

Cahier des Charges

Traducteur pseudo code



Caso Alexis - Delome Côme - Kieny Valentin
Lortet Mickaël - Rouquette Loïc

Table des matières

Table des matières	2
Partie I.Introduction	3
Partie II.Analyse des Besoins	4
I.Analyse des utilisateurs	4
II.I.1.Etudiants	4
II.I.2.Personnes souhaitant découvrir l'algorithmique	4
II.I.3.Personnes expérimentées	4
II.Analyse des Tâches	5
II.II.1.Simplifier l'approche de la programmation	5
II.II.2.Aider à la compréhension de l'algorithmique	5
II.II.3.Exécuter directement ses algorithmes sans connaître son implémentation machine	5
II.II.4.Donner un accès aux outils de développement	5
II.II.5.Proposer une gestion plus avancée aux utilisateurs expérimentés	5
Partie III.Spécifications	6
I.Spécifications applicatives	6
III.I.1.Organisation du Site	6
III.I.2.Exécuter et transformer le pseudo-code	7
III.I.3.Ajouter des fonctionnalités	7
III.I.4.Partage	7
III.I.5.Fonctionnalités éventuelles	7
II.Spécifications techniques	7
III.II.1.Le Traducteur	8
III.II.2.Gestion des utilisateurs	8
III.II.3.Le Site Web	8
Partie IV.Réalisation	9
Partie V.Annexes	11
I.Outils	11
II.Langages utilisés	12
III.Modélisation Merise	12

Partie I. Introduction

Nous avons comme projet un site web d'aide à l'apprentissage de l'algorithmique et de la programmation.

Le site s'adressera en priorité aux étudiants de l'IUT en première année qui souhaitent découvrir l'algorithmique avec une approche dynamique. C'est avant tout un site pédagogique qui permettra de voir ce que le code écrit ou donné en cours donne dans d'autres langages ou tout simplement de l'exécuter pour mieux le comprendre.

En effet, la fonctionnalité principale du site sera l'exécution et la traduction de pseudo-code. Cet outil de traduction permettra aux utilisateurs de visualiser directement la fonctionnalité de leurs algorithmes. De plus il permettra de visualiser les équivalences entre les langages.

Une fonctionnalité d'aide et de partage sera également disponible afin d'aider les utilisateurs à progresser. Nous offrirons également une fonctionnalité d'édition de structures pour les utilisateurs expérimentés voulant un traducteur plus poussé.

Tous les outils que nous utiliserons seront expliqués en annexe pour une meilleure compréhension de nos choix entre tel ou tel outil.

Partie II. Analyse des Besoins

I. Analyse des utilisateurs

II.I.1. Etudiants

Profil	Niveau d'algorithmique : Débutant à Moyen
Besoins	Comprendre les fonctionnements de l'algorithmique Avoir un apport dynamique contrairement à l'algorithmique papier Demander de l'aide à la communauté Avoir un complément aux cours de l'IUT qui permet de faire un lien entre les différentes matières de programmation Obtenir des outils de programmation

II.I.2. Personnes souhaitant découvrir l'algorithmique

Profil	Niveau d'algorithmique : Débutant à Moyen
Besoins	Nécessite une structure de code plus proche du langage que les structures de programmation usuelles Aspect dynamique Exécution d'algorithmes sans connaissance de langage de programmation Avoir des cours permettant l'apprentissage des bases de l'algorithmique

II.I.3. Personnes expérimentées

Profil	Niveau d'algorithmique : Moyen à Expérimenté
Besoins	Peut utiliser l'éditeur de pseudo-code pour la mise en page Créer et utiliser des structures de traduction plus avancées Gérer une bibliothèque de fonction Aider les utilisateurs novices Partager ses fonctions sur la plateforme du site

Cette partie est très importante pour la suite du projet, car le site est avant tout pour les utilisateurs et non pour les programmeurs. Ainsi en créant des profils types nous pouvons définir les besoins de chacun et ainsi implémenter de nouvelles fonctionnalités pour que les utilisateurs se sentent à l'aise et est envie de continuer sur notre site. Il faut aussi cerner leurs désirs pour que le site leur servent et ne soit pas qu'un outil inutile.

II. Analyse des Tâches

II.II.1. Simplifier l'approche de la programmation

Le langage machine est complexe pour un débutant et principalement basé sur l'anglais. Le pseudo-code devra proposer une approche plus proche de la langue naturelle ce qui permettra à l'utilisateur de se concentrer uniquement sur la partie algorithmique. L'aspect communautaire pourra donner envie et encourager l'exploration de l'algorithmique par l'utilisateur.

