

<b>BTS SERVICES INFORMATIQUES AUX ORGANISATIONS</b>	<b>SESSION 2025</b>
<b>ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)</b>	
<b>Épreuve E6 - Conception et développement d'applications (option SLAM)</b>	

<b>DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE</b>		<b>N° réalisation : 1</b>
Nom, prénom : Léo MERKEL		N° candidat : 2351134646
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>	Date : 27. / 04. / 2025..
Organisation support de la réalisation professionnelle		
Intitulé de la réalisation professionnelle StockSeeker : application de gestion de stock		
Période de réalisation : 2024 - 2025..... Lieu : dans le cadre de ma formation.....		
Modalité : <input checked="" type="checkbox"/> Seul(e) <input type="checkbox"/> En équipe		
Compétences travaillées <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données		
Conditions de réalisation <sup>5</sup> (ressources fournies, résultats attendus) Ressources fournis : Cahier des charges. Documentation officielle. Raspberry PI utilisé comme serveur Résultats attendus : - Conception et développement d'une API REST permettant la gestion des produits, entrépôts et stocks. - Interface front-end épuré et moderne, connectée à l'API - Mise en place d'un système d'authentification à l'aide de JWT token		
Description des ressources documentaires, matérielles et logicielles utilisées <sup>6</sup> Langages : Python (Back-end), JavaScript, html, css (Front-end) Frameworks : Django, Django REST Framework, Vue.JS, Primevue SGBD : PostgreSQL Serveur web : Gunicorn (Back-end), serve (Front-end) Hébergement en ligne : Raspberry PI avec Nginx Proxy Manager IDE : Visual Studio Code Gestion de version : Github, Gitea Outils : Vite.js, Docker, Postman, Node.js		
Modalités d'accès aux productions <sup>7</sup> et à leur documentation <sup>8</sup> Site disponible sur : stock.orabis.fr Documentation disponible sur : https://orabis.fr/documentation-stock Code source disponible sur mon gitea : https://gitea.orabis.fr/orabis/E6-BStockSeeker   https://gitea.orabis.fr/orabis/E6-FStockSeeker		

<sup>5</sup> En référence aux conditions de réalisation et ressources nécessaires du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

<sup>6</sup> Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

<sup>7</sup> Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

<sup>8</sup> Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

**ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle  
(verso, éventuellement pages suivantes)****Épreuve E6 - Conception et développement d'applications (option SLAM)****Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs**

Dans le cadre de mon BTS SIO option SLAM, j'ai développé en autonomie complète une application web baptisée StockSeeker, répondant aux besoins d'une gestion efficace des produits et entrepôts dans une entreprise fictive. Le projet s'inscrit dans une démarche réaliste, respectant les contraintes techniques, budgétaires et pédagogiques du BTS.

L'objectif principal : concevoir une Single Page Application (SPA) fluide, moderne, intuitive, et facilement utilisable au quotidien, sans compromettre la sécurité ou la maintenabilité.

