

History

Qui	Quand	Quoi
Paul Claverie	04/10/2023	création

CONTEXTE FONCTIONNEL :

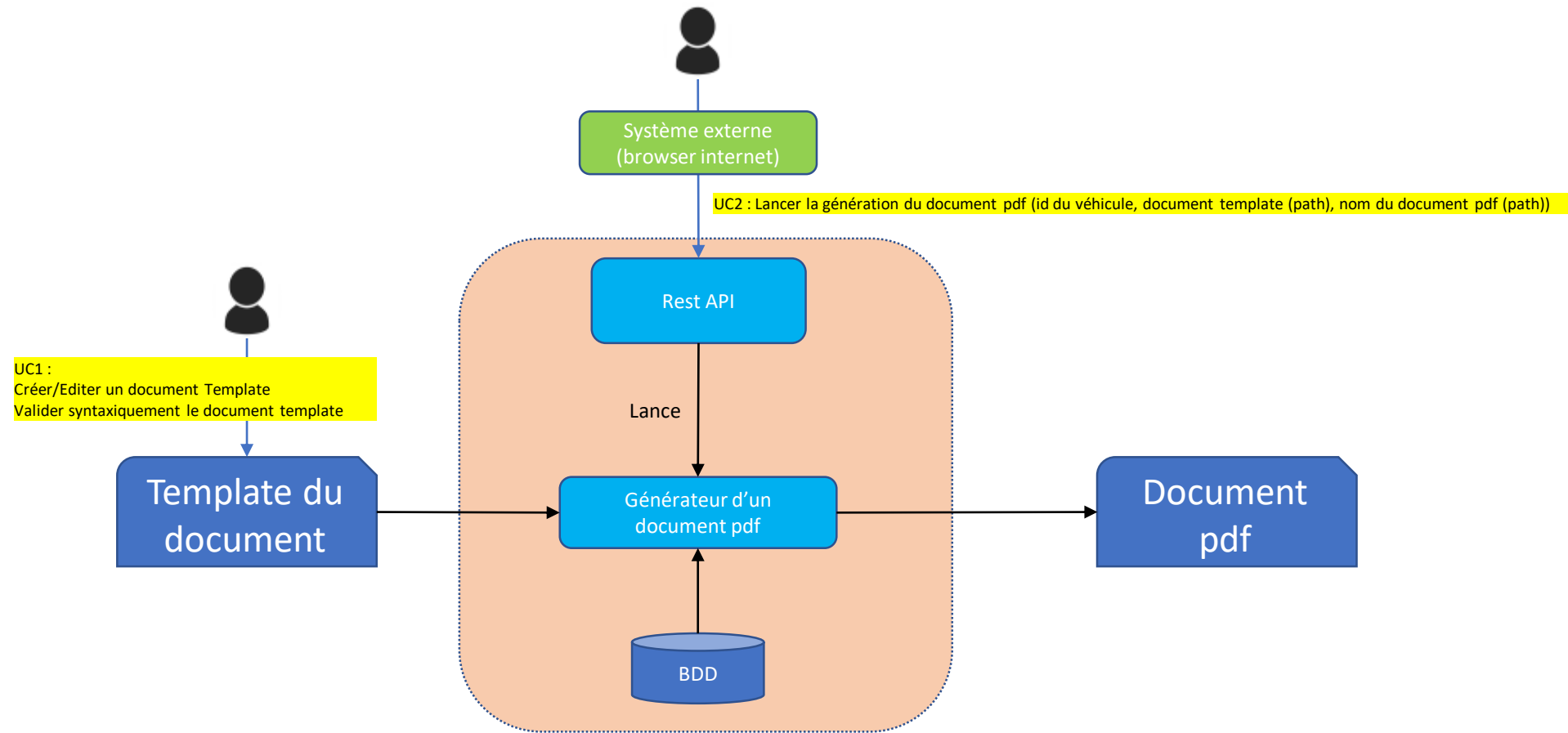
Usine de production de véhicules.

