalité permettra au couturier de créer une commande suite à la demande d'un client. Ceci sous-entend donc le fait que ce client soit au préalable enregistré dans la liste des clients du couturier.

➤ Créer un catalogue

Cette fonctionnalité permettra au couturier d'ajouter à son catalogue des modèles qu'il estime faire partir de ses plus belles réalisations.

➤ Publier des modèles

Cette fonctionnalité permet au couturier de publier au besoin les modèles qui se situent dans son catalogue. Il devra renseigner ses informations ainsi que celles en rapport avec le modèle qu'il veut publier.

II. Analyse orientée objet

Pour atteindre notre objectif et répondre aux besoins des utilisateurs, il est nécessaire de choisir une bonne méthode de modélisation. Il existe plusieurs techniques de modélisation et plus globalement des processus de développement.

UML est un langage de modélisation objet unifié. Il est né de la fusion de trois méthodes orientées objet Booch, OMT et OOSE.

UML 2 est une évolution majeure du langage. La modélisation et la formalisation à l'aide d'un vocabulaire standardisé et de surcroît orienté objet confèrent à la méthode tout son intérêt. La formalisation et la modélisation facilitent en effet la définition du problème à traiter et la compréhension par l'ensemble des principales parties prenantes.

UML 2.3 propose 14 types de diagrammes. Ces diagrammes sont répartis en deux types : les diagrammes comportementaux ou dynamiques et les diagrammes structurels ou statiques. Nous ne présenterons que 5 d'entre eux

dans ce travail d'analyse. Il s'agit des diagrammes de cas d'utilisation, de classe, d'objet, de séquence et d'état transition.

1. Les cas d'utilisation

1.1. Identification des acteurs et des cas d'utilisation du système

Notre application intègre différentes fonctionnalités accessibles aux différents acteurs identifiés. Ainsi, après analyse des différents besoins fonctionnels, nous retenons les cas d'utilisation ci-après :

Tableau 1 : Tableau d'identification des acteurs et des cas d'utilisation

Acteurs	Cas d'utilisation
Couturier	<ul style="list-style-type: none">• Gréer un client<ul style="list-style-type: none">▪ Créer client▪ Modifier client▪ Supprimer client• Créer une commande<ul style="list-style-type: none">▪ Créer commande▪ Modifier commande▪ Supprimer commande• Créer catalogue<ul style="list-style-type: none">▪ Ajouter modèle▪ Supprimer modèle• Publier modèle
Administrateur	Administrer le système

1.2. Diagramme des cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Il décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. La figure suivante illustre le diagramme de cas d'utilisation du système :

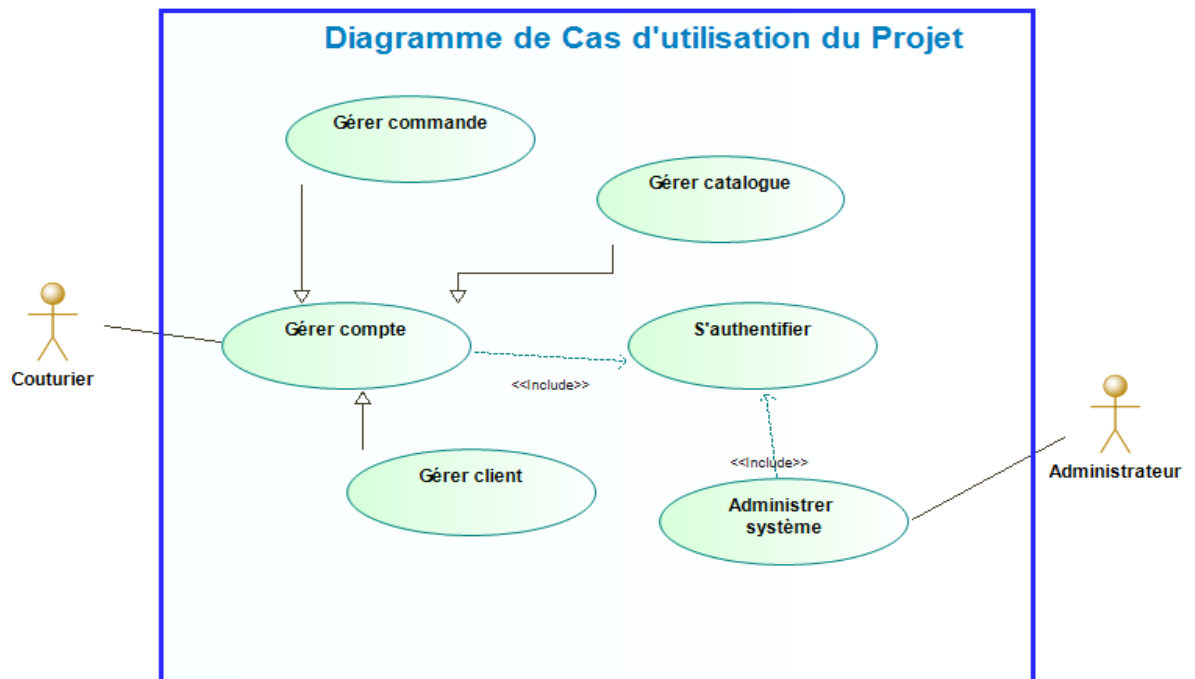


Figure 2 : Diagramme des cas d'utilisation du système

1.3. Description textuelle de quelques cas d'utilisation

Cas n°1 : Créer un client

Tableau 2 : Description textuelle du cas « créer un client »

SOMMAIRE D'IDENTIFICATION	
Titre :	Créer un client
Auteur :	Abed-Nego & Habib

Résumé :	Ce cas d'utilisation permet aux utilisateurs de créer des clients. Il s'agira de renseigner toutes les informations importantes à l'identification de chaque client, les informations de bases ainsi que les mensurations.
Acteur:	Couturier
Date de création :	26/02/2020
Version :	1.0
DESCRIPTION DES ENCHAINEMENTS	
Pré condition	Post condition
L'utilisateur doit s'être authentifié	Le système confirme l'envoi du formulaire par l'incrément de la liste des clients enregistrés.
Enchaînement nominal	
<ol style="list-style-type: none"> 1. L'utilisateur demande la page de création d'un client 2. Le système affiche le formulaire de création du client 3. L'utilisateur remplit le formulaire puis le soumet à la validation du système 4. Le système effectue des vérifications obligatoires sur l'ensemble des données renseignées et les enregistre dans la base de données si celles-ci sont correctes 	
Enchaînement alternatif	
<ol style="list-style-type: none"> 3-a) L'utilisateur décide de suspendre le remplissage du formulaire 3-b) L'utilisateur peut interrompre l'envoi du formulaire en quittant brusquement l'application 4-a) Les données renseignées ne sont pas correctes 4-b) Le système affiche un message d'erreur 4-c) Le cas d'utilisation se termine par un échec 	

Fin	Ergonomie et Performances
Scénario nominal aux étapes 2 et 3 sous décision de l'utilisateur	Le formulaire doit être simple et compréhensible par l'utilisateur.