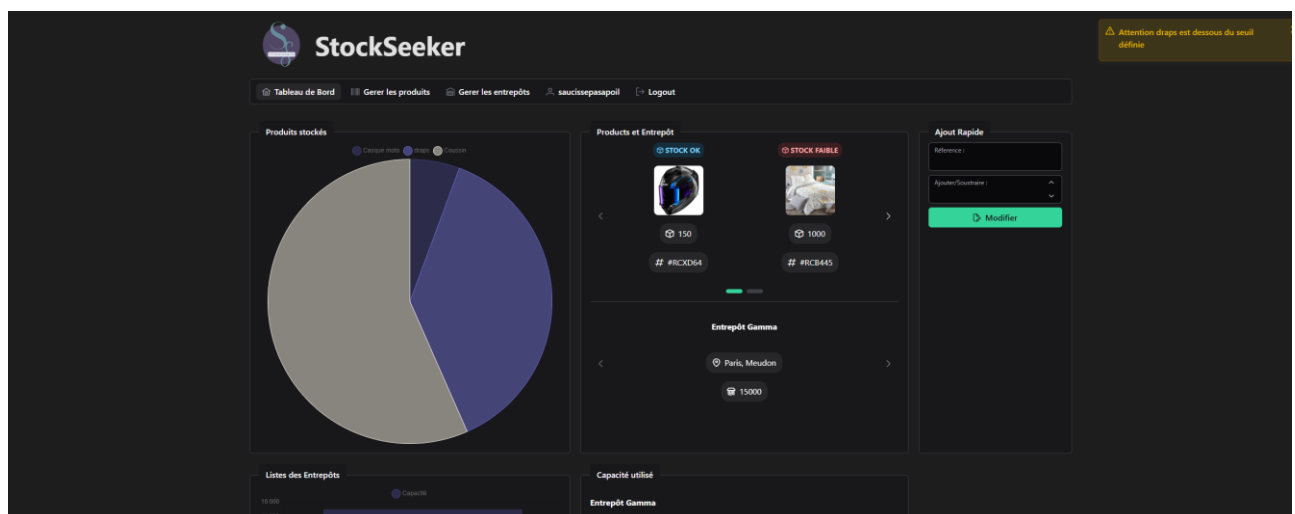
StockSeeker est une application web permettant la gestion centralisée de produits répartis dans plusieurs entrepôts. Développée pour répondre aux besoins d'une entreprise fictive de logistique, elle permet aux utilisateurs authentifiés de créer, consulter, modifier et supprimer des produits et des entrepôts via une interface web ou une API REST sécurisée.

Le projet est construit sur un backend Django REST Framework, utilisant l'authentification JWT pour sécuriser les endpoints. Les données sont stockées dans une base de données relationnelle, et les opérations CRUD sont accessibles via des endpoints RESTful clairement définis.

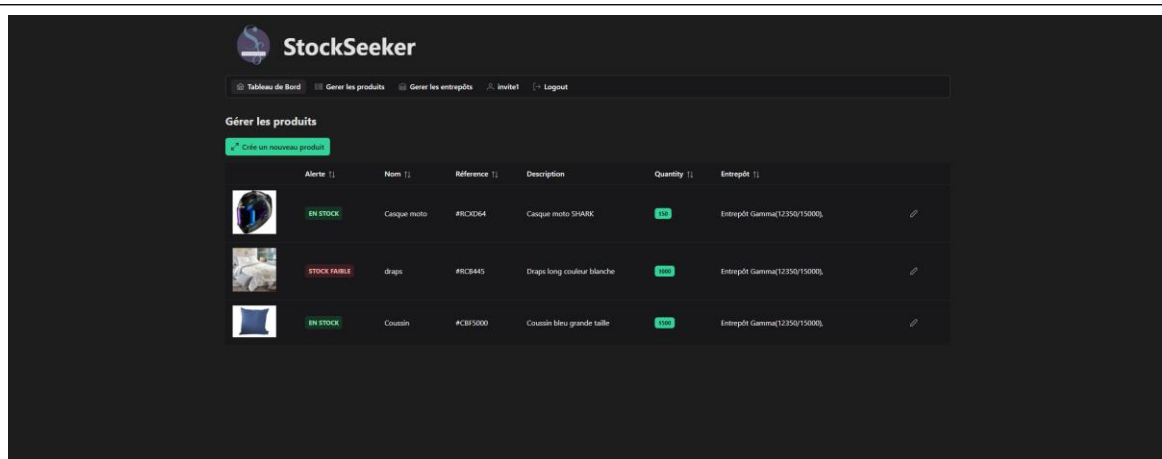
**Fonctionnalités principales**

- Gestion des produits (ajout, consultation, modification, suppression)
- Gestion des entrepôts (ajout, consultation, modification, suppression)
- Association de produits à un ou plusieurs entrepôts
- Page d'accueil graphique permettant de voir directement les ressources
- Création et modification de profil utilisateur via un espace personnel
- Authentification sécurisée via JWT

Il s'agit d'un projet « from scratch ». Que j'ai entièrement réalisé moi-même en utilisant les ressources que j'ai à ma disposition.



