

Projet L3AX1 - Facturation

Conception détaillée

Atia Salah Eddine Bouldja Lounis Namolaru Leonard Saghroun Amos

7 mars 2021

Conception détaillée - projet L3AX1

Les informations d'identification du document

Référence du document : D7

Version du document : 1.01

Date du document : 07/03/21

Auteurs : Atia Salah Eddine Bouldja Lounis
Namolaru Leonard Saghroun Amos

Les éléments de vérification du document

Soumis le : 07/03/21

Type de diffusion : Document électronique (.pdf)

Confidentialité : Réservé aux étudiants
UFR Maths-Info de l'université de Paris

Les éléments d'authentification

Maître d'ouvrage : M. Alaa Dandan

Chef de projet : Leonard Namolaru

Sommaire

Sommaire	2
1. Introduction	4
1.1. Objectifs et méthodes	4
1.2. Documents de référence	4
1.3. Guide de lecture	5
1.3.1 Maîtrise d'œuvre	5
1.3.2 Maîtrise d'ouvrage	5
2.Architecture logiciel	6
3.Technologies et outil	7
1. Back end	7
2. Front end	8
4.Description des modules	9
5.Arborescence de l'application	11
6.Diagramme de cas d'utilisation	12
7. Diagramme de classes	13
8.Conception de la base de données	14
9. Diagramme : cycle de vie	15
10.Organisation	17
11.Planification	19
11.1 Planning des projets	19
11.2 Diagramme de Gantt de la planification	20

12. Glossaire	21
13. Sources	23
14. Index	24

1. Introduction

La conception détaillée est un document qui aide à mieux comprendre les aspects techniques du projet ainsi que de décrire l'architecture de l'application d'une façon assez complète. C'est-à-dire de définir les différentes fonctions, classes, librairies, packages dont on a besoin qui permettront aux développeurs de traduire ce document ainsi que le cahier de charge et recette en langage de programmation.

Ce document est divisé en 5 grandes parties générales : la planification et l'organisation des tâches, présentation des généralités de la méthode de développement utilisée avec la décomposition du développement en phases successives. les méthodes et les outils de développement partie utilisateur et administrateur, les critères déterminant la qualité du logiciel.

1.1. Objectifs et méthodes

le projet est une application web de facturation, le site web permettra d'identifier un utilisateur, ajouter un client, modifier une partie des informations de client, relancer une facture, télécharger une facture, ajouter un contrat, ajouter un produit, éditer des avoirs, associer des contrat, associer des produits, editer une facture, modifier une facture avant sa validation, relancer un client, voir l'état d'une facture..., notre site web doit également respecter des normes juridiques comme par exemple le numéro de facture interchangeable, etc...

1.2. Documents de référence

Le cahier des charges et le cahier de recette ainsi que les sources que vous trouverez dans la partie source sont les documents qui ont servi à l'élaboration du présent document.

1.3. Guide de lecture

Pour chaque type de lecteur, comment utiliser efficacement le document

1.3.1 Maîtrise d'œuvre

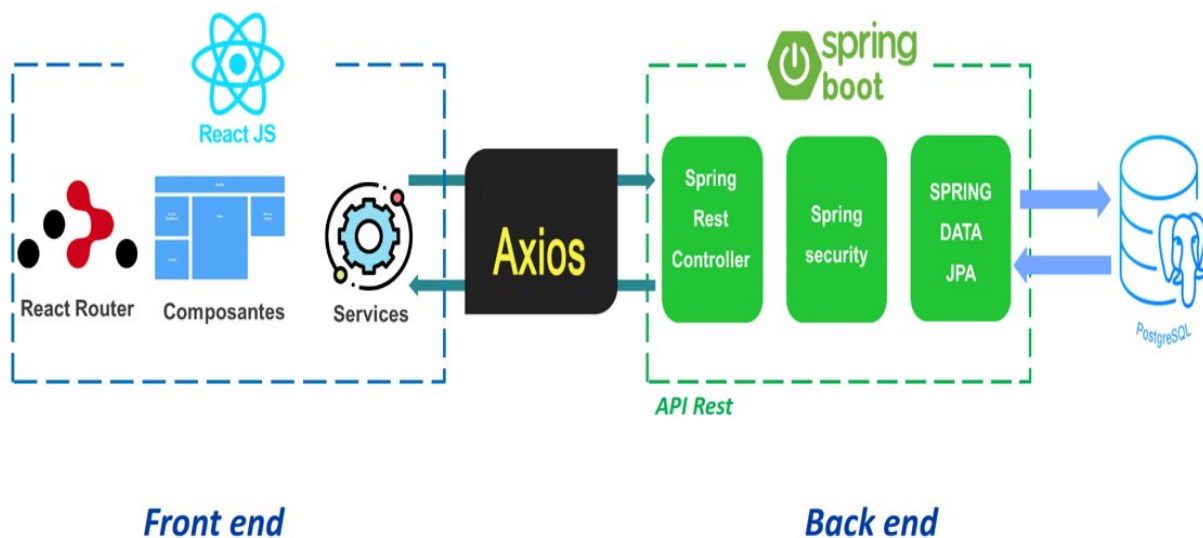
Pour utiliser efficacement ce document, il est conseillé de lire le document dans sa globalité afin de comprendre l'ensemble des éléments abordés, les parties concernant le produit du projet nécessitent une attention du personnel administratif et les parties concernant la description des modules de la part du personnel technique.