Cas n°2 : Créer une commande

Tableau 3 : Description textuelle du cas « créer une commande »

SOMMAIRE D'IDENTIFICATION	
Titre :	Créer une commande
Auteur :	Abed-Nego & Habib
Résumé :	Ce cas d'utilisation permet aux utilisateurs de créer des commandes.
Acteur:	Couturier
Date de création :	26/02/2020
Version :	1.0
DESCRIPTION DES ENCHAINEMENTS	
Pré condition	Post condition
L'utilisateur doit s'être authentifié en tant que couturier	Le système confirme l'envoi du formulaire par l'incrément de la liste des commandes créées.
Enchaînement nominal	
<ol style="list-style-type: none"> 1. L'utilisateur demande la page de création d'une commande 2. Le système affiche le formulaire de création d'une commande 3. L'utilisateur remplit le formulaire puis le soumet à la validation du système 4. Le système effectue des vérifications obligatoires sur l'ensemble des données renseignées et les enregistre dans la base de données si celles-ci sont correctes 	

Enchaînement alternatif	
<p>3-a) L'Utilisateur décide de suspendre le remplissage du formulaire</p> <p>3-b) L'utilisateur peut interrompre l'envoi du formulaire en quittant brusquement l'application</p> <p>4-a) Les données renseignées ne sont pas correctes</p> <p>4-b) Le système affiche un message d'erreur</p> <p>4-c) Le cas d'utilisation se termine par un échec</p>	
Fin	Ergonomie et Performances
Scénario nominal aux étapes 2 et 3 sous décision de l'utilisateur	Le formulaire doit être simple et compréhensible par l'utilisateur.

Cas n°3 : Créer un catalogue

Tableau 4 : Description textuelle du cas « créer catalogue »

SOMMAIRE D'IDENTIFICATION	
Titre :	Créer un catalogue
Auteur :	Abed-Nego & Habib
Résumé :	Ce cas d'utilisation permet aux utilisateurs de créer leur catalogue de produit tout en y ajoutant des modèles.
Acteur:	Couturier
Date de création :	26/02/2020
Version :	1.0

DESCRIPTION DES ENCHAINEMENTS	
Pré condition	Post condition
L'utilisateur doit s'être authentifié en tant que couturier	Le système confirme l'ajout du modèle par l'incrémentation de la liste des modèles sur l'interface du catalogue.
Enchaînement nominal	
<ol style="list-style-type: none"> 1. L'utilisateur clique sur le bouton d'ajout d'un nouveau modèle 2. Le système propose différentes options soit la prise directe d'une photo du modèle soit le choix depuis la galerie 3. L'utilisateur choisit l'option qui lui convient puis exécute l'action 4. Le système récupère la photo et l'affiche dans la liste à la suite des autres photos s'il y en avait déjà 	
Enchaînement alternatif	
<p>3-a) L'Utilisateur décide de suspendre l'ajout du modèle</p> <p>3-b) L'utilisateur peut interrompre le cas en quittant l'application</p> <p>4-a) Le système n'affiche pas la photo</p>	
Fin	Ergonomie et Performances
Scénario nominal aux étapes 2 et 3 sous décision de l'utilisateur	Le formulaire doit être simple et compréhensible par l'utilisateur.

Cas n°4 : Publier un modèle

Tableau 5 : Description textuelle du cas « Publier un modèle »

SOMMAIRE D'IDENTIFICATION	
Titre :	Publier un modèle
Auteur :	Abed-Nego & Habib
Résumé :	Ce cas d'utilisation permet aux utilisateurs de publier des modèles situés dans leur catalogue.

Acteur:	Couturier
Date de création :	26/02/2020
Version :	1.0
DESCRIPTION DES ENCHAINEMENTS	
Pré condition	Post condition
L'utilisateur doit s'être authentifié en tant que couturier	Un nouveau modèle est publié et apparaît à l'accueil si l'utilisateur est connecté à internet
Enchaînement nominal	
<ol style="list-style-type: none"> 1. L'utilisateur sélectionne le modèle qu'il souhaite publier puis clique sur le bouton publier 2. Le système affiche le formulaire de publication 3. L'utilisateur renseigne les informations demandées puis valide 4. Le système effectue sur les différents champs renseignés des vérifications et enregistre ensuite la publication si celles-ci sont correctes 	
Enchaînement alternatif	
<p>2-a) L'Utilisateur décide de suspendre la publication</p> <p>2-b) L'utilisateur peut interrompre le cas en quittant l'application</p> <p>4-a) Les données renseignées ne sont pas correctes , le système renvoi un message d'erreur</p>	
Fin	Ergonomie et Performances
Scénario nominal aux étapes 2 et 3 sous décision de l'utilisateur	Le formulaire doit être simple et compréhensible par l'utilisateur.

2. Les Classes

2.1.Dictionnaire des données du système

Tableau 6 : Dictionnaire des données

Termes	Attributs	Définitions	Types
Couturier	IdCouturier	Identifiant d'un couturier	N
	NomCouturier	Nom d'un couturier	A
	PrenomCouturier	Prénom d'un couturier	A
	Sexe	Sexe d'un couturier	A
	TypeCouturier	Type de couturier	A
	SpecialiteCouturier	Specialite d'un couturier	A
	TelephoneCouturier	Numero de telephone d'un couturier	N
	LocalisationCouturier	Localisation géographique d'un couturier	A
Commande	IdCommande	Identifiant d'une commande	N
	DateCommande	Date d'une commande	Date
	MontantCommande	Montant d'une commande	N
	DateLivraison	Date de livraison d'une commande	Date
Tenue	IdTenue	Identifiant d'une tenue	N
	imgTissu	Image du tissu	A
	imgModele	Image du modèle	A
	typeModele	Type de modèle	A
	prixTenue	Prix d'une tenue	N
	Observation	Observation en rapport avec une tenue	A
	IdModele	Identifiant d'un modele	N

Modèle	nomModele	Nom d'un modele	A
	imgModele	Image d'un modele	A
	typeModele	Type d'un modele	A
	genreModele	Genre d'un modele	A
	note	Note d'un modele	N
Client	idClient	Identifiant d'un client	N
	nomClient	Nom d'un client	A
	prenomClient	Prenom d'un client	A
	sexeClient	Sexe d'un client	A
	teintClient	Teint d'un client	A
	telephoneClient	Téléphone d'un client	N
	avatarClient	Photo d'un client	A
	H_poitrine	Hauteur poitrine	N
	Lg_corsage	Longueur corsage	N
	Lg_buste	Longueur buste	N
	Lg_robe	Longueur robe	N
Compte	idCompte	Identifiant d'un compte	N
	Login	Adresse email du compte	A
	Mot de passe	Mode de passe du compte	A
	StatutCompte	Statut du compte	Bool
	Avatar	Photo de profil du compte	A