Screenshot 1 - Tableau de bord du site StockSeeker



Screenshot 2 - Page d'ajouts de produits

Voici les différentes tâches que j'ai menées à bien :

Conception de l'architecture REST et développement du back-end :

- Création des modèles Django : Product, Warehouse, gestion des utilisateurs.
- Développement de l'API avec Django REST Framework
- CRUD complet
- Authentification JWT
- Transactions atomiques

Développement du front-end avec Vue.js :

- SPA réactive avec Vite.js
- Interface intuitive et responsive
- Utilisation de PrimeVue, Chart.js, Axios, js-cookie
- Gestion des vues :
- Tableau interactif des produits et entrepôts
- Dashboard avec alertes critiques
- Page de login et formulaire d'ajout rapide

Sécurité :

- CORS restreint
- Token JWT sécurisé
- HTTPS et permissions utilisateur

L'API REST permet aux utilisateurs d'interagir avec l'application par des appels HTTP standard (GET, POST, PUT, PATCH, DELETE). Voici un aperçu global :

api		^
GET	/api/products/	🔒 ▼
POST	/api/products/	🔒 ▼
GET	/api/products/{id}/	🔒 ▼
PUT	/api/products/{id}/	🔒 ▼
PATCH	/api/products/{id}/	🔒 ▼
DELETE	/api/products/{id}/	🔒 ▼
GET	/api/schema/	🔒 ▼
POST	/api/token/	▼
POST	/api/users	🔒 ▼
GET	/api/users/me/	🔒 ▼
PUT	/api/users/me/	🔒 ▼
PATCH	/api/users/me/	🔒 ▼
GET	/api/warehouses/	🔒 ▼
POST	/api/warehouses/	🔒 ▼
GET	/api/warehouses/{id}/	🔒 ▼
PUT	/api/warehouses/{id}/	🔒 ▼
PATCH	/api/warehouses/{id}/	🔒 ▼
DELETE	/api/warehouses/{id}/	🔒 ▼

*Documentation 1 généré par Swagger UI - Endpoint de mes routes REST API*

Ce projet m'a permis de :

- Maîtriser un stack full stack moderne (Vue.js + Django),
- Implémenter des transactions atomiques, de la validation métier avancée,
- Créer une SPA performante et responsive avec une gestion de stock en temps réel.

J'ai appris à faire le lien entre interface utilisateur, logique métier, et cohérence de bdd, tout en sécurisant les différents accès, en optimisant la fluidité des échanges front/back.

J'ai aussi particulièrement appris à déployer mes applications de moi-même, en passant par l'installation d'un serveur Raspberry pi et sa configuration. Par l'achat et la configuration d'un DNS pour pouvoir héberger toutes mes applications en lignes.

*Les informations détaillées sont dans le compte rendu, accessible en ligne, via la page du portfolio qui présente cette réalisation professionnelle voir lien au recto de cette fiche)*

<b>BTS SERVICES INFORMATIQUES AUX ORGANISATIONS</b>	<b>SESSION 2025</b>
<b>ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)</b>	
<b>Épreuve E6 - Conception et développement d'applications (option SLAM)</b>	

