

Introduction

Le présent document explicite la démarche du projet modélisé dans le cadre de notre formation BTS SLAM.

1 Organisation du projet

1.1 Contexte

- Une société souhaite ouvrir une antenne de maintenance PC
- Vendre des matériels sur catalogue via un de ses fournisseurs
- Elle crée la société de distribution de matériel informatique PC DM

Le processus de maintenance se déroule ainsi : Le client se présente avec son matériel. Sa demande de réparation est étudiée. (Le professionnel prend des notes sur une petite fiche). S'il s'agit d'un matériel sous garantie, la remise en état est effectuée gratuitement et un bon de prise en charge (BPC) sous garantie est immédiatement édité. Dans le cas contraire, un devis est établi et la réparation du matériel intervient seulement après l'accord du client sur ce devis. Cet accord déclenche la facturation, le client paye immédiatement, ce qui provoque l'acquittement de sa facture et l'édition d'un bon de réparation (BR). A défaut d'accord, la procédure est abandonnée et le client repart avec son matériel. Il faudra enregistrer le client avec un identifiant : un numéro par exemple.

1.2 Objectif

Délai de livraison : le produit doit être remis le 14 avril 2022.

1.3 Mission

Élaborer textuellement le processus de maintenance.

Établir une modélisation de ce processus.

Proposer un maquettage de l'application.

Réaliser le processus.

1.4 Conduite du projet

1.4.1 Organisation du groupe

Les parties prenantes du projet : l'équipe de documentation, l'équipe bdd, l'équipe programmation.

1.4.2 Planning

Nom	Prénom	Groupe Modélisation	Groupe Base de Données	Groupe Codage Java
GODONOU	Lauriane	X	-	-
Guirado	Léo	-	-	X
Marcel	Stephane	X	-	-
Bouguerra	Zakaria	-	X	-
EL BALI	Youssef	-	X	-
GANDEGA	Samba	X	-	-
DAN	Xavier	-	-	X
BOINA HADJI	Chakrane	X	-	-
NANA TCHUIGOUA	Sydney	X	-	-
Messaoudi	Mohamed	-	X	-
	Amina	-	-	X
	Dimitri	-	-	X
	Bawgitan	-	X	-
	Eli	-	X	-

a. un tableau de bord permettant de suivre l'avancement du projet

~~c. Présentation issue de vote veille concernant les sujets suivants : // Sans objet~~

- ~~l'intégration continue d'une application~~

- ~~Liste et comparatif des outils de déploiement.~~

- ~~La présentation reprendra les éléments principaux du projet auquel vous ajouterez un bilan personnel sur cette version.~~

- ~~La démonstration du fonctionnement lors de votre livraison~~

[illegible]

Avancement	Valeur à saisir	Valeur à saisir	Livrable
En avance	1	1	Pas de difficultés
Conforme au prévu	2	2	Quelques difficultés
En retard	3	3	Grandes difficultés