Les responsables de la maîtrise d'œuvre se compose de quatre développeurs :
Atia Salah Eddine ; Bouldja Lounis ; Namolaru Leonard ; Saghroun Amos.

1.3.2 Maîtrise d'ouvrage

Pour utiliser efficacement ce document, il est conseillé de lire le document dans sa globalité afin de comprendre l'ensemble des éléments abordés, c'est-à-dire de vérifier que la demande du client (le projet) a bien été comprise d'une façon claire par tous les développeurs travaillant sur ce projet. Grâce à ce document, le maître d'œuvre pourra visualiser les différentes fonctionnalités de l'application et les moyens adoptés par les développeurs. Ici, la maîtrise d'ouvrage se compose de M. Alaa Dandan.

2. Architecture logiciel



Spring boot permet de créer une API qui devra exposer des endpoint correspondant aux actions du **CRUD** (Create, Read, Update, Delete), et communiquer avec la base de données pour récupérer ou modifier les informations (des factures, contrats ...). À noter que l'Api sera de type **REST**.

Pour la bonne séparation entre les données, la présentation et les traitements, Spring boot utilise le design pattern le plus connu qui est **MVC** :

Modèle : implémentation des objets métiers qui seront manipulés par les autres couches

Vue : correspond à l'interface graphique avec l'utilisateur interagit. Son rôle est d'afficher et de présenter les données renvoyées par les modèles. Dans notre cas la vue ce fera avec **reactJs**.

Contrôler : son rôle est de récupérer les requêtes, et de la transmettre au modèle, puis rediriger vers la vue.

3. Technologies et outil

1. Back end

- ❖ **Spring Boot** : un framework open source pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests. Dans notre cas, Spring boot permet de créer **l'API Rest** et permet aussi de mettre en œuvre les autres composants de Spring avec facilité, notamment grâce aux **starters de dépendances** et à **l'autoconfiguration**.

Voici les dépendances que on va utiliser dans notre projet Spring boot :

- Spring web : permet de faire de RESTful, ce qui correspond à notre api
 - LOMBOK : une librairie pour optimiser certaines classes avec une annotation.
 - Spring Data JPA : permet de gérer la persistance des données avec la base de données à l'aide de **Spring Data et Hibernate**.
 - PostgreSQL : pour connecter API à la base de données postgresql.
-
- ❖ PostgreSQL : un système de gestion de base de données relationnelle et objet.
-
- ❖ **Outils** :
 - @WebMvcTest, @AutoConfigureMockMvc : pour tester un projet Spring.
 - PgAdmin 4 : un outil d'administration graphique pour postgresql.
 - Postman : interface graphique
 - IDE, Tomcat, Maven, SVN ...

2. Front end

- ❖ **React** : grâce à react il est facile de créer des interfaces utilisateurs interactives.
définissez des vues simples pour chaque état de l'application.
- ❖ **HTML5** : langage de balisage conçu pour représenter les pages web.
- ❖ **CSS3** : langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML.

- ❖ **Outils** :
 - Npm : gestionnaire de paquets officiel de Node.js permet de créer un projet react et installer des Api et des librairies.
 - Axios : une bib JS fonctionne comme un client HTTP. permet de communiquer avec api en utilisant des requêtes.
 - React router : navigation entre les composantes react.
 - jest react : pour tester l'application react
 - Ide, React dev tools ...

