|  |  |
| --- | --- |
| PROJET SQL  IONIS STM | sujet  Conception de système de gestion de base de données pour la gestion de trottinettes  membre du groupe   * NKUMBE ENONGENE Astrid Aurelien * SIMA BALLA Joseph Martin Ludovic * TAMO FOSSOUO Wilfred Kevin |
|  |  |

SOMMAIRE

[CADRAGE 2](#_Toc118651849)

[I. ENONCE DU BESOIN INITIALE 2](#_Toc118651850)

[II. CIBLE 2](#_Toc118651851)

[III. MISSION 2](#_Toc118651852)

[IV. PERIMETRE FONCTIONNEL 2](#_Toc118651853)

[V. PERIMETRE TECHNIQUE 3](#_Toc118651854)

[DOSSIER DE CONCEPTION 4](#_Toc118651855)

[I. MODELE CONCEPTUEL DE DONNEES 4](#_Toc118651856)

[II. MODELE LOGIQUE DE DONNEES 4](#_Toc118651857)

[III. MODELE PHYSIQUE DE DONNEES (UML ERD DIAGRAM) 0](#_Toc118651858)

[Cardinalités et règles de gestion 0](#_Toc118651859)

[Dictionnaire des données 0](#_Toc118651860)

[IV. REQUETTES SQL 1](#_Toc118651861)

[Langage de définition des données pour MYSQL 1](#_Toc118651862)

# CADRAGE

## ENONCE DU BESOIN INITIALE

Bolt souhaite renouveler son système de gestion de base de données pour sa gestion de trottinettes. Ce nouveau système devra enregistrer :

* les modèle des trottinettes ainsi que leurs caractéristiques,
* les coordonnées GPS (longitude, latitude) du lieu où il se trouve,
* les utilisateurs (informations personnelles, nom, prénom, mail, etc.),
* un historique des utilisateurs par numéro de série d'une trottinette,
* un historique des locations faites par l'utilisateur (coût).

Un utilisateur peut visualiser sur son application, les trottinettes disponibles. Lorsqu'il souhaite prendre une trottinette, il peut visualiser les caractéristiques de celle-ci. A la fin de l'utilisation, la distance parcourue est enregistrée et le prélèvement se fait automatiquement via ses coordonnées bancaires.

## CIBLE

Toutes les personnes aptes à conduire une trottinette résidant en France.

## MISSION

Notre équipe d’étudiant de IONIS STM,

* NKUMBE ENONGENE Astrid Aurelien
* SIMA BALLA Joseph Martin Ludovic
* TAMO FOSSOUO Wilfred Kevin

avons pour obligation vis-à-vis de Bolt le client:

* Analyser le besoin,
* Concevoir la base de données (MCD, MLD)
* Livrer les instructions de création de DB (DDL)

## PERIMETRE FONCTIONNEL

À la suite des échanges avec le client Bolt, La solution finale retenue aura la responsabilité de proposer les fonctionnalités suivantes :

* Gestion des modèles de trottinettes.
* Gestion des locations de trottinettes
* Gestion des mouvements des trottinettes
* Gestion des paiements de location des trottinettes
* Gestion des utilisateurs des trottinettes

L‘application devra proposer deux consoles de gestion (backend) protèges par une page d’authentification par un identifiant (email) et mot de passe

* Une console d’administration
* Une console des utilisateurs

L’application devra aussi proposer un frontend ouvert au public

* Page d’accueil
* Page de contact

## PERIMETRE TECHNIQUE

La solution proposée sera développée en un seul module tels une application web monolithique non distribué compatible avec toutes les plateformes

|  |
| --- |
| Matériels requis |
| Un PC (Serveur d’administration) de bureau complet avec ses périphériques E/S ayant les caractéristiques minimales suivantes.   * CPU: DUAL CORE * RAM: 4GO * ESPACE DE STOKAGE: 80GO |

|  |
| --- |
| Déploiement |
| * OS : Windows * Logiciels à jour : Firefox / Google Chrome, MySQL 5, XAMPP |

|  |
| --- |
| Hébergement Web ou Cloud Dédié |
| * Cpanel * 1 nom de domaine + sous domaines offerts * Certificat SSL multi sous-domaine * Adresse électronique * MySQL 5 ou 8 + Access a distance * PHP 7 ou 8 * Access SFTP ou FTPS   Recommandation : OVH, LWS, IONOS |