II.II.2. Aider à la compréhension de l'algorithmique

En plus de la facilité de compréhension qu'apporte le traducteur de pseudo-code, une partie apprentissage fournira des cours de base pour la création des algorithmes les plus simples. Il sera possible également pour l'utilisateur de demander de l'aide sur des codes qu'il a écrit. L'utilisateur aura également accès à une bibliothèque de fonctions écrites en pseudo-code disposant de spécifications ce qui lui permettra de comprendre leur fonctionnement et de les utiliser.

De plus l'outil, via sa traduction en plusieurs langages, avec une colorisation des éléments équivalents permettra à l'utilisateur de visualiser les différences et les points communs entre les différents langages.

II.II.3. Exécuter directement ses algorithmes sans connaître son implémentation machine

En plus de la traduction du pseudo-code en différents langages machines, l'utilisateur devra pouvoir exécuter directement son algorithme en pseudo-code dans son navigateur sans avoir besoin d'autres logiciels.

II.II.4. Donner un accès aux outils de développement

Cet espace donnera accès aux utilisateurs à des outils de développement (libres) pour les différentes plateformes ceci afin d'encourager les utilisateurs à développer leurs propres algorithmes sur machine et à pousser plus loin leur technicité.

II.II.5. Proposer une gestion plus avancée aux utilisateurs expérimentés

Les utilisateurs expérimentés pourront éditer des structures de traduction de pseudo-code en langage de programmation. Pour le moment, les structures de traduction seront uniquement accessibles à leur propriétaire, ceci dans le but de garder la stabilité du traducteur, nous nous réservons le droit de changer ce point en cas d'ajout de structures stables et utiles pour les utilisateurs.

Partie III. Spécifications

I. Spécifications applicatives

III.I.1. Organisation du Site

Accueil	<p>La page d'accueil explique le concept du site et affichera les mises à jour des fonctionnalités que nous apporterons.</p> <p>Cette page affichera également les fonctions partagées par les autres utilisateurs ainsi que leurs demandes d'aide.</p>
Tutoriel	<p>Le tutoriel a pour but d'expliquer l'utilisation du traducteur de pseudo-code ainsi qu'une grande partie du site. Une sous-partie permettra aux utilisateurs expérimentés de connaître le fonctionnement de l'éditeur de structures.</p>
Progresser	<p>La partie apprentissage sera principalement composée de cours, d'explications et des exercices d'algorithmique et de programmation. Il expliquera les bases de l'algorithmique et les différences d'approche entre les langages.</p>
Outils de Développement	<p>Cette partie donnera les liens vers les sites et applications pour développer de manière autonome.</p>
Gérer mes outils	<p>Dans cette partie, l'utilisateur pourra gérer sa bibliothèque de fonction et créer de nouvelles structures de traduction. Elle contiendra également les pseudo-codes enregistrés par l'utilisateur. Une gestion de version permettra à l'utilisateur de récupérer jusqu'à la version n-3 de ses pseudo-codes.</p> <p>Cette partie contiendra également, les commentaires et aides apportés par l'ensemble des autres utilisateurs.</p>
Coder	<p>Cette partie sera l'éditeur de pseudo-code qui permettra de traduire le code en langage de programmation et/ou de l'exécuter directement.</p>

III.1.2. Exécuter et transformer le pseudo-code

Le traducteur doit permettre de traduire un algorithme écrit en pseudo-code en différents langages. Il devra aussi permettre d'enregistrer, charger et exécuter les pseudo-codes.

III.1.2.a. Colorisation syntaxique

Une colorisation syntaxique sera disponible sur les langages traduits ce qui permettra une lecture plus claire pour l'utilisateur. Le système de colorisation sera sensiblement la même que pour les principaux IDE afin que l'utilisateur garde ses repères.

III.1.2.b. Colorisation comparative

La colorisation comparative permettra, lors du survol d'un bloc d'instructions par la souris, de visualiser la partie correspondante dans le langage traduit.

III.1.3. Ajouter des fonctionnalités

Via son compte, l'utilisateur pourra ajouter/modifier des structures de traduction pour un ou plusieurs langages. De plus il pourra ajouter des fonctions/procédures en pseudo-code afin de les utiliser dans d'autres pseudo-codes ultérieurs.