4. Description des modules

1. Page de connexion (ID et MDP)

1. Page Réinitialisation du Mot De Passe
2. Page Mot de Passe modifié avec succès

2. Page des Factures

1. Toutes les factures
2. Bouton “Créer Une Facture”
 - a. Ajout du client sur la facture courante
 - b. Ajout/suppression d’articles
 - c. Bouton “envoyer la facture”
 - d. Page “facture envoyé avec succès”

3. Page des clients (avoir un visu et détail sur tous les clients Ajout/Suppression)

1. Fenêtre Ajout de Client
2. Page Détails et Historique Client
3. Fenêtre Modifier Client (Raison Sociale, adresse, n° de TVA)

4. Page des articles (avoir un visu et détail sur tous les articles)

1. Fenêtre Ajout Article (Produit, Service)

5. Page de Paiement

1. Onglet Paiements réussis
2. Onglet Paiements remboursés

-
3. Onglet Paiements non capturés

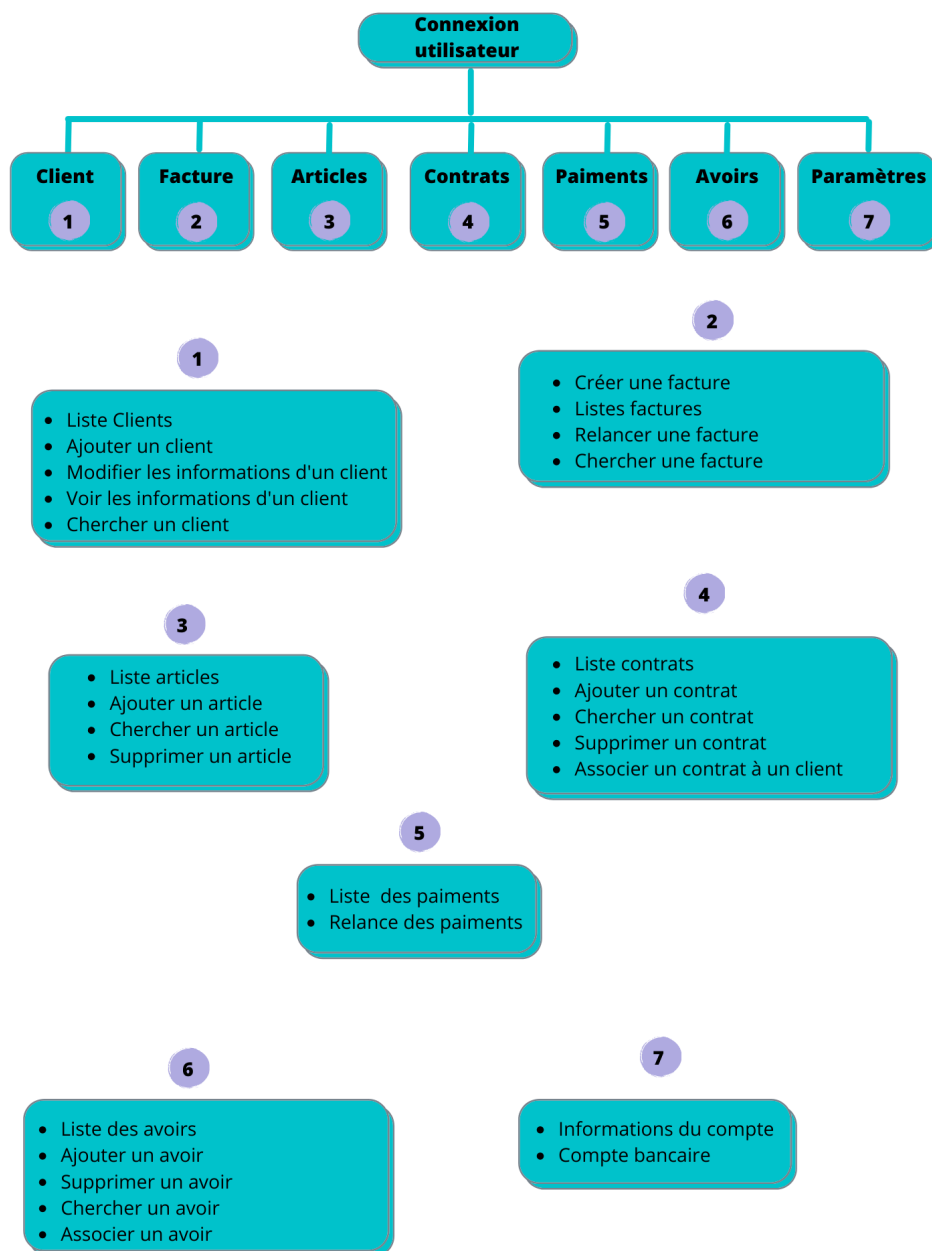
7. Page Paramètre

1. Déroulant Info du Compte et Compte Bancaire
2. Déroulant Taux de TVA

8. Page Déconnexion

1. Bouton “Se déconnecter”

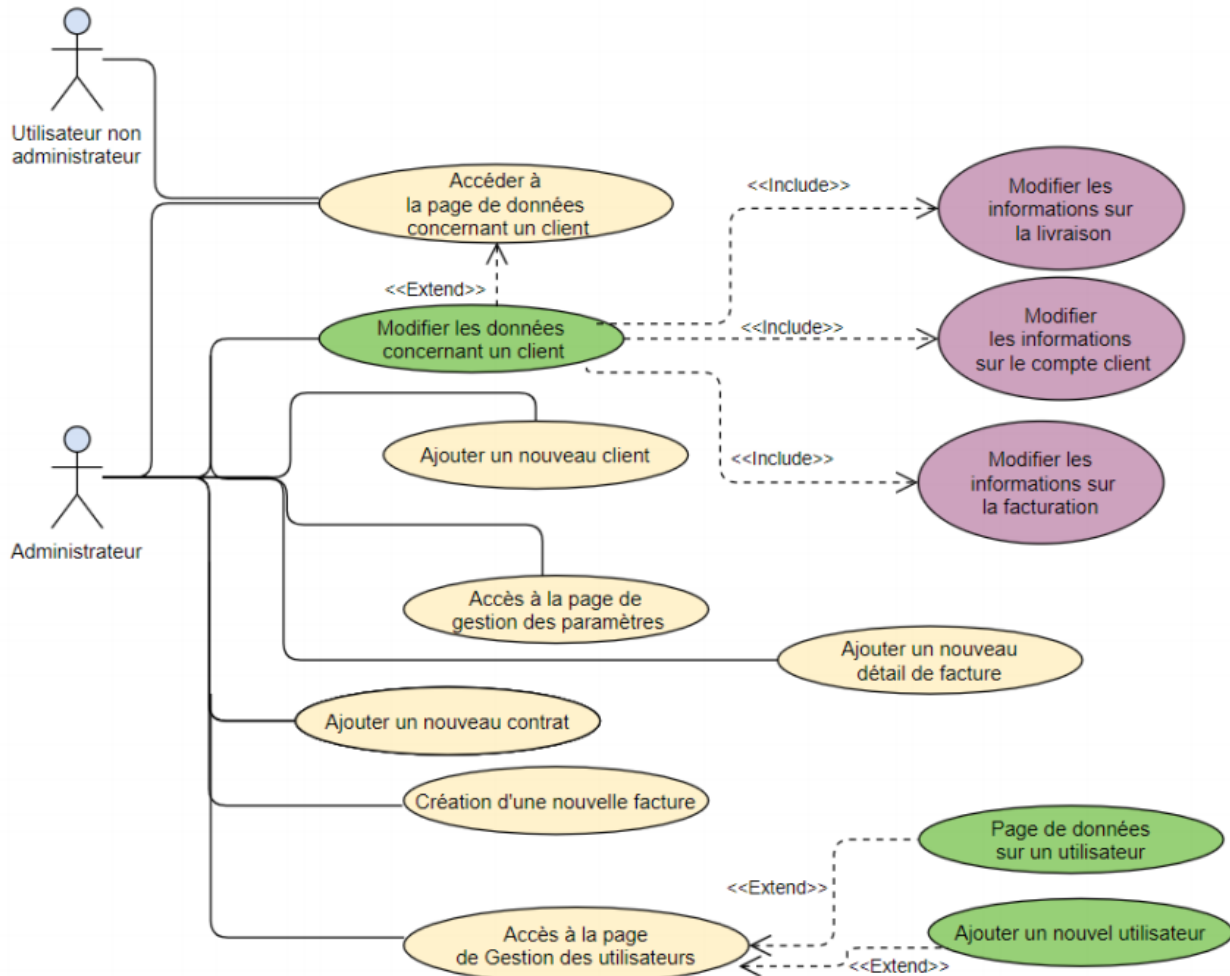
5.Arborescence de l'application



6. Diagramme de cas d'utilisation