Légende : A pour **Alphabétique**, N pour **Numérique**, AN pour **Alphanumérique**, Bool pour **Booléen**, Date pour **Date**

2.2. Règles de gestion du système

Les règles de gestion ci-après régissent le fonctionnement de notre système :

RG 1 : un couturier peut créer plusieurs clients

RG 2 : une commande ne peut être passée que par un client et un seul

RG 3 : une commande peut concerner plusieurs tenues

RG 4 : un couturier peut publier plusieurs modèles

RG 5 : un modèle ne peut être publié que par un couturier et un seul

RG 6 : un couturier peut créer plusieurs comptes

RG 7 : un compte ne peut appartenir qu'à un couturier unique

2.3. Diagramme de classes

Le diagramme de classe décrit clairement la structure du système en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. Les classes issues de l'analyse du système sont regroupées dans le diagramme ci-dessous avec les relations qui les lient :

Diagramme de classe

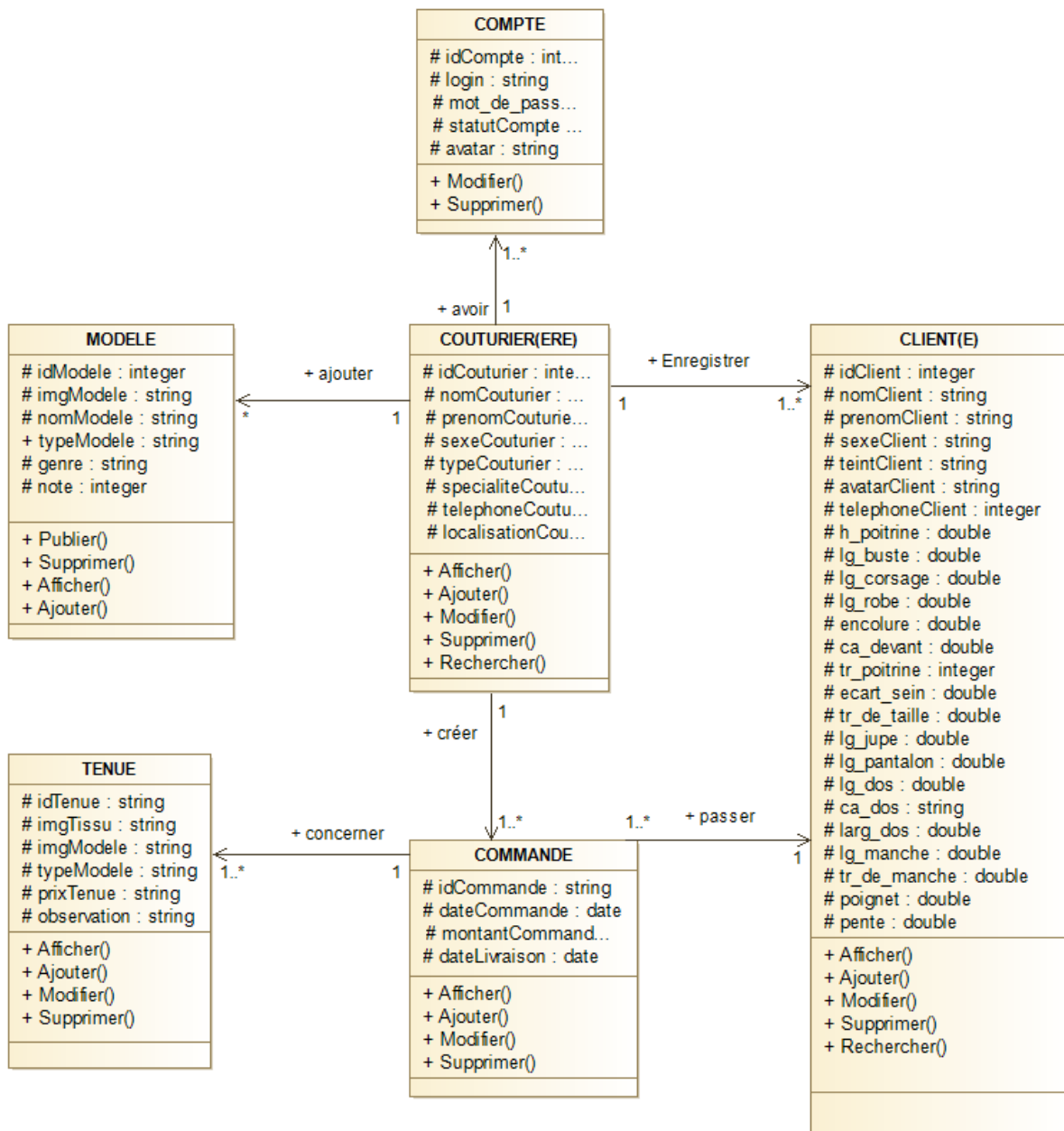


Figure 3 : Diagramme de classe du système

3. Diagrammes d'objets

Un diagramme d'objet permet d'avoir une vue d'ensemble du système. Il représente une instance spécifique du diagramme de classe à un moment précis.

Réalisons les diagrammes d'objets entre les classes Commande et Client d'une part, Client et Couturier d'autre part et enfin Commande et Tenue :