III.1.4. Partage

L'utilisateur aura la possibilité de partager ses fonctions et procédures sur une bibliothèque commune. Les administrateurs devront vérifier l'efficacité et la fonctionnalité de celles-ci avant sa publication. De plus il sera possible aux utilisateurs de demander de l'aide à la communauté, le code ainsi enregistré sera posté sur la page d'accueil et les utilisateurs pourront commenter le code pour apporter des modifications et des conseils.

Cependant, dans un souci de stabilité, il n'est pas souhaitable d'ouvrir pour le moment le partage de structure de traduction, ce qui pourrait complexifier la relecture et la correction du système de traduction.

III.1.5. Fonctionnalités éventuelles

III.1.5.a. Calcul de la complexité

Il devra être possible de demander le calcul de la complexité de l'algorithme, pour le cas le moins lourd et le cas le plus lourd. Ceci afin de permettre d'évaluer la performance de son algorithme.

III.1.5.b. Représentation Schématique de l'appel des fonctions/procédures

Cet outil devra permettre de représenter l'appel des fonctions et procédures sous la forme d'un diagramme. Cet outil est avant tout pédagogique et permet de mieux expliquer le fonctionnement des programmes aux nouveaux utilisateurs.

II. Spécifications techniques

III.II.1. Le Traducteur

Le traducteur de pseudo-code en langage machine sera écrit en PHP. Il utilisera plusieurs modules qui lui permettront de transcrire le code utilisateur en langage avec des optimisations.

Outils Utilisés

IDE Netbeans, Langage PHP, WAMP Server

III.II.1.a. Outils de sauvegarde et d'analyse de version

L'outil de sauvegarde et d'analyse de version servira pour l'optimisateur afin qu'il n'ait pas à retraduire le même code. De plus il offrira à l'utilisateur le droit d'exporter ou d'importer son projet en format texte sur son ordinateur (ou directement de le sauvegarder sur le site via un nom d'utilisateur et un mot de passe). L'analyse de version permettra de vérifier les changements entre deux documents.

III.II.1.b. Le débogueur

Le débogage s'effectuera à la lecture étant donné que les structures mal saisies ne seront pas traduites correctement en langage programmation.

III.II.1.c. Optimisation

L'optimisation repose sur l'outil d'analyse de version puisqu'elle consiste à ne retraduire que les éléments nouveaux dans un même pseudo-code. De plus un seul langage ne sera traduisible à la fois afin de ne pas utiliser le serveur trop longtemps.

III.II.2. Gestion des utilisateurs

Outils Utilisés

IDE Netbeans, PhpMyAdmin, WAMP Server

III.II.2.a. Gestion des inscriptions

L'inscription se fera de manière autonome par l'utilisateur. Cette inscription permettra de sauvegarder les informations de l'utilisateur (code, fonction, structure de traduction) ainsi que d'avoir accès aux aides des autres utilisateurs.

III.II.2.b. Gestion des bibliothèques

La gestion de la bibliothèque se fera de manière automatique lors de l'enregistrement. Une table dans la base de données permettra de créer virtuellement des bibliothèques individuelles pour chaque utilisateur.

III.II.2.c. Le système d'aide

Lors de la saisie d'un code, un utilisateur pourra effectuer une demande d'aide ce qui affichera son code sur la page d'accueil. Les autres utilisateurs pourront alors commenter son code sous forme de bulles qui se chargeront à l'ouverture du code par l'auteur. L'utilisateur pourra également aller dans la partie « mes outils » pour voir le nombre de commentaire qu'il a pour ses sujets.

Outils Utilisés

Langage JavaScript, il permet avec PHP de dynamiser l'affichage des pages (requêtes asynchrones).

III.II.3. Le Site Web

Outils Utilisés

IDE Netbeans, Langage HTML, CSS3, WireFrameSketcher, W3C Markup Validator Service, PHP

III.II.3.a. Interface

L'interface devra être simple et intuitive pour permettre à l'utilisateur une navigation simple et un accès aux outils rapide. Même si l'application sera principalement orientée pour les ordinateurs, il semble important qu'il y ait une adaptation minimale pour les tablettes.

III.II.3.b. La Charte Graphique

La charte graphique devra être simple avec un maximum de 5 couleurs et de 7 éléments principaux par fenêtre ce qui simplifiera l'interface et la rendra plus claire. Un système de barre de tâches pour les éléments courants avec un menu déroulant pour les tâches plus rares devra être utilisé le plus possible (voir la maquette en annexe).