Projet L3AX1 – Facturation

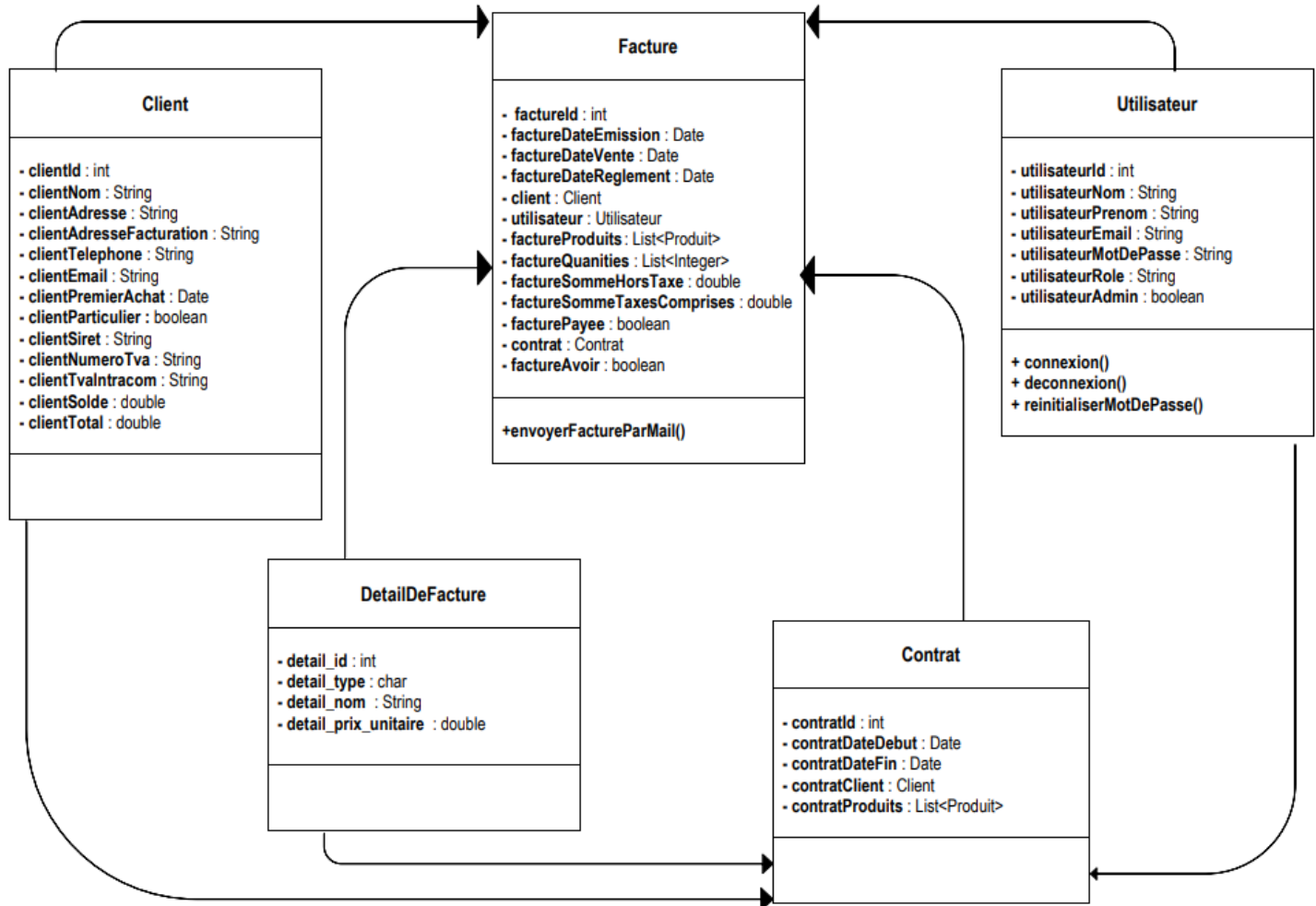
Diagramme de cas d'utilisation



7. Diagramme de classes

Projet L3AX1 – Facturation

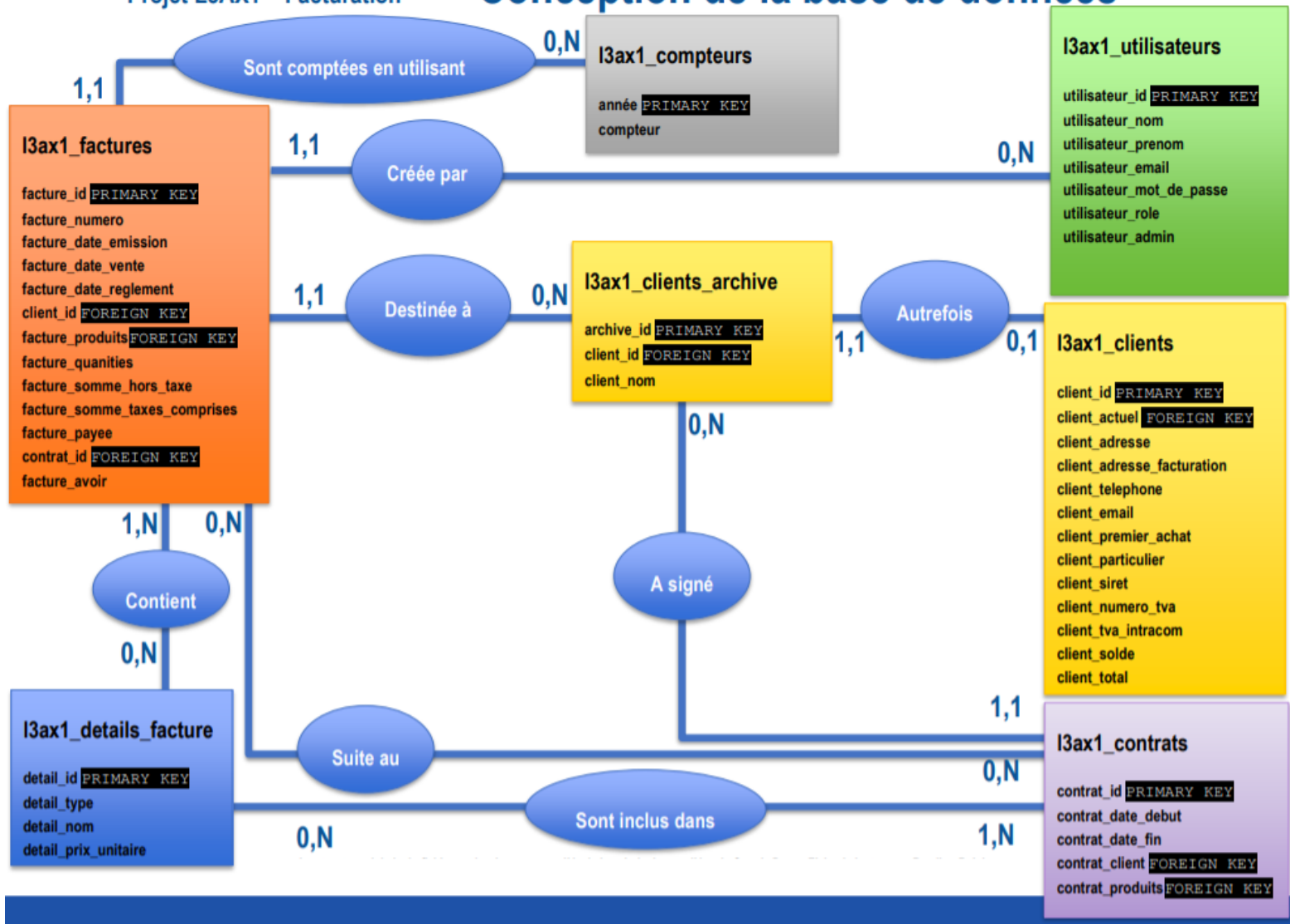
Diagramme de classes



8. Conception de la base de données

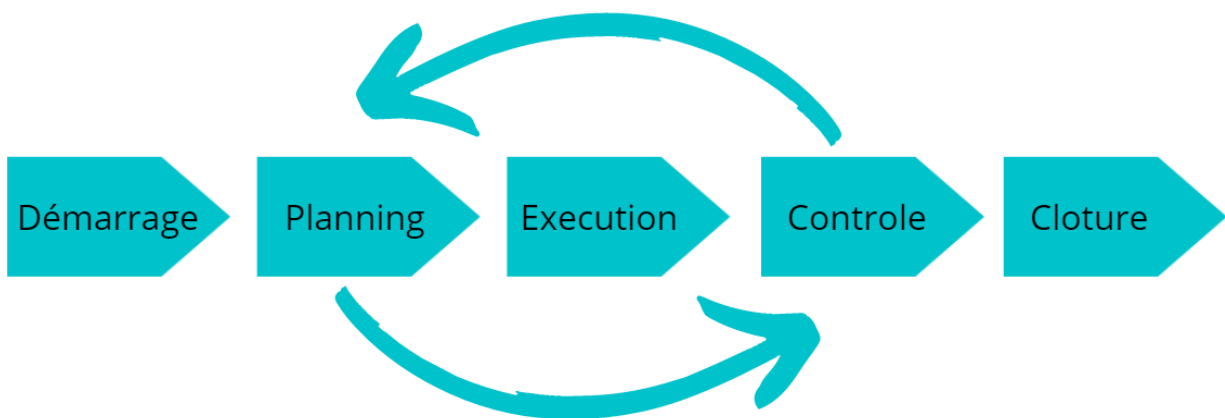
Projet L3AX1 – Facturation

Conception de la base de données



9. Diagramme : cycle de vie

Le cycle de vie d'un projet est un ensemble des étapes de la vie du projet. Il y a plusieurs façons de faire un cycle de vie, nous allons nous baser sur le schéma général suivant :



Démarrage : définir la phase et valider son début.

planning : précisions des objectifs et définir les actions et les moyens à mettre en œuvre pour atteindre ce dernier.

Exécution : accomplir le travail défini précédemment.

contrôle : pour suivre, vérifier et régulariser l'avancement ainsi que le respect des contraintes et des objectifs mais aussi pour identifier et implémenter les modifications nécessaires à l'atteinte des objectifs du projet.

clôture : finaliser les activités et clore le projet ou la phase.