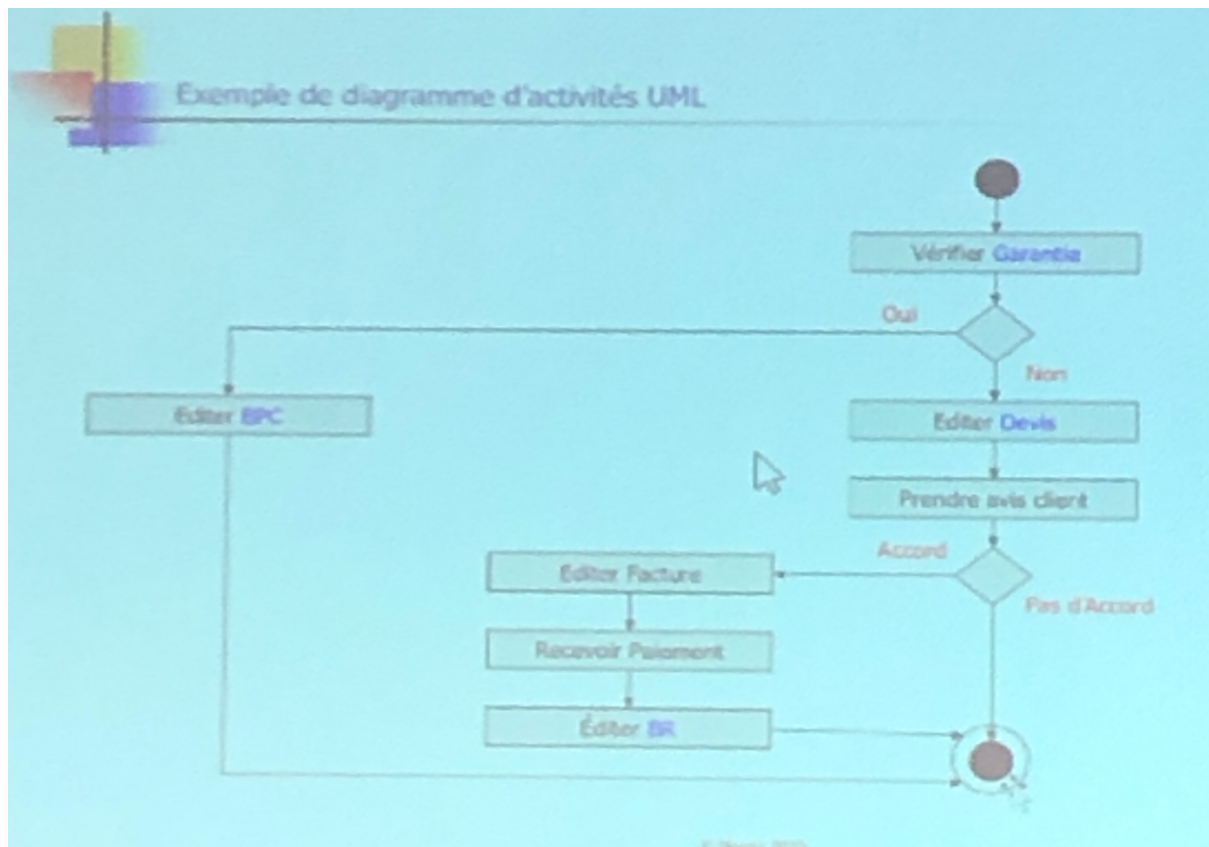
Lauriane GODONOU

Commentaires :	
Réalisé	D4M - fin modélisation
À venir	Fin
Point dur et solution	
Risque	Moyenne
	Critiqué
Cotation Risque	Moyenne
	Bonne

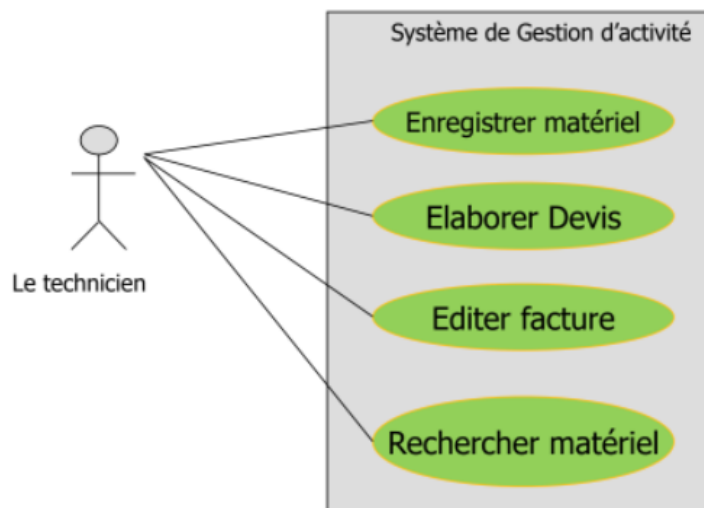
2.1 Le besoin

Création d'une IHM permettant la saisie de données par un technicien informatique

2.1.1 Texte et Diagramme Activité



2.1.2 Diagramme des cas d'utilisation (DUC)



2.2 UC1

2.2.1 (Analyse) Spécification du besoin

2.2.1.0 Dictionnaire de données

N° DEVIS

Date d'enregistrement
Nom, Prénom
Code, Article, pHT, Quantité, Montant HT -> information persistante
Total HT, Tva, Total TTC
Date accord