III.II.3.c. La Structure

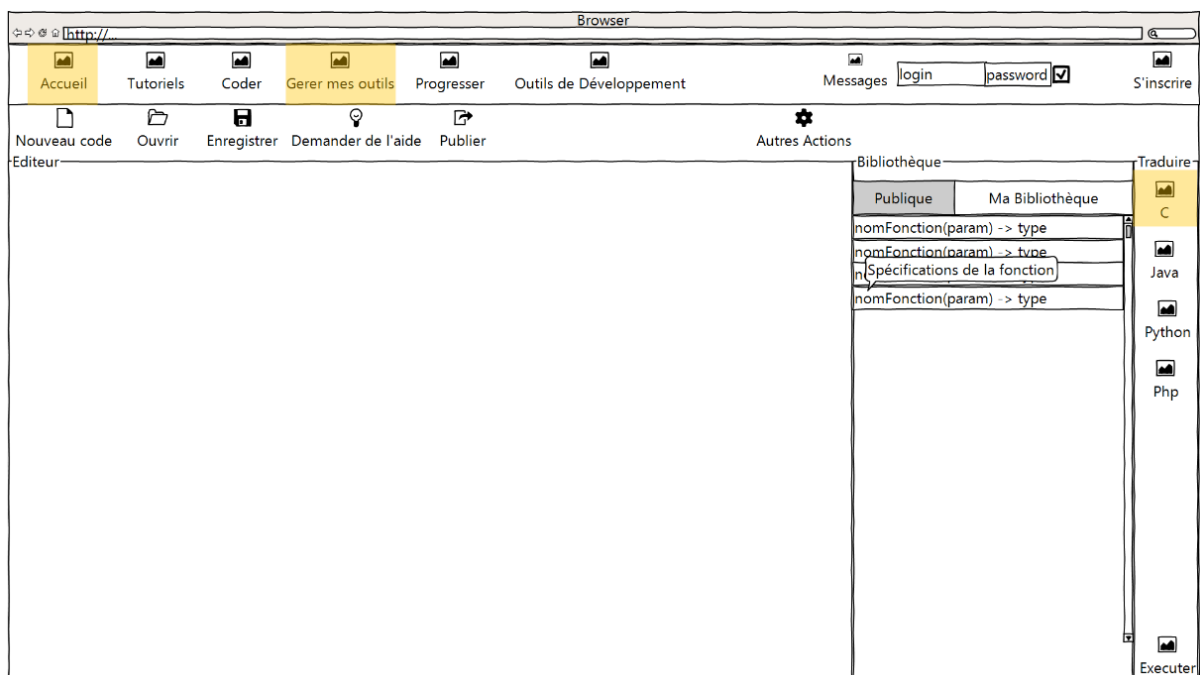
La partie back-end du site est la plus importante car elle permet de faire le lien entre l'interface et le serveur. Elle doit permettre de faire appel à toutes les fonctionnalités implémentées dans le site. On implémentera cette structure grâce au diagramme de merise défini en annexe, cela nous servira pour les fonctionnalités et leur apparition sur le site.