# DOSSIER DE CONCEPTION

## MODELE CONCEPTUEL DE DONNEES

Diagram, schematic

Description générée automatiquement

## MODELE LOGIQUE DE DONNEES

Diagram

Description générée automatiquement

## What is an ER Diagram and How to Implement it? | EdurekaDiagram Description générée automatiquementMODELE PHYSIQUE DE DONNEES (UML ERD DIAGRAM)

### Cardinalités et règles de gestion

* Un modèle de trottinette peut instancier plusieurs trottinettes.
* Une trottinette peut faire l’objet de plusieurs locations ou aucune
* Une location a obligatoirement une facture et une seule et ne peut pas être supprimé si la facture est payer (quand date de paiement de paiement est renseigné).
* Un utilisateur enregistré doit avoir au moins une location de trottinette sinon il n’en vaut pas la peine de l’enregistrer, donc ne pourra pas être supprimer
* Un utilisateur peut louer un ou plusieurs trottinettes mais pas en même temps.
* La suppression d’un modèle de trottinette n’entraine pas la suppression en cascade des instances de trottinettes identifiées par le numéro de série.

### Dictionnaire des données

## REQUETTES SQL

|  |
| --- |
| -- Q1 : Afficher  les caracteristiques de toutes les trotinnettes  SELECT \*  FROM trottinettes T LEFT JOIN trottinettes\_models TM ON TM.id = T.trottinettes\_models\_id;  SELECT details\_complementaires  AS 'caracteristiques complementaires' FROM trottinettes;  -- Q2 : Afficher les utilisateurs qui ont un prenom commencant  par A  SELECT \* FROM utilisateurs WHERE prenom LIKE 'A%';  -- Q3 : Afficher les utilisateurs qui ont un nom et  prenom commencant par A et ordonne par ordre aplha de  nom de famille  SELECT \* FROM utilisateurs WHERE  nom LIKE 'A%' AND prenom LIKE 'A%' ORDER BY nom ASC ;  -- Q4 : Afficher le numéro de série de chaque trottinette ordonné par ordre croissant de sa position en longitude  SELECT serial\_number FROM trottinettes ORDER BY longitude ASC ;  -- Q5 : Afficher le nom et prénom de tous les utilisateurs qui ont pris une trottinette durant le mois de mai par ordre décroissant de prix, afficher l'historique des courses  SELECT  U.nom ,  U.prenom ,  L.trottinettes\_serial\_number,  L.date\_enreg ,  F.cout  FROM locations L      LEFT JOIN utilisateurs U ON L.utilisateurs\_id = U.id      LEFT JOIN factures F ON F.locations\_id = L.id  WHERE MONTH(L.date\_enreg) = 5  AND YEAR(L.date\_enreg) = YEAR(CURDATE())  ORDER BY F.cout DESC; |

### Langage de définition des données pour MYSQL

<https://raw.githubusercontent.com/nkaurelien/ionis-stm/main/Sql/ionis-stm_mysql_ddl_gestion_trottinettes.txt.sql>