N° BPC
Nom, Prénom
Garantie N°
Date enregistrement matériel
Date de restitution

N° BR
Nom, Prénom
N° Facture (réglée)
Date enregistrement matériel
Date de restitution

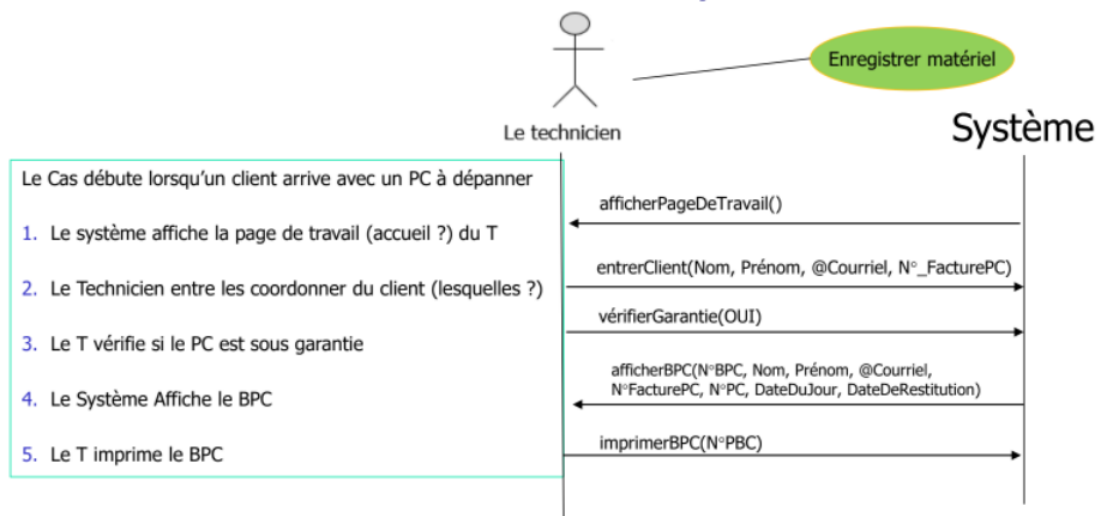
2.2.1.1 User story

En tant que <technicien>, je veux <enregistrer le matériel> afin de <...>.
En tant que <technicien>, je veux <élaborer un devis> afin de <...>.
En tant que <technicien>, je veux <éditer une facture> afin de <...>.
En tant que <technicien>, je veux <rechercher un matériel> afin de <...>.
En tant que <technicien>, je veux <lister les Devis refusés> afin de <...>.

2.2.1.2 DSS du scénario nominal

1. Le système affiche d'accueil du T
2. Le T vérifie si garantit == oui :
3. Le T saisie donnée client (nom, prénom, email, numéro de tel)
4. Le système affiche le BPC
5. Le T édite le BPC

DSS précisant le scénario nominal



2.2.1.3 Maquette IHM et enchainement

Ihm1 Accueil					
N° Facture PC	<input type="text"/>	Garantie	<input type="button" value="Oui"/>	N° BPC	<input type="text"/>
1ere Information Client pour diagnostic					
Nom Client	<input type="text"/>	<input type="text"/>			
Prénom Client	<input type="text"/>	<input type="text"/>			
Adresse	<input type="text"/>	Code postal	<input type="text"/>	Ville	<input type="text"/>
Devis N°	<input type="text"/>	Accord Client	<input type="button" value=""/>		
Facture N°	<input type="text"/>	Payée	<input type="button" value=""/>	N° BR	<input type="text"/>
Date de restitution F(réparation terminée)				<input type="button" value="Valider"/>	
N° Bon Recherché				<input type="text"/>	