III.II.3.d. Le Contenu

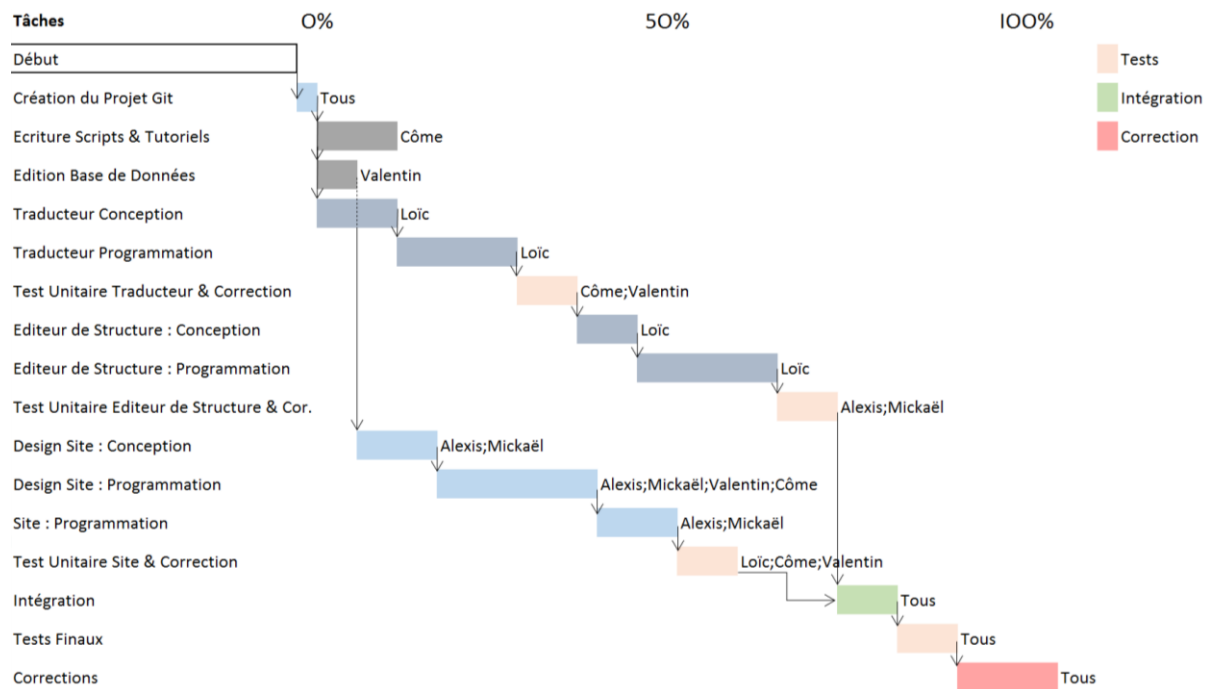
Le site aura un contenu minimum à sa sortie pour permettre aux utilisateurs une navigation plus aisée. Les tutoriels de bases ainsi que les « cours » de pseudo-code seront écrit à la sortie du site. De plus quelques fonctions de bases seront implémentées avec leur documentation.

III.II.3.e. La maquette

La maquette simplifiée du site sera comme ceci : une version dynamique est disponible en ligne à l'adresse suivante : <https://www.dropbox.com/s/v757q8u3to6b752/maquette.pdf?dl=0>



Partie IV. Réalisation



Ci-dessus le diagramme de Gant de notre projet. Toutes les tâches à effectuer sont représentées et réparties entre les différents membres du groupe. Il y aura donc une première phase importante de programmation pour mettre le site et ses éléments en place, puis viendront ensuite une série de tests et de corrections par unité de programmation pour enfin tout intégrer ensemble et refaire une dernière phase de tests et de corrections.

Partie V. Annexes

I. Outils

IDE Netbeans	Nous utiliserons l'IDE Netbeans pour coder le site ainsi que le traducteur. Cet IDE est compatible avec tous les langages nécessaires ce qui permettra de maintenir une homogénéité des documents et de la mise en forme. Il permettra également de passer plus rapidement d'un langage à l'autre lors de l'écriture du traducteur sans avoir à utiliser un IDE spécifique pour chaque langage.
WAMP	Wamp Server est un assemblage de logiciel comprenant Apache, MySQL et PhpServeur. Habituellement, les sites sont hébergés sur des serveurs à distance, cependant, pour le développement, il est plus rapide d'installer un serveur sur sa machine de travail, afin de ne pas dépendre de la connexion et d'avoir une gestion des fichiers plus facile.
FileZilla	FileZilla est un outil de transfert FTP qui permettra d'envoyer les fichiers du site web au serveur hébergeur.
Git	Git est un logiciel de gestion de versions qui permettra de vérifier l'avancement du projet, de le valider au fur et à mesure. De plus il est très intéressant lors de la mise en commun des modules.
Drives Online	Nous utiliserons deux drives pour améliorer le partage du travail. Dropbox sera utilisé pour les fichiers de code ainsi que les documents graphiques (images, icônes). Pour les documents écrits et pour la gestion du planning, nous utiliserons Google Drive pour simplifier la modification des fichiers.
WireFrame Sketcher	Wireframe Sketcher est un outil de maquettage. Il permet de créer des maquettes fil-de-fer ainsi que de créer des déroulements d'actions.
Windows Project	Le W3C est le consortium de validation des standards du web. Le site W3C validator nous permettra de valider la conformité de la partie HTML de notre site afin qu'il soit correctement visible sur un maximum de navigateurs différents.
W3C Markup Validator Service	Windows Project est similaire à Gantt, il permet de gérer la répartition des tâches et des ressources. De plus il permet de créer un suivi du projet pour vérifier le respect des échéances.
PowerAMC	PowerAMC est un outil de modélisation UML/Merise, il est nécessaire pour aider à concevoir la base de données sur laquelle reposera la gestion du site.

