|  |
| --- |
| **API de conversions numériques** |
| https://lien |
|  |
| Ce projet web est dans les grandes lignes celui d’une api qui pourrait faire des conversions entre bases de numériques a l’occurrence 2, 8 et 16, ainsi que des calculs de bases. |
|  |
| **Joël Bertrand Tsafack** |
| **29 septembre 2023** |
|  |



Sommaire

[I. Pré requis et décisions 3](#_Toc147394265)

[1. Spécifications fonctionnelles 3](#_Toc147394266)

[2. Spécifications techniques 3](#_Toc147394267)

[II. Administration 4](#_Toc147394268)

[1. Les conversions 4](#_Toc147394269)

[2. Les calculs 4](#_Toc147394270)

[III. Ressources à disposition et visions futurs 4](#_Toc147394271)

[IV. Autres aspects techniques 5](#_Toc147394272)

[1. Modèle logique de données 5](#_Toc147394273)

[Explications : 5](#_Toc147394274)

# PRÉ REQUIS ET DÉCISIONS

## Spécifications fonctionnelles

* Elle doit être capable de faire les conversions entre la base 2, 8, 10 et 16 pour des nombres entiers
* Elle doit être capable de faire les conversions entre la base 2, 8, 10 et 16 pour des nombres décimaux
* Elle doit être capable de faire les complémentations en 1, 2, 7, 8, 15, 16
* Elle doit être capable d’effectuer des calculs de bases (addition, soustraction, division, multiplication) dans les bases spécifiées
* Les utilisateurs doivent être capables de reporter un bug via la page officielle de la documentation
* Elle doit aussi être capable de

## Spécifications techniques

* Pour utiliser l’API, les utilisateurs devrons posséder une clé, qui seras générée seulement après son inscription
* L’API sera dument documentée, possiblement grâce a git hub page
* Il a été décidé que l’API devra être conçue avec laravel et PHP
* Elle doit être capable de gérer les mauvais paramètres (incohérences de bases, de type de données)
* Elle doit être capable de gérer les mauvaises entrées d’url
* L’API devra garder une trace de chacun de ses appels : pour utiliser l’API, le client devra s’inscrire en spécifiant son pays, un mot de passe et un numéro de récupération sera généré
* Le numéro de récupération est unique, provient d’un chiffre au hasard(entre 0 et 100) haché avec l’id d’utilisateur

# ADMINISTRATION

## Les conversions

L’API doit avoir une url pour les conversions : <https://domaine/conversions/JsonObject>

JsonObject seras un objet json qui contiendras :

* Le nombre a convertir
* La base de départ
* La base d’arrivé
* Le format : (paramètres facultatifs, elle pourrait spécifier si l’on veut séparer chaque nombre par un caractère)

Son type de retour sera toujours un JSON, qui contiendra :

* Le résultat de la conversion
* Le nombre à convertir
* La base de départ
* La base d’arrivé

## Les calculs

L’API aura une url pour faire les calculs : <https://domaine/operations/nomOperation/JsonObject>

Elle contiendra les informations suivantes :

* Le tableau de nombres à calculer
* La base de départ
* La base d’arrivé
* Le résultat du calcul
* Le format d’arrivé
* Type de l’opération

Le résultat contiendra les informations suivantes :

* La base d’arrivé
* Le résultat du calcul

# Ressources à disposition et visions futurs

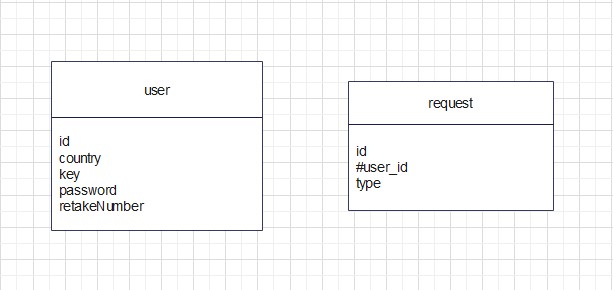
Je dispose d’une moyenne de 3heures par jour à allouer au projet, qui pourrait me prendre une semaine

Dans un futur certainement proche, je me servirais cette api pour monter une application web et mobile autour de conversions

# AUTRES ASPECTS TECHNIQUES

## Modèle logique de données

Logiciel utilisé : edraw max (mauvaise exportation)



*Image :* Modèle logique de données

### Explications :

* Retake\_number est le numéro de récupération, qui sera utilisé avec id du client pour permettre la récupération de la clé et la réinitialisation du mot de passe
* Requests.type est le type de la requête effectuée (s’il s’agit d’une conversion ou d’un calcul) les valeurs possibles sont : conversion, addition, substraction, multiplication, division