Editer BPC

Editer Devis

Editer Facture
Editer BR

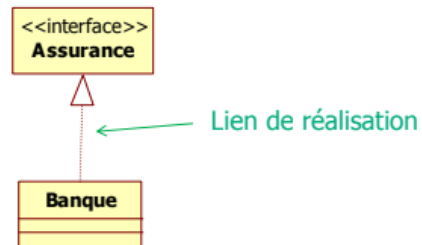
Rechercher BPC
Rechercher BR

2.2.1.4 Diagramme de classe

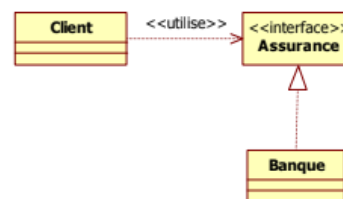


Les diagrammes de classes

Les interfaces



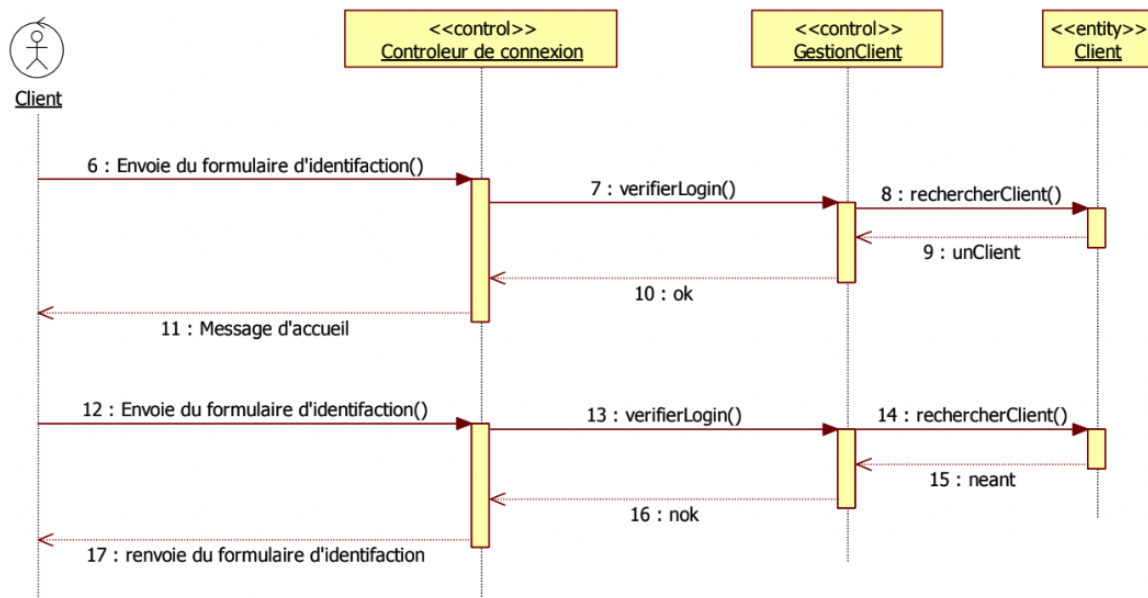