Génération d'un document pdf qui contient la liste des incidents détectés lors de la production d'un véhicule

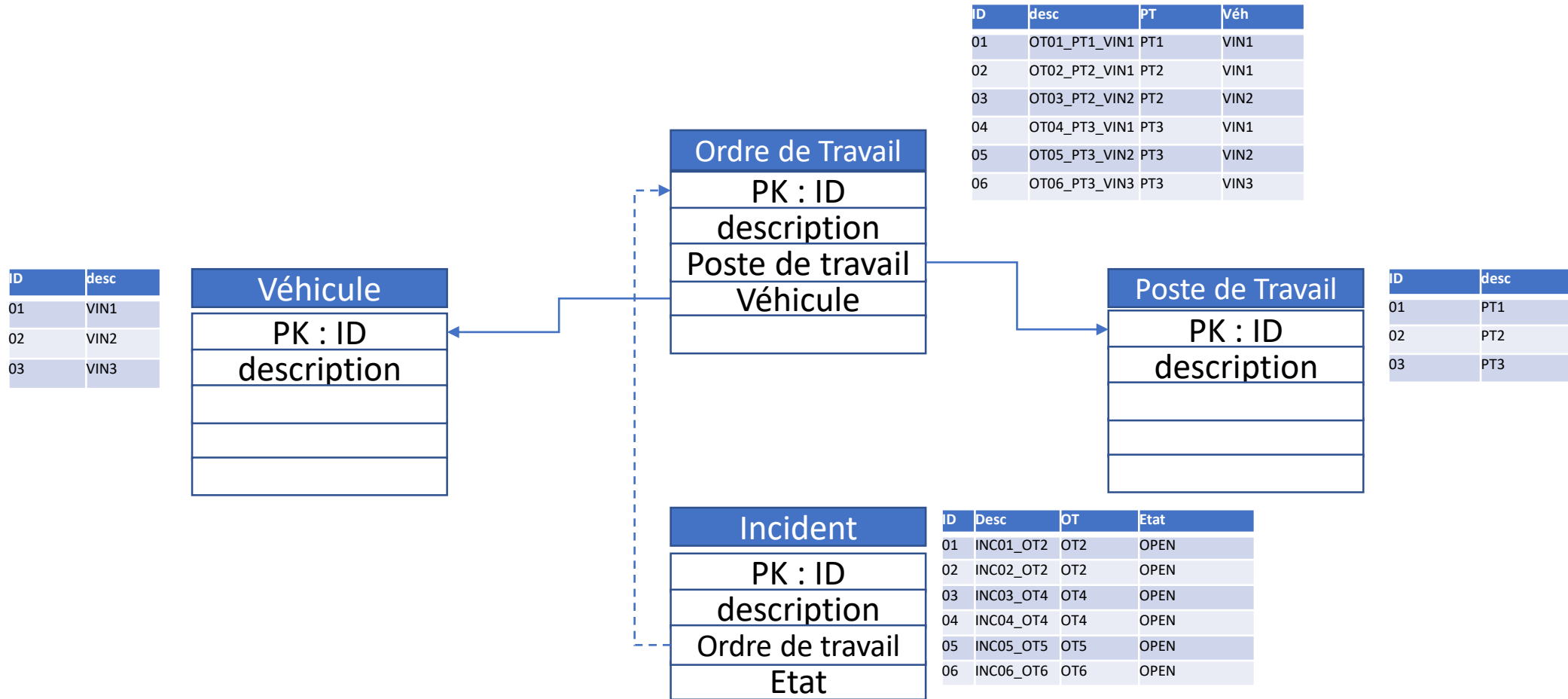
OBJECTIF du mini projet :

Créer une rest API qui génère un document pdf pour un véhicule à partir :

- d'un document template
- De données « véhicule » stockées en base de données SQL



Modèle de données de la base



Pour produire un véhicule, plusieurs ordres de travail sont créés.