<b>DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE</b>		<b>N° réalisation : 2</b>
Nom, prénom : Léo MERKEL		N° candidat : 2351134646
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>	Date : .27. / .04. / .2025..
<b>Organisation support de la réalisation professionnelle</b> Direction du Numérique DIP, Université de Strasbourg		
<b>Intitulé de la réalisation professionnelle</b> ExportGitLab : application d'exportation d'issues		
Période de réalisation : .2023.-2024..... Lieu : Portique, Université de Strasbourg.....		
Modalité : <input checked="" type="checkbox"/> Seul(e) <input type="checkbox"/> En équipe		
<b>Compétences travaillées</b> <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Concevoir et développer une solution applicative</li> <li><input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative</li> <li><input checked="" type="checkbox"/> Gérer les données</li> </ul>		
<b>Conditions de réalisation<sup>5</sup> (ressources fournies, résultats attendus)</b> Ressources fournis : Cahier des charges. Description du contexte et des besoins. Outils de développements. Résultats attendus : <ul style="list-style-type: none"> <li>- Conception et développement d'une application permettant l'exportation des issues Gitlab au format PDF</li> <li>- Un système d'authentification passant par la CAS (Central Authentication Service) ainsi que le token Gitlab</li> <li>- Plusieurs choix d'exportations, un rendu pdf propre et lisible, un site simple et pratique.</li> </ul>		
<b>Description des ressources documentaires, matérielles et logicielles utilisées<sup>6</sup></b> Langages : Python, Jinja, Javascript Frameworks : Django, Vue.js SGBD : PostgreSQL Serveur web : Gunicorn(WSGI), Nginx Hébergement en ligne : Raspberry PI avec Nginx Proxy Manager IDE : PyCharm Gestion de version : Github Outils : Docker, Postman, Node.js		
<b>Modalités d'accès aux productions<sup>7</sup> et à leur documentation<sup>8</sup></b> Site disponible sur : export.orabis.fr Documentation disponible sur : <a href="https://orabis.fr/documentation-export">https://orabis.fr/documentation-export</a> Code source disponible sur github : <a href="https://github.com/Orabis/export-gitlab">https://github.com/Orabis/export-gitlab</a>		

<sup>5</sup> En référence aux conditions de réalisation et ressources nécessaires du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

<sup>6</sup> Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

<sup>7</sup> Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

<sup>8</sup> Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

**ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle  
(verso, éventuellement pages suivantes)****Épreuve E6 - Conception et développement d'applications (option SLAM)****Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs**

Dans le cadre de mon BTS SIO option SLAM, pendant ma période de formation en entreprise, j'ai été sollicité pour développer une application web interne à l'Université de Strasbourg. Le besoin, exprimé par un Product Owner, était de créer un outil simple permettant d'exporter les issues de projets GitLab au format PDF, afin de faciliter la présentation des tickets lors des réunions de suivi.

Le projet a été réalisé seul, avec le soutien de mon tuteur pour les questions relatives au fonctionnement interne et à l'implémentation des technologies de l'université de Strasbourg, dans un contexte professionnel réel avec des contraintes de sécurité. Afin de pouvoir présenter cette application, j'ai ensuite modifié l'application pour permettre à tous de pouvoir exporter leurs propres issues liées à leurs projets personnels.

Avant la mise en place de ce projet, il n'existait aucun outil interne permettant de :

- Extraire facilement les issues GitLab d'un projet donné,
- Générer un PDF clair et organisé.

Les équipes devaient effectuer cette extraction manuellement via GitLab, ce qui était long, peu ergonomique et difficilement présentable.

L'objectif du projet était donc de concevoir une application web permettant d'exporter les issues GitLab sélectionnées dans un document PDF structuré, en proposant plusieurs modes d'exportation, avec une interface simple et directe. L'application intègre également le système d'authentification CAS et une sécurité supplémentaire avec le token GitLab de l'utilisateur.

Fonctionnalités principales :

- Ajout des projets pour lesquels on souhaite exporter les issues, stockés dans une base de données pour éviter de devoir rechercher les projets à chaque fois.
- Exportation des issues en PDF avec WKHTMLTOPDF, proposant plusieurs modes d'exportation (toutes les issues en un fichier ou chaque issue séparée), ainsi qu'une template HTML pour une meilleure lisibilité.
- Sécurisation des accès avec l'authentification CAS de l'Université et ajout d'un token GitLab, limitant ainsi l'accès uniquement aux utilisateurs authentifiés.
- Ajout de tests unitaires pour valider les modifications ou les appels à l'API GitLab.