|  |
| --- |
| CREATE TABLE utilisateurs (  id bigint(19) NOT NULL AUTO\_INCREMENT,  uuid char(36),  id\_document\_type varchar(255) NOT NULL,  id\_document\_number varchar(255) NOT NULL,  nom varchar(100) NOT NULL,  prenom varchar(100) NOT NULL,  email varchar(100) NOT NULL UNIQUE,  telephone varchar(20) NOT NULL UNIQUE,  date\_enreg timestamp NOT NULL,  PRIMARY KEY (id)) CHARACTER SET UTF8;  CREATE TABLE trottinettes (  serial\_number varchar(100) NOT NULL,  ean varchar(255) NOT NULL,  sku varchar(255) NOT NULL,  couleur varchar(255),  version varchar(100),  latitude double NOT NULL,  longitude double NOT NULL,  details\_complementaires varchar(2555),  date\_enreg timestamp NULL,  trottinettes\_models\_id smallint(5) NOT NULL,  PRIMARY KEY (serial\_number),  UNIQUE INDEX (ean),  UNIQUE INDEX (sku)) CHARACTER SET UTF8;  CREATE TABLE trottinettes\_models (  id smallint(5) NOT NULL AUTO\_INCREMENT,  model varchar(100) NOT NULL,  marque varchar(100) NOT NULL,  coloris varchar(1000),  dimensions varchar(100),  poids double,  pliable bit(1),  position\_moteur double,  puissance\_moteur int(10),  vitesse\_max int(10),  longueur double,  largeur double,  charge\_max double,  hauteur double,  diametre\_max\_roue double,  autonomie\_max int(10),  PRIMARY KEY (id),  UNIQUE INDEX (model)) CHARACTER SET UTF8;  CREATE TABLE locations (  id bigint(19) NOT NULL AUTO\_INCREMENT,  ref varchar(10) NOT NULL,  date\_enreg timestamp NOT NULL,  date\_retour\_prevu timestamp NULL,  date\_retour\_effectif timestamp NULL,  trottinettes\_serial\_number varchar(100) NOT NULL,  utilisateurs\_id bigint(19) NOT NULL,  PRIMARY KEY (id),  UNIQUE INDEX (ref)) CHARACTER SET UTF8;  CREATE TABLE trottinettes\_custom\_attributes (  id int(10) NOT NULL AUTO\_INCREMENT,  valeur varchar(100) NOT NULL,  value\_type varchar(100) NOT NULL,  custom\_attributes\_id smallint(5) NOT NULL,  trottinettes\_serial\_number varchar(100) NOT NULL,  PRIMARY KEY (id)) CHARACTER SET UTF8;  CREATE TABLE custom\_attributes (  id smallint(5) NOT NULL AUTO\_INCREMENT,  code varchar(100) NOT NULL,  libelle varchar(255) NOT NULL,  value\_type varchar(100),  values\_enums varchar(255),  PRIMARY KEY (id),  UNIQUE INDEX (code)) CHARACTER SET UTF8;  CREATE TABLE locations\_trackings (  id bigint(19) NOT NULL AUTO\_INCREMENT,  latitude double NOT NULL,  longitude double NOT NULL,  geo\_point varchar(2555),  time timestamp NOT NULL,  trottinettes\_serial\_number varchar(100) NOT NULL,  utilisateurs\_id bigint(19) NOT NULL,  PRIMARY KEY (id)) CHARACTER SET UTF8;  CREATE TABLE factures (  id bigint(19) NOT NULL AUTO\_INCREMENT,  ref varchar(10) NOT NULL,  cout decimal(10, 0) NOT NULL,  mode\_paiement varchar(255),  date\_facturation timestamp NOT NULL,  date\_paiement timestamp NULL,  locations\_id bigint(19) NOT NULL,  PRIMARY KEY (id),  UNIQUE INDEX (ref)) CHARACTER SET UTF8;  ALTER TABLE locations ADD CONSTRAINT FKlocations793757 FOREIGN KEY (utilisateurs\_id) REFERENCES utilisateurs (id);  ALTER TABLE locations ADD CONSTRAINT FKlocations884825 FOREIGN KEY (trottinettes\_serial\_number) REFERENCES trottinettes (serial\_number);  ALTER TABLE trottinettes ADD CONSTRAINT FKtrottinett637072 FOREIGN KEY (trottinettes\_models\_id) REFERENCES trottinettes\_models (id);  ALTER TABLE trottinettes\_custom\_attributes ADD CONSTRAINT FKtrottinett291612 FOREIGN KEY (custom\_attributes\_id) REFERENCES custom\_attributes (id);  ALTER TABLE trottinettes\_custom\_attributes ADD CONSTRAINT FKtrottinett94753 FOREIGN KEY (trottinettes\_serial\_number) REFERENCES trottinettes (serial\_number);  ALTER TABLE factures ADD CONSTRAINT FKfactures14491 FOREIGN KEY (locations\_id) REFERENCES locations (id);  ALTER TABLE locations\_trackings ADD CONSTRAINT FKlocations\_102057 FOREIGN KEY (trottinettes\_serial\_number) REFERENCES trottinettes (serial\_number);  ALTER TABLE locations\_trackings ADD CONSTRAINT FKlocations\_395064 FOREIGN KEY (utilisateurs\_id) REFERENCES utilisateurs (id); |