Un ordre de travail est réalisé sur un poste de travail (e.g. poste de travail d'usinage, poste de travail d'assemblage)

Un ordre de travail est créé pour un couple (véhicule, poste de travail)

Des incidents peuvent être déclarés (état='OPEN') pendant l'exécution d'un ordre de travail.

UC1 : Créer/Editer un document template

La solution doit permettre de valider syntaxiquement le document template

Le document template doit contenir :

- Des balises de style (formatage du texte (titre, gras, ...) , header, footer, tableau, saut de page, ...).
- Du texte en dur
- Requête SQL qui doit être exécutée par la solution

Livrables attendus :

- Recommander un format de fichier (HTML avec balises CSS, ...)
- Recommander un éditeur de ce format de fichier
- Faire une démonstration des capacités de cet éditeur

UC2 : Lancer la génération du document pdf

Le document template doit contenir :

- Des balises de style (format texte (titre, gras, ...) , header, footer, tableau, ...).
- Du texte en dur
- Requête SQL qui doit être exécuté par la solution



Exemples de requête SQL :

- Requête pour récupérer une valeur (exemples: description du véhicule, nb total incidents sur le véhicule)
- Requête pour remplir un tableau (exemple : liste des incidents détectés sur le véhicule – quelque soit le poste de travail)
- Requête pour remplir un tableau dans un tableau (exemple, faire un **tableau** pour chaque poste de travail qui a travaillé sur le véhicule. Chaque **tableau** donne la liste des incidents de ce poste de travail. Ajouter l'OT pendant lequel l'incident a été déclaré)

Livrables attendus :

- Doc de conception
 - identifier l'OSS (Open Source Software) pour chaque composant du logiciel
 - Signature de la rest API (faire une opération synchrone) – code erreur compris
- Organiser l'équipe de dev en fonction des composants logiciels de la solution
- Construire la BDD et la populer avec des données inventées
- Faire une démonstration des capacités de la solution avec un document template dont la spécification des requêtes SQL est donnée ci-dessus
- Simuler un cas d'erreur dans une requête SQL présente dans le document template
- Code sous Git

Si le temps le permet, vous pouvez faire les améliorations suivantes :

UC 3 : la solution notifie la génération du document de manière asynchrone (faire un serveur http pour recevoir la notification de la fin de génération du document pdf)

**UC 4 : Dans un tableau présent dans le document template, possibilité de rajouter une feuille de style conditionnelle au résultat d'une requête SQL
(Exemple, Si le nombre total d'incidents sur le véhicule = 0, la couleur de fond doit être verte
Dans le cas contraire, la couleur de fond doit être rouge)**

UC 4+ : idem UC4 mais le faire sur la valeur d'une cellule d'un tableau

UC 5 : Possibilité de configurer une table des matières dans le document template