Comme la planification initiale n'est jamais parfaite les activités du contrôle détecte des déviations qui est nécessaire de les corriger, il faut donc passer par la planification l'exécution et de contrôle.

défini par un début et une fin. Il facilite la préparation initiale du projet en définissant par défaut les phases de son déroulement. Il matérialise l'élaboration progressive du produit ou service au cours du déroulement du projet.

qui est applicable notamment dans les projets de développement de système d'information et dans les projets de réorganisation, dès qu'il y a volonté de s'assurer de la pertinence de la solution qui va être généralisée :

- Un premier contenu partiel est réalisé puis testé sur un site pilote.
- Le retour d'expérience du site pilote définit les ajustements à appliquer sur le premier contenu réalisé, et permet d'affiner la définition du reste du projet.

10. Organisation

Décomposition en tâches

ID	La tâche
1	Développement FrontEnd des pages principales du site Création de facture ; Page avec la liste des produits ; Affichage des détails d'un client spécifique Page avec la liste des clients
2	Développement FrontEnd + BackEnd : La page de connexion au panneau d'administration
3	Développement BackEnd : Création de facture
4	Développement BackEnd : Page avec la liste des produits
5	Développement BackEnd : Page avec la liste des clients
6	Développement BackEnd : Affichage des détails d'un client spécifique

7

Développement FrontEnd + BackEnd : Page de gestion des paramètres.

8

Développement FrontEnd + BackEnd : Panneau de gestion des contrats d'un client.

9

Développement FrontEnd + BackEnd : Suivi des paiements

10

Développement FrontEnd + BackEnd : Gestion des utilisateurs

Gérer les utilisateurs qui ont accès au panneau d'administration.