Les relations d'utilisation



~~2.2.1.5 Stéréotype de Jacobson (architecture MVC) // Sans objet~~

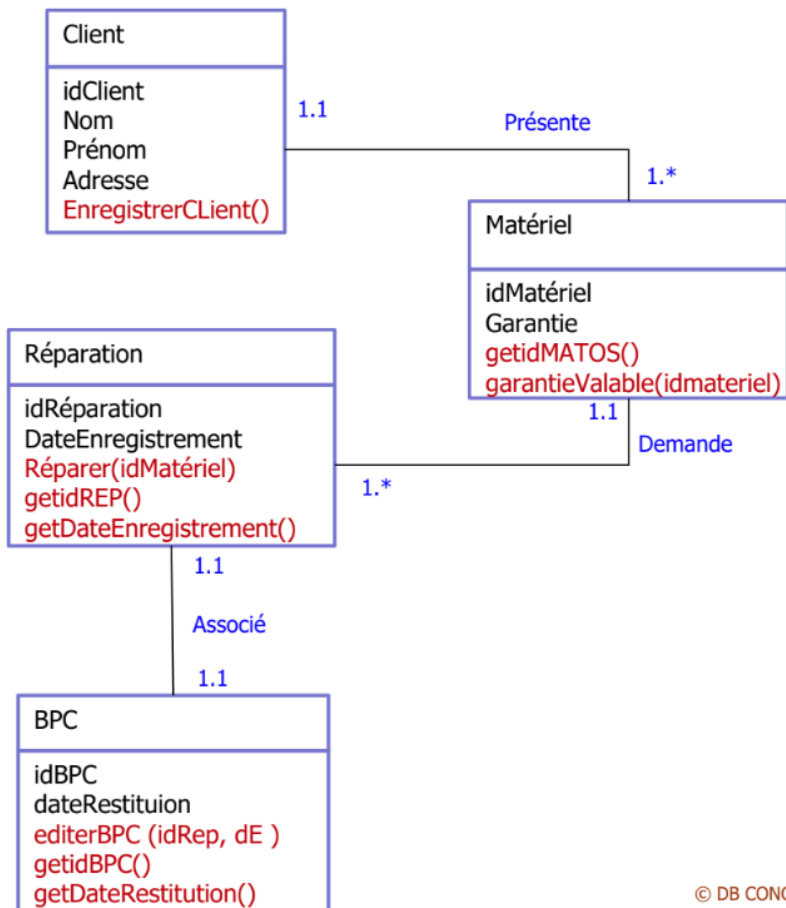
2.2.2 (Conception) de la solution

2.2.2.1 Diagramme de séquence



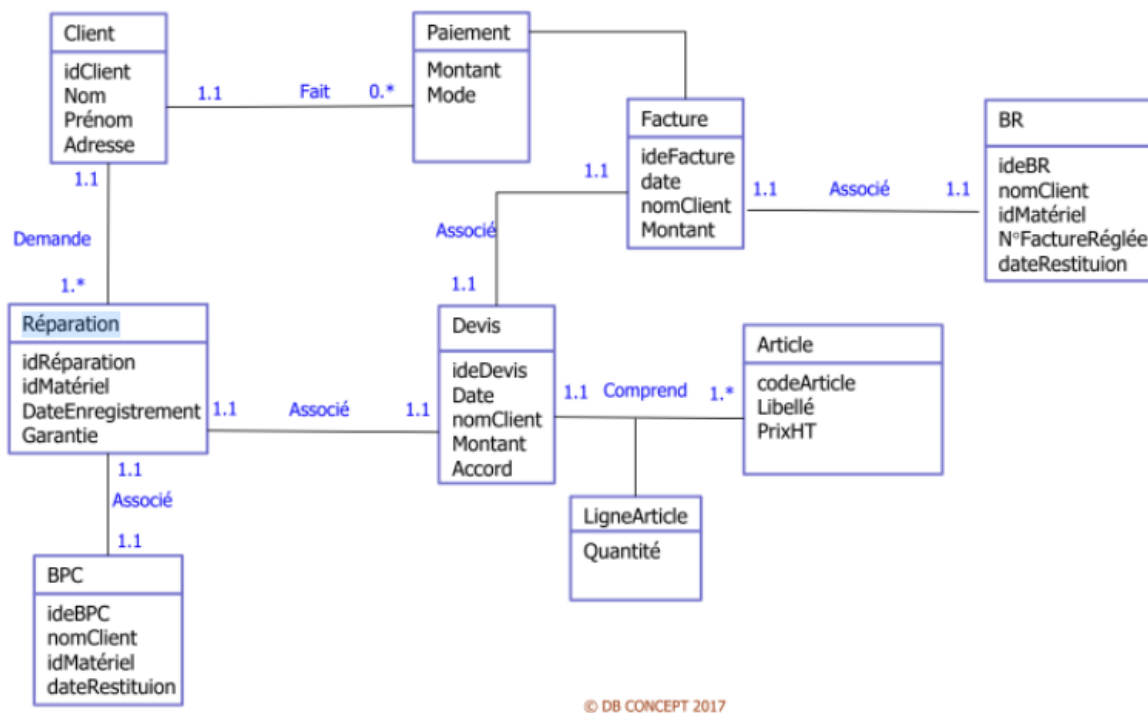
~~2.2.2.2 Diagramme de séquence (MVC) // Sans objet~~