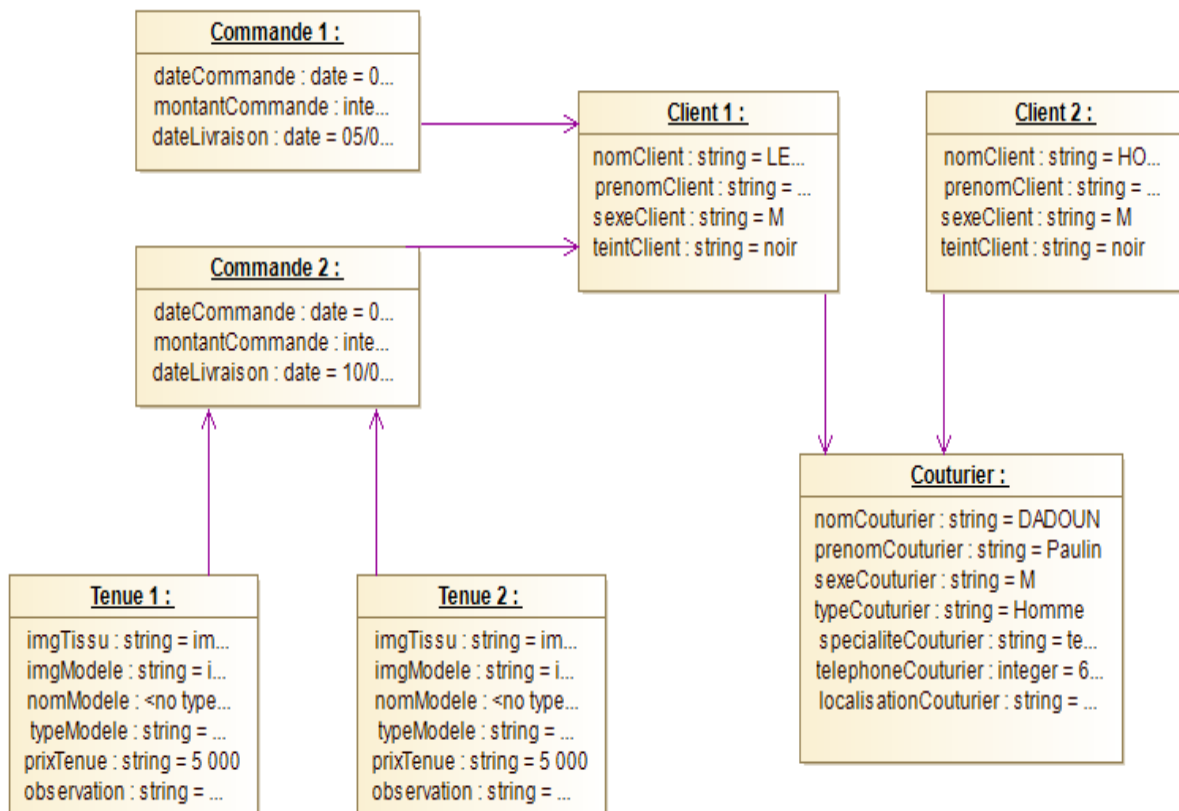


Figure 4 : Diagramme d'objets du système

4. Diagrammes de séquences

Les diagrammes de séquences permettent de décrire **COMMENT** les éléments du système interagissent entre eux et avec les acteurs. Les objets au cœur d'un système interagissent en s'échangeant des messages. Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

Nous avons représenté les diagrammes de séquences de quelques cas d'utilisation :

*Cas d'utilisation : créer un client

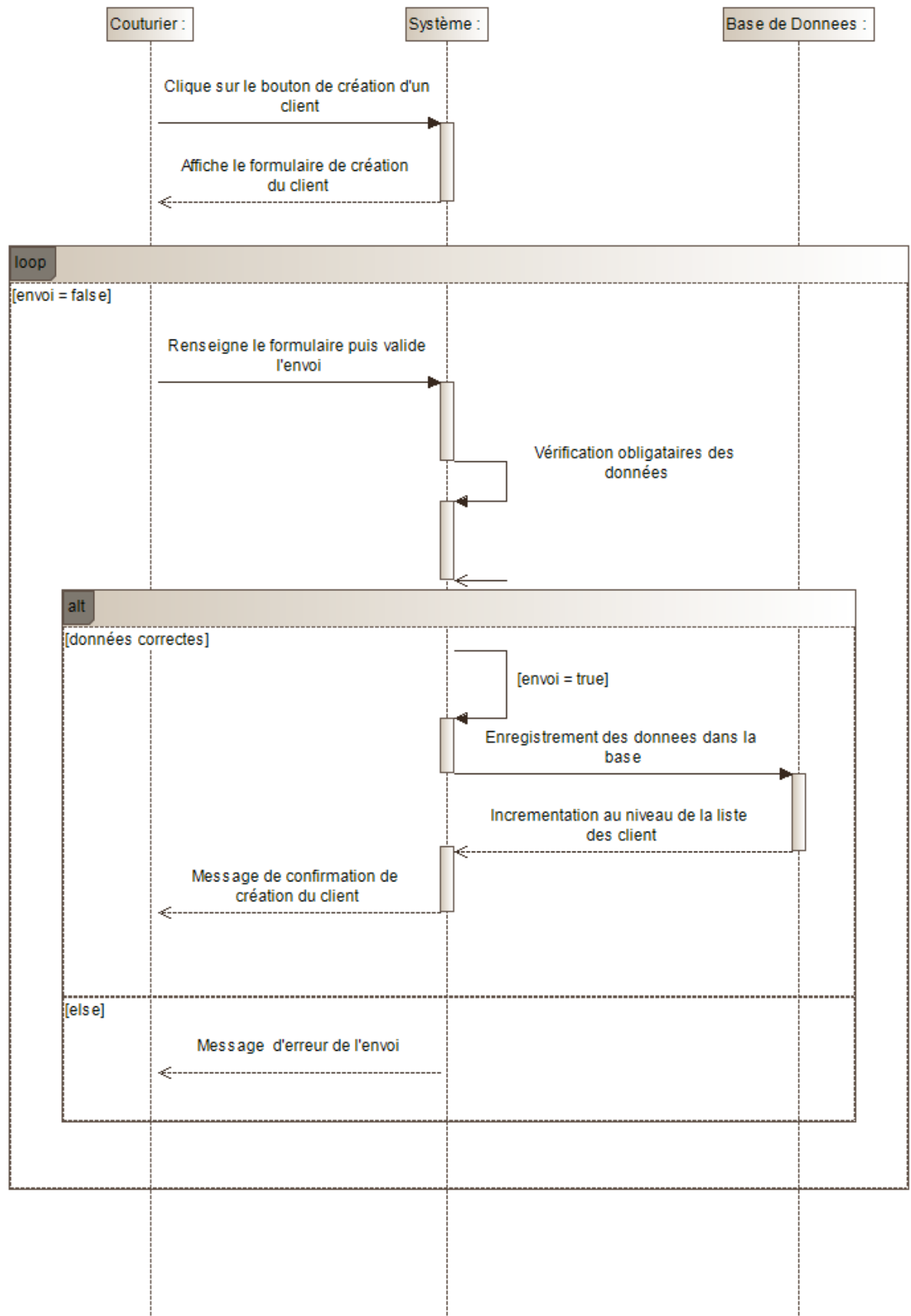


Figure 5 : Diagramme de séquence du cas « créer client »