II. Langages utilisés

V.II.1.a. Langages du Web

Le langage PHP est celui utilisé pour communiquer avec le serveur du site web. Le langage MySQL permettra de faire le lien entre le php et la base de données.

Le couple de langage HTML/CSS sert à la mise en page de la page web et de l'application.

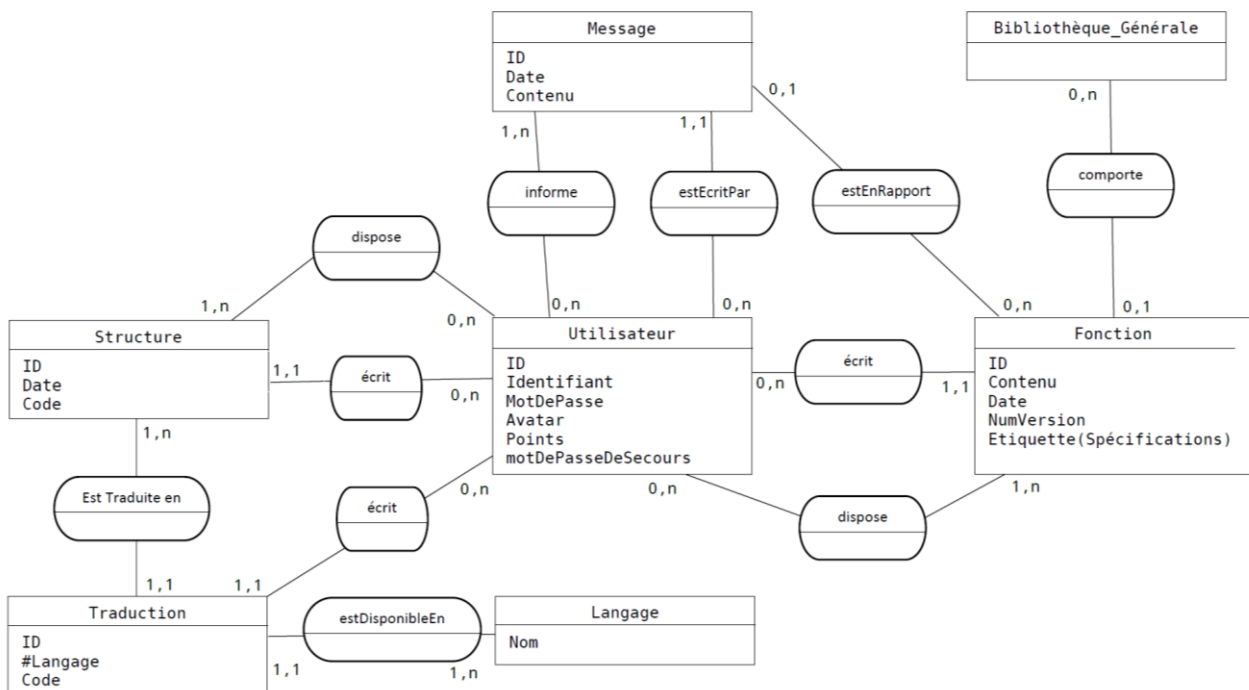
V.II.1.b. JavaScript

Le JavaScript sera utile pour l'exécution du pseudo-code sans transformation visible pour l'utilisateur. Le pseudo-code sera traduit en JavaScript et directement exécuté par le navigateur. Il est également important pour effectuer des requêtes asynchrones avec le serveur et donner une impression dynamique à l'utilisateur.

V.II.1.c. Langages traduits (Python, JavaScript, C, Java et PHP)

Les langages traduits ne sont pas à proprement parler nécessaire à la réalisation du projet, mais permettront d'offrir une traduction du pseudo-code correcte et plus diversifiée.

III. Modélisation Merise



On peut voir ici la représentation de la base de données en diagramme de merise. C'est grâce à cela que nous pourrons programmer la structure qui accueillera l'ensemble des données. Ce premier diagramme est un outil essentiel pour la suite car nous n'avons plus qu'à implémenter en sachant déjà les liaisons entre les tables.