2.2.2.3 Diagramme de Classe de conception



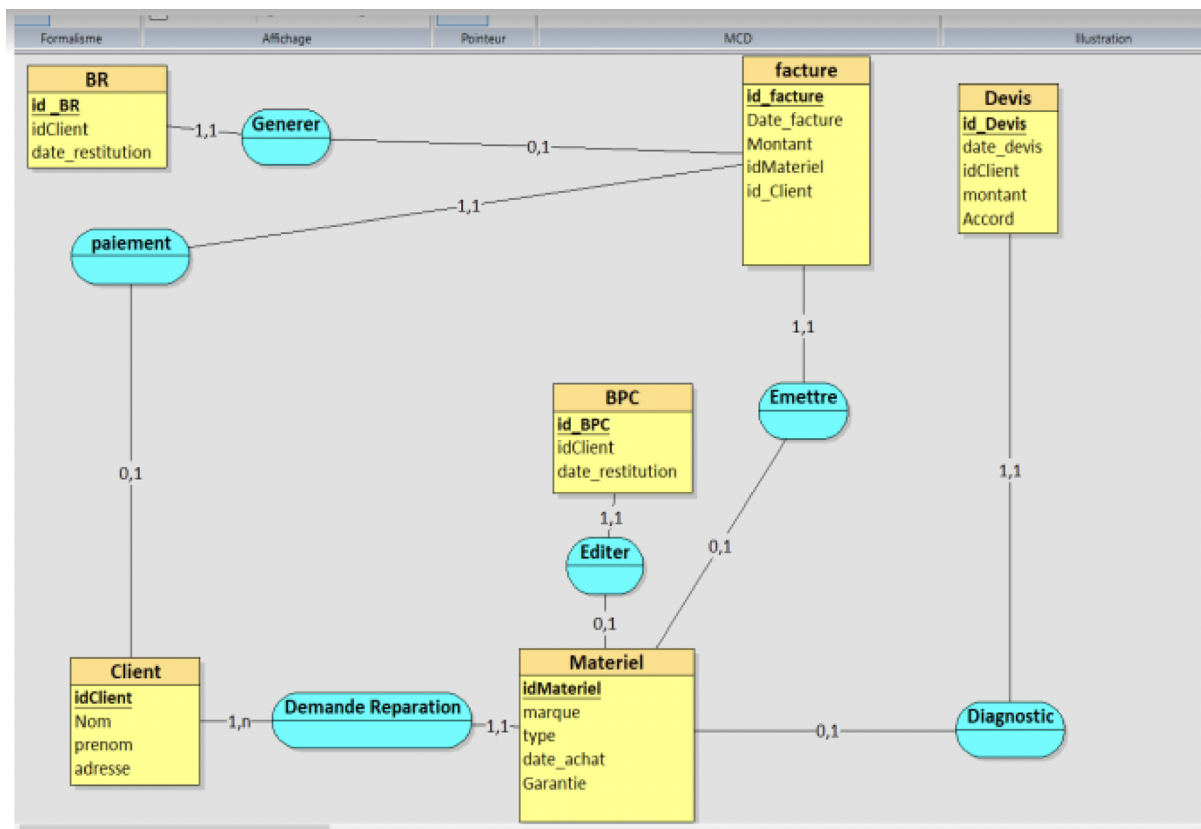
2.3 UC2 (démarche idem à UC1) // Sans objet

2.4 Architecture applicative (Diagramme de classe globale)



2.5 La base de données

2.5.1 Diagramme entité association



2.5.2 Modèle logique de donnée

MLD :

Client = (idClient INT, Nom VARCHAR(50), prenom VARCHAR(50), adresse VARCHAR(50));

Matériel = (idMatériel INT, marque VARCHAR(50), type VARCHAR(50), date_achat DATE, Garantie LOGICAL, #idClient);

Devis = (id_Devis INT, date_devis DATE, idClient INT, montant DECIMAL(7,2), Accord LOGICAL, #idMatériel);

facture = (id_facture INT, Date_facture DATE, Montant DECIMAL(7,2), idMatériel INT, id_Client INT, #idClient, #idMatériel_1);

BPC = (id_BPC INT, idClient INT, date_restitution DATE, #idMatériel);