*Cas d'utilisation : créer commande

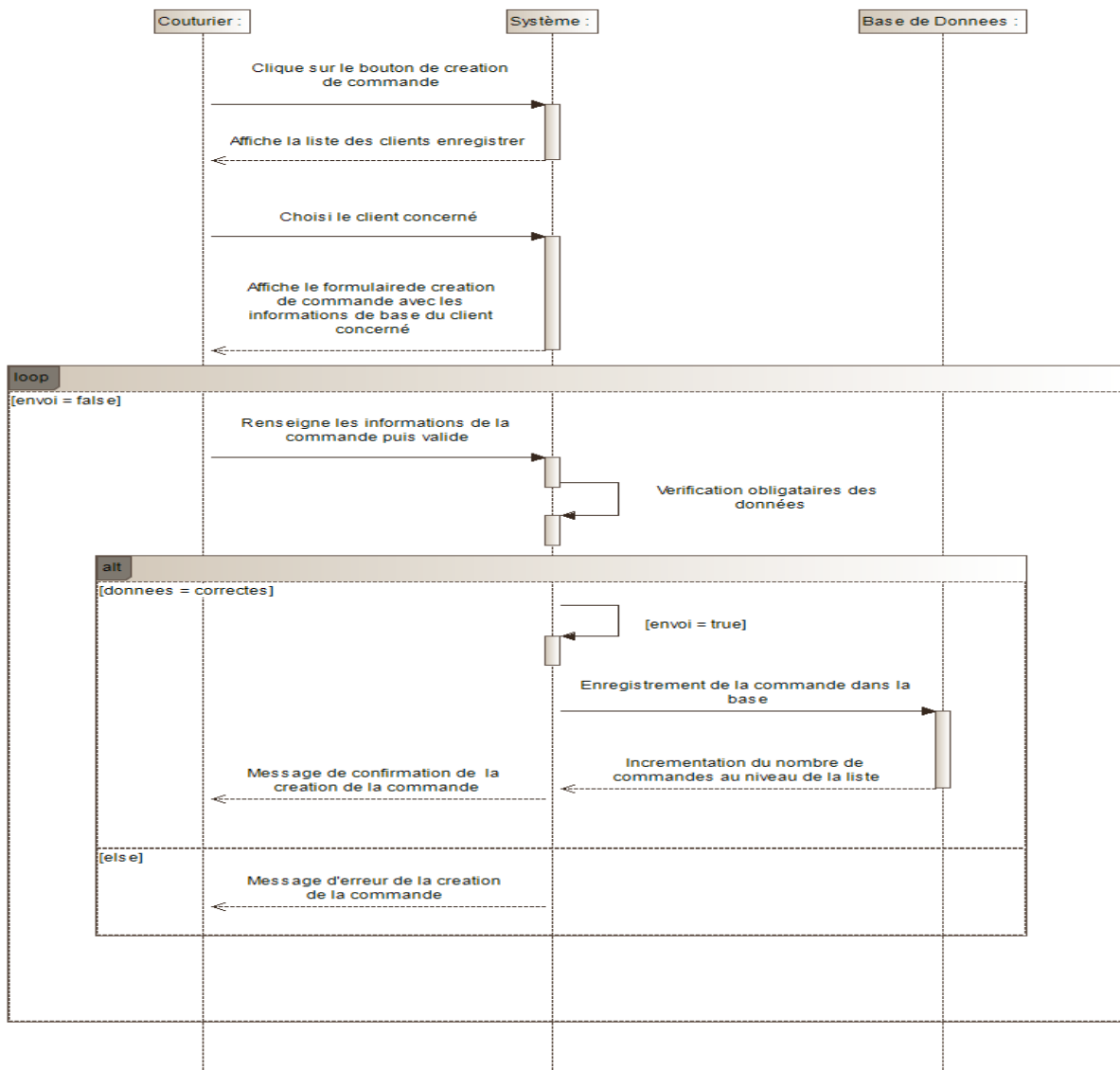


Figure 6 : Diagramme de séquence du cas « créer commande »

*Cas d'utilisation : Publier modèle

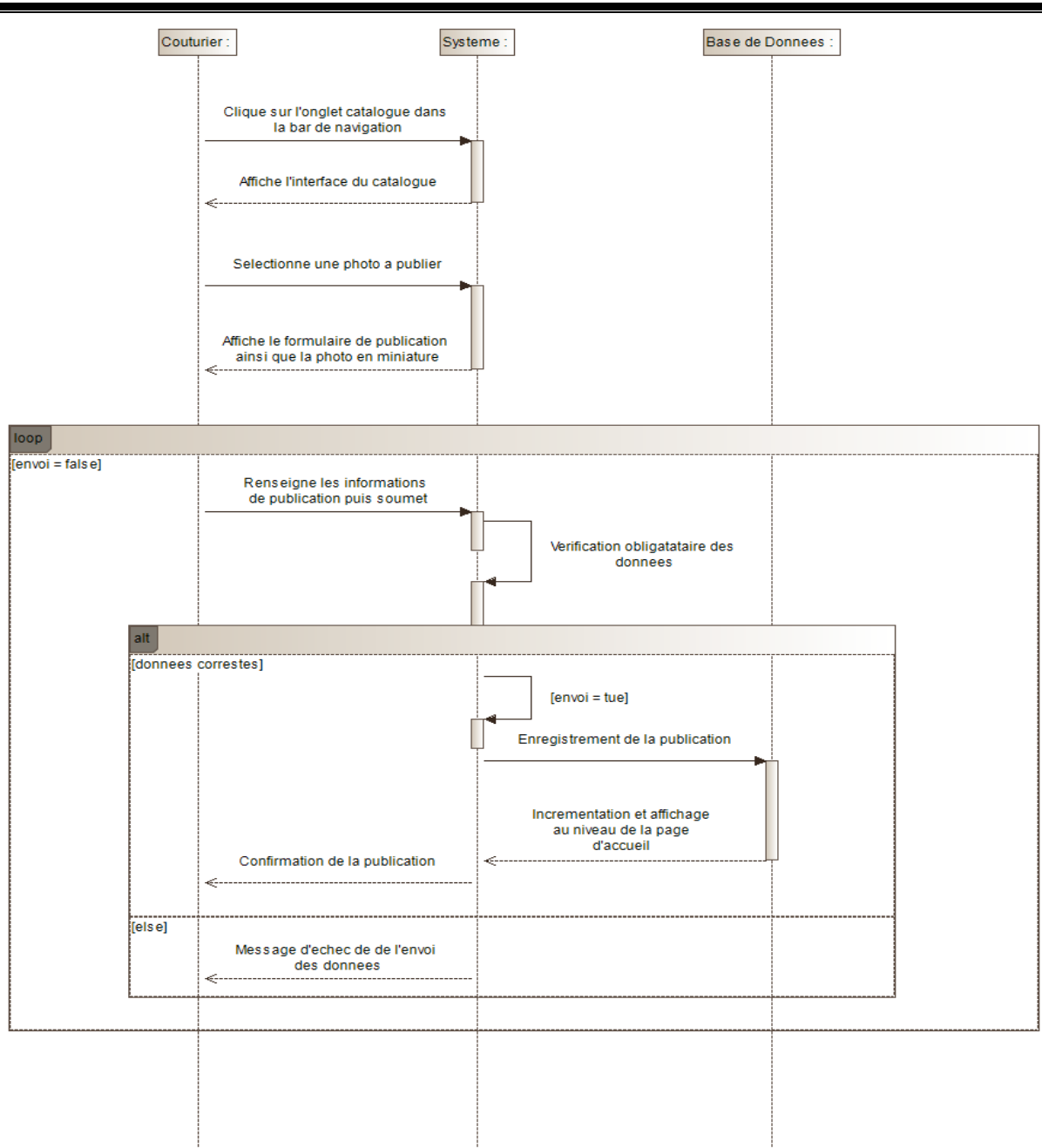


Figure 7 : Diagramme de séquence du cas « publier modèle »

5. Diagrammes d'activités

Le diagramme d'activités représente les activités que réalisent un ou plusieurs objets. Il peut correspondre à la description en détail d'une activité du diagramme d'états transitions, à la description d'une méthode. Il peut également décrire l'activité d'un système ou d'un sous-système en assignant les

responsabilités à chaque acteur. Le diagramme d'activités constitue aussi un bon choix pour décrire un cas d'utilisation.