UC 5+ : La table de matières dans le document pdf a des signets qui permettent au lecteur d'accéder à un chapitre depuis la table des matières

```

BEGIN TRANSACTION;
CREATE TABLE IF NOT EXISTS "vehicule" (
    "vehicule_id"    INTEGER NOT NULL,
    "vehicule_desc"  TEXT NOT NULL,
    PRIMARY KEY("vehicule_id" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "poste" (
    "poste_id"    INTEGER NOT NULL,
    "poste_desc"  TEXT NOT NULL,
    PRIMARY KEY("poste_id" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "ordre" (
    "ordre_id"    INTEGER NOT NULL,
    "ordre_desc"  TEXT NOT NULL,
    "poste"       INTEGER NOT NULL,
    "vehicule"    INTEGER NOT NULL,
    PRIMARY KEY("ordre_id" AUTOINCREMENT),
    FOREIGN KEY("vehicule") REFERENCES "vehicule"("vehicule_id"),
    FOREIGN KEY("poste") REFERENCES "poste"("poste_id")
);
CREATE TABLE IF NOT EXISTS "incident" (
    "incident_id"    INTEGER NOT NULL,
    "incident_desc"  TEXT NOT NULL,
    "ordre"          INTEGER NOT NULL,
    "etat"           TEXT NOT NULL,
    PRIMARY KEY("incident_id" AUTOINCREMENT),
    FOREIGN KEY("ordre") REFERENCES "ordre"("ordre_id")
);
INSERT INTO "vehicule" ("vehicule_id","vehicule_desc") VALUES
(0,'vehicule4

');
INSERT INTO "vehicule" ("vehicule_id","vehicule_desc") VALUES
(1,'vehicule1');
INSERT INTO "vehicule" ("vehicule_id","vehicule_desc") VALUES
(2,'vehicule2');
INSERT INTO "vehicule" ("vehicule_id","vehicule_desc") VALUES
(3,'vehicule3');
INSERT INTO "poste" ("poste_id","poste_desc") VALUES (0,'');
INSERT INTO "poste" ("poste_id","poste_desc") VALUES (1,'poste1');
INSERT INTO "poste" ("poste_id","poste_desc") VALUES (2,'poste2');
INSERT INTO "poste" ("poste_id","poste_desc") VALUES (3,'poste3');
INSERT INTO "ordre" ("ordre_id","ordre_desc","poste","vehicule") VALUES
(1,'ordrel_postel_vehicule1',1,1);
INSERT INTO "ordre" ("ordre_id","ordre_desc","poste","vehicule") VALUES
(2,'ordre2_postel_vehicule2',1,2);
INSERT INTO "ordre" ("ordre_id","ordre_desc","poste","vehicule") VALUES
(3,'ordre3_poste2_vehicule1',2,1);
INSERT INTO "ordre" ("ordre_id","ordre_desc","poste","vehicule") VALUES
(4,'ordre4_poste3_vehicule2',3,2);
INSERT INTO "ordre" ("ordre_id","ordre_desc","poste","vehicule") VALUES
(5,'ordre5_poste2_vehicule2',2,2);
INSERT INTO "incident" ("incident_id","incident_desc","ordre","etat")
VALUES (1,'incident1_ordrel',1,'OPEN');

```

```
INSERT INTO "incident" ("incident_id","incident_desc","ordre","etat")
VALUES (2,'incident2_ordre2',2,'OPEN');
INSERT INTO "incident" ("incident_id","incident_desc","ordre","etat")
VALUES (3,'incident3_ordre2',2,'OPEN');
INSERT INTO "incident" ("incident_id","incident_desc","ordre","etat")
VALUES (4,'incident4_ordre4',4,'OPEN');
INSERT INTO "incident" ("incident_id","incident_desc","ordre","etat")
VALUES (5,'incident5_ordre4',4,'OPEN');
COMMIT;
```


Liste des incidents du véhicule <Véhicule.description>

Table des matières

Résumé.....	3
Liste des incidents et status des incidents	4
Liste détaillé des incidents	5
Poste de travail <description du poste de travail>.....	5

Résumé

Ce rapport a pour but d'informer sur la liste des incidents des incidents déclarés sur le véhicule <description du véhicule>.

<nb total d'incidents déclarés sur le véhicule> ont été trouvés sur le véhicule

Liste des incidents

Le tableau ci-dessous montre la liste des incidents pour le véhicule <description du véhicule> :

ID	Description de l'incident	Etat
<id de l'incident>	<description de l'incident>	<etat de l'incident>

Liste détaillé des incidents

Ce chapitre contient la liste des incidents déclarés par poste de travail

Poste de travail <description du poste de travail>

Le tableau ci-dessous montre la liste des incidents déclarés sur le poste de travail <description du poste de travail> pour le véhicule <description du véhicule> :

ID	OT
<id de l'incident>	<ordre de travail>