Ce projet a été entièrement réalisé par mes soins, en utilisant les ressources mises à ma disposition par l'Université. J'ai travaillé avec Django, leur propre template, Bootstrap, et le serveur interne de l'Université.

Project ID: 67410665 - Orabis / Projet sérieux - Issues

Filtering issues :

ID:

1,2,3,4

☐

les soucis

☐

non

Status :

☒ Open

☐ Fermer

Download issues selections :

☐ Download separately

☒ Download in group

Download

☐ Check all

☐ 2 - Améliorer la performance du backend Orabis

les soucis

☐ 1 - Correction du bug d'affichage de la page d'accueil Orabis

les soucis

non

Screenshot 1 - Page de sélection d'issue à exporter

## Projet sérieux - ID 2

### Améliorer la performance du backend

Nous avons constaté que le temps de réponse de l'API augmente de manière significative lorsqu'il y a un grand nombre de requêtes simultanées. Cela impacte l'expérience utilisateur. Nous devons analyser et optimiser le code backend pour réduire ce délai.

## Projet sérieux - ID 1

### Correction du bug d'affichage de la page d'accueil

Sur certains navigateurs, la page d'accueil ne se charge pas correctement, avec des éléments qui se superposent. Ce bug doit être corrigé pour assurer une bonne expérience utilisateur sur tous les navigateurs.

Tâches que j'ai réalisées :

- Création de l'interface utilisateur avec Vue.js intégrée aux templates Django (Jinja).
- Mise en place d'un système de connexion sécurisé : création de comptes, gestion des tokens GitLab, et intégration de la surcouche SSO via le CAS de l'Université de Strasbourg.
- Génération dynamique de PDF à partir de modèles HTML.
- Développement de pages de visualisation des projets et des issues à exporter.
- Écriture de tests unitaires en Python avec simulation (mock) des appels à l'API GitLab.

Ce que ce projet m'a apporté :

- Une réelle adaptation aux contraintes d'entreprise, notamment l'utilisation de technologies spécifiques comme le système d'authentification CAS dans un environnement sécurisé, et l'intégration du framework Django.
- Le développement complet d'une application web fullstack en utilisant Django pour le back-end et Vue.js pour le front-end.
- Une bonne compréhension de l'interfaçage entre un back-end Python et l'API GitLab, tout en respectant les bonnes pratiques d'intégration.
- Une montée en compétence sur Docker, utilisé à la fois en développement et en production, notamment avec la conteneurisation de WKHTMLTOPDF.
- La capacité à structurer un front-end intégré à des templates serveur tout en maintenant une interface utilisateur claire, fluide et réactive.

Ce projet m'a surtout permis de me confronter aux réalités d'un environnement professionnel : délais serrés, demandes spécifiques, contraintes techniques. J'ai appris à être plus flexible, à mieux comprendre le cycle de vie complet d'un projet en entreprise. De la demande initiale jusqu'au déploiement final. Travailler dans une architecture aussi complète m'a énormément fait progresser.

*Les informations détaillées sont dans le compte rendu, accessible en ligne, via la page du portfolio qui présente cette réalisation professionnelle voir lien au recto de cette fiche)*



## CONTRÔLE DE L'ENVIRONNEMENT TECHNOLOGIQUE

En référence à l'annexe II.E –« Environnement technologique pour la certification » du référentiel du BTS SIO

Identification <sup>2</sup>	2351134646 MERKEL Léo	SLAM
-----------------------------	-----------------------	------

## 1. Environnement commun aux deux options

## 1.1 L'environnement technologique supportant le système d'information de l'organisation cliente comporte au moins :

Éléments	Description de l'implantation dans le centre d'examen (nom du service ou de l'outil et caractéristiques techniques)	Remarques de la commission d'interrogation
Un service d'authentification	JWT Token   Django	
Un SGBD	PostgreSQL	
Un accès sécurisé à internet	Https   Firewall	
Un environnement de travail collaboratif	GitHub   Gitea	
Deux serveurs, éventuellement virtualisés, basés sur des systèmes d'exploitation différents, dont l'un est un logiciel libre ( <i>open source</i> )	Raspberry PI sous linux   Windows 11	