Diagramme d'activités du cas « Créer un client » :

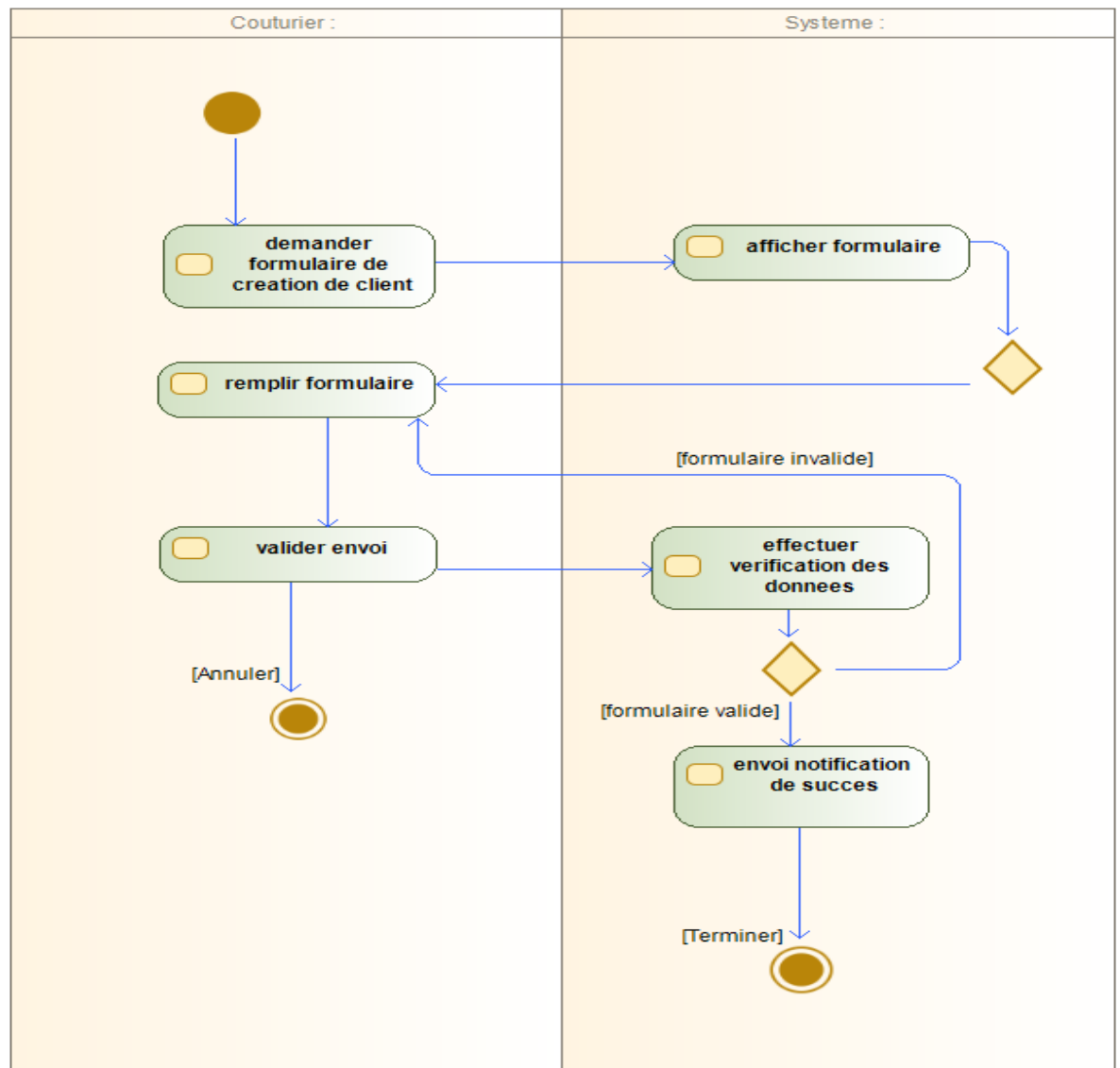


Figure 8 : Diagramme d'activité du cas « créer client »

Chapitre 3 : Implémentation du système

I. Outils de développement utilisés

1. Langages et Framework utilisés

1.1. Flutter



FLUTTER C'EST QUOI ?

- Flutter est un nouveau Framework créé par Google, construit en C, C++ et Dart, dont l'objectif est de nous proposer une nouvelle approche « Cross-Platform » de développement mobile. En se basant uniquement sur le langage Dart, il permet de construire des applications natives Android et iOS.

- Flutter se situe au même niveau que React, Native ou Xamarin, ce qui signifie qu'à partir du code écrit en Dart va ensuite être généré du code Android et iOS.

POURQUOI FLUTTER ?

- L'application communique avec la plateforme pour créer des widgets ou accéder à des services comme l'appareil photo. Le rendu des widgets est fait sur un Canvas et les événements sont renvoyés aux widgets. C'est une architecture assez simple, cependant, il faut créer des applications différentes pour chaque plateforme car les widgets et le langage de développement sont différents chez iOS et Android.

- Les premières plateformes multi-OS se sont basées sur le Javascript et les webviews. On pourra citer différents Framework : Titanium, PhoneGap, Cordova, Ionic...

- Flutter propose des vues dynamiques similaire à React Native. Ce Framework choisit cependant une approche différente pour réduire les problématiques de performance (dus au pont). Ainsi, Flutter a choisi d'utiliser Dart, un langage compilé (contrairement au JavaScript).

POINTS FORT DE FLUTTER

- Haute vitesse de développement avec des fonctionnalités comme le « hot reload », un Framework réactif, des widgets préexistants et des outils intégrés aux environnements de développement

- designs de qualité, personnalisables, reposant sur des widgets, une bibliothèque d'animations, une architecture en couches extensible

- expérience de haute qualité sur les différentes plateformes grâce à un cœur à haute performance et une prise en compte de l'interopérabilité des plateformes

• HOT RELOAD

Flutter implémente un système d'hot reload (rafraichissement à chaud) qui permet un rechargement de l'ordre de la seconde sur le mobile ou l'émulateur. Le hot reload est stateful, ce qui signifie que l'état de l'application est conservé après chaque rechargement.