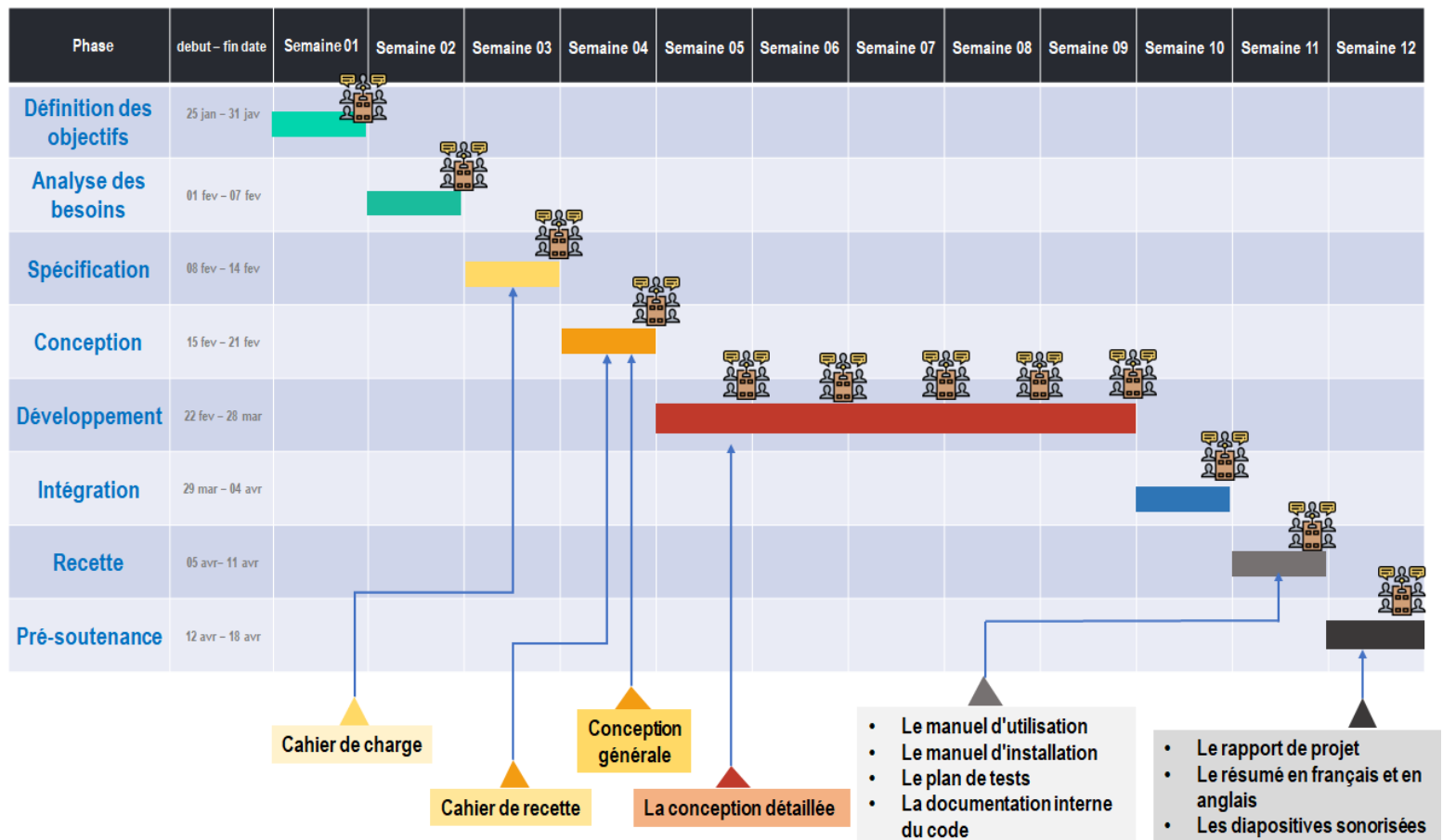
11

Développement FrontEnd + BackEnd : Moteur de recherche

11. Planification

11.1 Planning des projets

Le projet est encadré sur 12 semaines notamment au travers de réunions régulières. Ils s'achèvent par des soutenances et la remise d'un rapport. Les quatre premières semaines sont réservées à l'élaboration du cahier des charges. Les 6 suivantes sont dédiées à la phase de développement et finalement les deux dernières à la rédaction du rapport et à la préparation de la soutenance.



Réunion hebdomadaire.

Planning			
Vacances			
n°5	01/03/21	Développement	Rendu de la conception détaillée
n°6	08/03/21	Développement	
n°7	15/03/21	Développement	
n°8	22/03/21	Développement	
n°9	29/03/21	Développement	
n°10	05/04/21	Intégration	
n°11	12/04/21	Recette	Rendu du projet (documentation + code source + exécutable)
n°12	19/04/21	Pré-soutenance	

- **Date de fermeture de la forge : le 16/05/2021 à 18h**

11.2 Diagramme de Gantt de la planification

	Semaine 5	Semaine 6	Semaine 7	Semaine 8	Semaine 9	Semaine 10	Semaine 11	Semaine 12
Tâche 1								
Tâche 2								
Tâche 3								
Tâche 4								
Tâche 5								
Tâche 6								
Tâche 7								
Tâche 8								
Tâche 9								
Tâche 10								
Tâche 11								

12. Glossaire

❖ **Back end**

C'est l'arrière-plan d'une application. Il représente les parties que ne voit pas les utilisateurs et peut conserver des données saisies dans le Front-End. Il peut aussi traiter, modifier des informations.

❖ **Front end**

Ensemble des éléments visibles et accessibles directement sur un site web (voire sur une application web ou une application web mobile). Il s'oppose par définition au back-end.

❖ **API Rest**

Representational State Transfer Application Program Interface est un style architectural qui permet aux logiciels de communiquer entre eux sur un réseau ou sur un même appareil. Le plus souvent les développeurs utilisent des API REST pour créer des services web. Souvent appelés services web RESTful, REST utilise des méthodes HTTP pour récupérer et publier des données entre un périphérique client et un serveur.

❖ **Framework**

Un framework désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel.

❖ **java EE**

Une spécification pour la plate-forme Java d'Oracle, destinée aux applications d'entreprise.

❖ Spring

Spring est un framework open source pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests. En 2004, Rod Johnson a écrit le livre *Expert One-on-One J2EE Design and Development* qui explique les raisons de la création de Spring.

❖ PostgreSQL

Un système de gestion de base de données relationnelle et objet. C'est un outil libre disponible selon les termes d'une licence de type BSD. Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres, ou propriétaires.

❖ HTML

Un langage de balisage conçu pour faire des pages web.

❖ CSS

Un langage informatique qui permet de faire la présentation de l'HTML. Ce sont des feuilles de style en cascade.

❖ React

Une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état.

13. Sources

- ❖ [Spring | Home](#)
- ❖ [React – Une bibliothèque JavaScript pour créer des interfaces utilisateurs](#)
- ❖ [API REST : Comprendre et construire une API Restful](#)
- ❖ [Développement front-end et back-end : Quelles différences ? | Blog Les](#)
- ❖ [Loi anti-fraude TVA : Point sur l'obligation d'utiliser un logiciel certifié loi anti-fraude TVA](#)
- ❖ <https://www.legalstart.fr/fiches-pratiques/comptabilite-entreprise/logiciel-facturation-obligatoire/>

14. Index

Les mots-clés du document et où les trouver dans celui-ci.

Back End	7,16,17
Conception détaillée	4,5
Diagramme	12,13,14,18
Front End	8,16,17