<sup>2</sup> Nom et adresse du centre d'examen ou identification de la personne candidate individuelle (numéro, nom, prénom)

**ANNEXE 10-B (suite) : Modèle d'attestation de respect de l'annexe II.E – Environnement technologique pour la certification du référentiel**  
**Épreuve E6 - Conception et développement d'applications (option SLAM)**

Éléments	Description de l'implantation dans le centre d'examen (nom du service ou de l'outil et caractéristiques techniques)	Remarques de la commission d'interrogation
Une solution de sauvegarde	Github	
Des ressources dont l'accès est sécurisé et soumis à habilitation	SSH   Droits accès utilisateurs linux	
Deux types de terminaux dont un mobile (type <i>smartphone</i> ou encore tablette)	Ordinateur de bureau   Smartphone	

**1.2 Des outils sont mobilisés pour la gestion de la sécurité :**

Éléments	Description de l'implantation dans le centre d'examen (nom du service ou de l'outil et caractéristiques techniques)	Remarques de la commission d'interrogation
Gestion des incidents	Github	
Détection et prévention des intrusions	Non traité	
Chiffrement	Certificats SSL   serveur https	
Analyse de trafic	Google Search Console	

**Remarque : les logiciels de simulation ou d'émulation sont utilisés en réponse à des besoins de l'organisation. Ils ne peuvent se substituer complètement à des équipements réels dans l'environnement technologique d'apprentissage.**

**ANNEXE 10-B (suite) : Modèle d'attestation de respect de l'annexe II.E – Environnement technologique pour la certification du référentiel**  
**Épreuve E6 - Conception et développement d'applications (option SLAM)**

**2. Savoirs spécifiques à l'option « solutions logicielles et applications métiers » (SLAM)**

**2.1 L'environnement technologique supportant le système d'information de l'organisation cliente comporte au moins :**

<b>Éléments</b>	<b>Description de l'implantation dans le centre d'examen (nom du service ou de l'outil et caractéristiques techniques)</b>	<b>Remarques de la commission d'interrogation</b>
Un ou deux environnements de développement disposant d'outils de gestion de tests et supportant un cadre applicatif ( <i>framework</i> ) et au moins deux langages	Visual Studio Code   WebStorm	
Une bibliothèque de composants logiciels	Bootstrap   Vue.js	
Un SGBD avec langage de programmation associé	PostgreSQL	
Un logiciel de gestion de versions et de suivi de problèmes d'ordre logiciel	GitHub   Gitea	
Une solution permettant de tester les comportements anormaux d'une application	PostMan	

**2.2 Les activités de l'organisation cliente s'appuient sur aux moins deux solutions applicatives opérationnelles permettant d'offrir un accès sécurisé à des données hébergées sur un site distant. Au sein des architectures de ces solutions applicatives doivent figurer l'exploitation de mécanismes d'appel à des services applicatifs distants et au moins trois des situations ci-dessous :**

Éléments	Description de l'implantation dans le centre d'examen (nom du service ou de l'outil et caractéristiques techniques)	Remarques de la commission d'interrogation
Du code exécuté sur le système d'exploitation d'une solution technique d'accès fixe (type client lourd)	Non traité	
Du code exécuté dans un navigateur Web (type client léger ou riche)	Application web HTML   CSS   JS	
Du code exécuté sur le système d'exploitation d'une solution technique d'accès mobile	Non traité	
Du code exécuté sur le système d'exploitation d'un serveur	REST API avec django rest framework	

**2.3 Une solution applicative peut être issue d'un développement spécifique ou de la modification du code d'un logiciel notamment open source.**

**2.4 Les solutions applicatives présentes dans le contexte sont opérationnelles et leur code source est accessible dans un environnement de développement opérationnel au moment de l'épreuve.**