- Le hot reload fonctionne en injectant le code mis à jour dans la machine virtuelle en cours d'exécution. Cela inclut non seulement l'ajout de nouvelles classes mais aussi l'ajout de méthodes et de champs aux classes existantes.

1.2. DART comme langage utilisé



Dart (initialement appelé Dash4) est un langage de programmation web développé par Google. Son but initial est de remplacer JavaScript pour devenir la nouvelle lingua franca du développement web, néanmoins la priorité actuelle des développeurs est que le code Dart puisse être converti en code JavaScript compatible avec tous les navigateurs modernes, ainsi que sur le développement d'application multi-plateforme. Dart peut aussi être utilisé pour la programmation côté serveur, ainsi que le développement d'applications mobiles (via l'API Flutter). Dart offre deux modes de fonctionnement. Le premier, nommé AOT (pour Ahead Of Time), permet de générer une application native pour chaque plateforme. Le deuxième mode de fonctionnement est dit JIT (Just-In-Time) et offre la fonctionnalité de Hot Reload lors des développements. L'idée du Hot Reload en Flutter est de corriger le problème du temps nécessaires entre chaque build en ne mettant plus que quelques millisecondes (voire secondes dans le pire des cas) entre chaque modification. Le développement de son application est alors bien plus rapide

2. SGBD (Système de Gestion de Base de Données)

2.1. Firebase



Firebase est une plateforme mobile créée en 2011 par James Tamplin et Andrew Lee, puis rachetée par Google en 2014 pour être intégrée à leur offre de services Cloud (Google Cloud Platform). L'objectif premier de Firebase est de vous libérer de la complexité de création et de la maintenance d'une architecture serveur, tout en vous garantissant une scalabilité à toute épreuve (plusieurs milliards d'utilisateurs) et une simplicité dans l'utilisation.

Sources : Google (www.firebaseio.com)

2.2. SQLite



SQLite est un SGBDR dit embarqué, c'est à dire qu'il n'est pas basé sur un serveur de base de données sur lequel des applications viennent se connecter. Il s'agit d'une simple librairie écrite en C qu'il suffit de lier à votre programme pour pouvoir y accéder. Cette librairie implémente les principales fonctions d'un SGBDR et est donc compatible avec le langage SQL mais avec certaines restrictions. Comme cette librairie est très légère et qu'elle ne nécessite ni installation ni configuration, elle est très utilisée par certains navigateurs web pour sauvegarder des données localement ou par les applications pour smartphones.

3. Outils de développement

3.1. Modelio



Successeur de l'atelier Objecter, Modelio est un outil de modélisation UML disponible sur plusieurs systèmes. Modelio Studio est un éditeur graphique utilisé pour étendre et personnaliser l'outil ... ou définir de nouveaux types de diagrammes qui supportent vos modèles.

Il propose une gamme d'outils étendant ses fonctionnalités permettant, entre autres, la mise en œuvre de l'approche MDA.

Nous nous en sommes servi pour reproduire différents les diagrammes utilisés pour l'analyse de notre système.

3.2. Figma



Figma est une application de conception UI et UX basée sur un navigateur, avec d'excellents outils de conception, de prototypage et de génération de code. Il s'agit actuellement (sans doute) du meilleur outil ; de conception d'interface du secteur, avec des fonctionnalités robustes qui prennent en charge l'équipe travaillante à chaque phase du processus de conception.

Nous nous en sommes servi pour la réalisation des maquettes de notre application.

3.1. Android Studio



Nous nous sommes servis de Android Studio comme environnement de développement intégré de développement pour notre application.

II. Sécurité de l'application

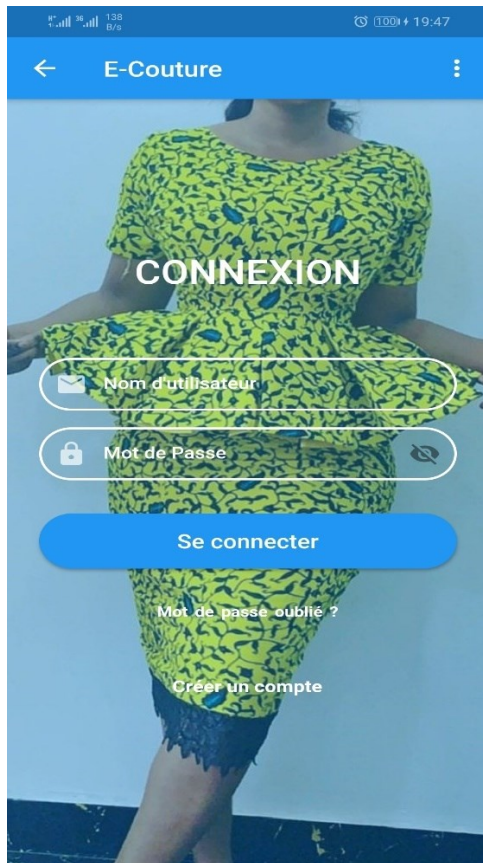
Compte tenu de l'importance et de la sensibilité des données traitées, il est nécessaire de s'assurer de la sécurité du système en prenant certaines mesures :

- L'accès à l'application requiert le renseignement d'un identifiant et d'un mot de passe permettant d'identifier l'acteur ;
- Les fonctionnalités réservées aux utilisateurs ayant un certain privilège sont cachées aux autres utilisateurs de sorte qu'ils en ignorent même l'existence ;
- Les mots de passes sont cryptés grâce à l'algorithme de chiffrement **bcrypt** avant d'être stockés dans la base de données ;

III. Travaux réalisés

Quelques captures qui montrent à quoi ressemble la version démo de notre SMI.

- ❖ Interface de connexion



Description : L'utilisateur entre son email et son mot de passe afin de pouvoir se connecter à la plateforme.

❖ Interface de création d'un client

MTN BENIN 12:30

Choisir un client



Nom : _____
Prénom : _____
Telephone : _____
Sexe : _____
Teint : _____

Mesures (cm)

H Poitrine : _____

Lg Jupe : _____

Lg Buste : _____

Lg Pantalon : _____

Lg Corsage : _____

Lg Dos : _____

Lg Robe : _____

Ca Dos : _____

Encolure : _____

Larg Dos : _____

Ca devant : _____

Lg Manche : _____

Tr Poitrine : _____

Tr de Manche : _____

Ecart Sien : _____

Poignet : _____

Tr de Taille : _____

Pente : _____

ACCUEIL

CLIENTS

COMMANDES

CATALOGUE

Description : Cette interface permet au couturier de renseigner les informations relatives à chacun de ses clients.

CONCLUSION

Ce stage académique effectué à Acumen nous a permis d'une part de mettre en application les connaissances théoriques reçues au cours de notre formation de trois (03) ans en Analyse Informatique et Programmation (AIP) et d'autre part d'avoir un aperçu des réalités dans le monde professionnel.