BR = (id__BR VARCHAR(50), idClient INT, date_restitution DATE, #id_facture);

2.6 Réalisation

2.6.1 Itération 1

- Créer la BDD avec des données représentatives et de quoi réinitialiser la base

Requêtes SQL :

```

CREATE TABLE Client(
idClient INT,
Nom VARCHAR(50),

```



```

prenom VARCHAR(50),
adresse VARCHAR(50),
PRIMARY KEY(idClient)
);
CREATE TABLE Materiel(
idMateriel INT,
marque VARCHAR(50),
type VARCHAR(50),
date_achat DATE,
Garantie LOGICAL,
idClient INT NOT NULL,
PRIMARY KEY(idMateriel),
FOREIGN KEY(idClient) REFERENCES Client(idClient)
);
CREATE TABLE Devis(
id_Devis INT,
date_devis DATE,
idClient INT,
montant DECIMAL(7,2),
Accord LOGICAL,
idMateriel INT NOT NULL,
PRIMARY KEY(id_Devis),
UNIQUE(idMateriel),
FOREIGN KEY(idMateriel) REFERENCES Materiel(idMateriel)
);
CREATE TABLE facture(
id_facture INT,
Date_facture DATE,
Montant DECIMAL(7,2),
idMateriel INT NOT NULL,
id_Client INT,
idClient INT NOT NULL,
idMateriel_1 INT NOT NULL,
PRIMARY KEY(id_facture),
UNIQUE(idClient),
UNIQUE(idMateriel_1),
FOREIGN KEY(idClient) REFERENCES Client(idClient),
FOREIGN KEY(idMateriel_1) REFERENCES Materiel(idMateriel)
);
CREATE TABLE BPC(
id_BPC INT,
idClient INT,
date_restitution DATE,
idMateriel INT NOT NULL,
PRIMARY KEY(id_BPC),
UNIQUE(idMateriel),
FOREIGN KEY(idMateriel) REFERENCES Materiel(idMateriel)
);

```

```
CREATE TABLE BR(
id__BR VARCHAR(50),
idClient INT,
date_restitution DATE,
id_facture INT NOT NULL,
PRIMARY KEY(id__BR),
UNIQUE(id_facture),
FOREIGN KEY(id_facture) REFERENCES facture(id_facture)
);
```

--Requete d'insertion des données entrées via le formulaires
INSERT INTO `client`(`idClient`, `Nom`, `prenom`, `adresse`) VALUES ();--//Les 4 valeurs
nominatives du groupe IHM//;

--Affichage du BPC post generation
SELECT `id_BPC`, `idClient`, `date_restitution`, `idMateriel` FROM `bpc` WHERE 1;

- Coder un UC « enregistrer matériel » ;-)

```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

- Réaliser l'IHM associée à l'UC

Form Preview

Preview Look and Feel

Accueil

N° Facture PC ☐ Garantie N° BPC

1ere Information Client pour diagnostic

Nom Client

Prenom Client

Adresse Code postal Ville

Devis N° ☐ Accord Client

Facture N° ☐ Payée N° BR

Date de restitution F(réparation terminée)

N° Bon Recherché

->Accueil

~~2.6.2 Itération 2 // Sans objet~~

~~• Coder autre UC « »~~

~~• Réaliser l'IHM associée à l'UC~~

2.7 Description des tests

Il vous faut réutiliser les diagrammes classe, de séquence, de séquence système

2.7.1 Itération 1

2.7.1.1 Bâtir la matrice de traçabilité

2.7.1.2 Test unitaire

Sur une classe

2.7.1.3 Test d'intégration

Un diagramme de classe de votre choix

2.7.1.4 Test de validation fonctionnelle

Les diagrammes de séquences réalisant tous les scénarii d'un UC

~~3 Annexe // Sans objet~~

~~3.1 Manuel d'installation // Sans objet~~

~~3.2 Manuel d'utilisation // Sans objet~~

~~3.3 Thèmes de veille // Sans objet~~

~~3.3.1 L'intégration continue d'une application // Sans objet~~

~~3.3.2 Liste et comparatif des outils de déploiement // Sans objet~~