Ayant eu pour objectif de participer à la promotion des Tics dans les métiers de l'artisanat au Bénin en particulier celui de la couture qui a l'ère du numérique demeurent encore à la vieille époque, nous avons à travers ce projet développé une application qui numérise tout le processus de prise de mesure par les couturiers. Son but est de participer à l'intégration des Tics dans le quotidien des couturiers et dans le cadre de leur travail.

Cette application permet entre autres l'enregistrement en ligne des mesures des clients, la création d'un catalogue numérique, l'enregistrement des commandes des clients et la publication en ligne des modèles par les couturiers. Le travail d'analyse et de conception ayant abouti à la conception du système est basé sur le langage de modélisation UML. L'utilisation de diverses méthodes de cryptages comme le bcrypt ou le JWT à divers niveaux pour garantir la confidentialité des données et assurer la sécurité du système.

PERSPECTIVES

Tout le système étant appelé à évoluer dans le temps, des améliorations peuvent être apportées à ce travail afin de le rendre plus utile. On pourrait penser au développement d'un espace réserver aux clients enregistrés par les couturiers afin de leur permettre de suivre leurs commandes, de pouvoir visiter les différentes publications des clients, pouvoir entrer en contact avec leur couturier a travers la plateforme. Nous envisageons aussi la possibilité de permettre aux vendeurs de tissus de pouvoir à travers l'application faire la publicité de leurs articles. La nouvelle version de notre système prendra aussi en compte les insuffisances qui auraient été relevées dans l'utilisation de la première version.

BIBLIOGRAPHIE & WEBOGRAPHIE

- L'artisanat au Benin : Secteur ;laisse pour compte malgré ses 13% de contribution au PIB (<http://news.acotonou.com/h/88711>, consulter en Janvier 2020),
- Nomenclature des métiers de l'artisanat au Benin PDF (www.gouv.bj/download/, consulter en Janvier 2020)
- RAPPORT ARTISANAT_definiti INSAE (PDF)
- Benin : Metiers.africa pour connecter les particuliers et les petits artisans (<http://google.com/amp/s/amp.rfi>)
- [https://fr.wikipedia.org/wiki/Acteur_\(UML\)](https://fr.wikipedia.org/wiki/Acteur_(UML)) – consulté le 17/02/20
- <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation> – consulté le 17/02/20.
- <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes> consulté le 19/02/20.
- https://fr.wikipedia.org/wiki/Diagramme_de_séquence - consulté le 19/02/20.
- <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-etats-transitions> – consulté le 20/02/20.
- https://www.frandroid.com/android/535194_quest-ce-que-flutter-loutil-permettant-de-cree-des-applications-android-et-ios - consulté le 22/02/20.
- [https://fr.wikipedia.org/wiki/Dart_\(langage\)](https://fr.wikipedia.org/wiki/Dart_(langage)) - consulté le 22/02/20.
- <https://www.modelio.org/about-modelio/features.html>, consulté le 22/02/2020
- www.figma.com, consulté le 22/02/2020
- www.firebase.google.com, consulte le 23/02/2020

ANNEXE

Code source de quelques interfaces :

- Page d'accueil :

```
import 'package:flutter/material.dart';
import 'package:myapp/AccueilPage.dart';
import 'package:myapp/SousCategorie_Homme.dart';
import 'package:myapp/detailsPhotos.dart';

class ContenuAccueil extends StatefulWidget {
  @override
  _ContenuAccueilState createState() => _ContenuAccueilState();
}

class _ContenuAccueilState extends State<ContenuAccueil> {

  @override
  Widget build(BuildContext context) {
    List<GridModel> model = [
      new GridModel(name:"Homme", chemin:"assets/boubou-femme.jpg"),
      new GridModel(name:"Femme", chemin:"assets/boubou-femme.jpg"),
      new GridModel(name:"Couples", chemin:"assets/boubou-femme.jpg"),
      new GridModel(name:"Enfants", chemin:"assets/boubou-femme.jpg")
    ];
    return Column(
      children: <Widget>[
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Container(
              padding: EdgeInsets.symmetric(vertical: 15.0,
horizontal: 15.0),
              height: MediaQuery.of(context).size.height-565.9,
              width: MediaQuery.of(context).size.width,
              child:
                Card(
                  elevation: 2.0,
                  child:
                    Container(
                      padding: EdgeInsets.symmetric(vertical: 7.0),
                      child: Image(
                        image: AssetImage('assets/boubou-femme.jpg'),
                        fit: BoxFit.contain,
                      ),
                    ),
                ),
            ),
          ],
        ),
        Container( //Les gridTilesS
          margin: EdgeInsets.only(top: 10.0),
          height: MediaQuery.of(context).size.height-365,
```

```

        child: GridView.count(
          crossAxisCount: 2,
          padding: EdgeInsets.all(16.0),
          childAspectRatio: 0.9,
          crossAxisSpacing: 25.0,
          mainAxisSpacing: 25.0,
          children:
            List.generate(model.length, (index)
            {
              return
                GestureDetector(
                  child:
                    GridTile(
                      child: model[index],
                    ),
                  onTap: () {
                    if(model[index].name == 'Homme'){
Navigator.of(context).pushNamed(SOusCategorieHomme.tagHomme);
                    }
                    else if(model[index].name == 'Femme'){
Navigator.of(context).pushNamed(SOusCategorieHomme.tagFemme);
                    }else if(model[index].name == 'Couples'){
Navigator.of(context).pushNamed(SOusCategorieHomme.tagCouples);
                    }else{
Navigator.of(context).pushNamed(SOusCategorieHomme.tagEnfants);
                    }
                  },
                ),
            ),
          ],
        );
      }
    }

class GridModel extends StatelessWidget {
  String name;
  String chemin;

  GridModel({this.name, this.chemin});

  @override
  Widget build(BuildContext context) {
    return Container(
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(7.0),
        boxShadow: [
          BoxShadow(
            color: Colors.black45,
            blurRadius: 2,
          )
        ]
      ),
      child:
        Stack(

```

```
children: <Widget>[
  Container(
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(7.0),
      image: DecorationImage(
        image: AssetImage(chemin),
        fit: BoxFit.fill
      )
    ),
  ),
  Positioned( // la bande sur les gridTiles
    bottom: 0.0,
    child:
      Container(
        width: 151.5,
        height: 30.0,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.only(bottomRight:
Radius.circular(7.0), bottomLeft: Radius.circular(7.0)),
          color: Colors.black.withOpacity(0.45)
        ),
        child:
          Text(name,
            textAlign: TextAlign.center,
            style: TextStyle(
              fontSize: 20.0,
              fontFamily: 'Roboto Medium',
              color: Colors.white
            ),
          ),
        ),
      ),
    ),
  ],
),
);
}
```