BTS SERVICES INFORMATIQUES AUX ORGANISATIONS **SESSION 2026**

Épreuve E5 - Conception et développement d'applications (option SLAM)

ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle (recto)

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE

N° réalisation : 5

Nom, prénom : Juhasz Klaudia **N° candidat :**

Épreuve ponctuelle ☐ **Contrôle en cours de formation** ☒ **Date :** 15 / 02 /2026

Organisation support de la réalisation professionnelle

Intitulé de la réalisation professionnelle

CryptoVault – Coffre-fort numérique sécurisé

Période de réalisation : 24/11/2025 – 15/02/2026 **Lieu :** Nice

Modalité : ☐ **Seul(e)** ☒ **En équipe**

Compétences travaillées

☒ Concevoir et développer une solution applicative
☒ Assurer la maintenance corrective ou évolutive d'une solution applicative
☒ Gérer les données

Conditions de réalisation[1] (ressources fournies, résultats attendus)

Ressources fournies :

- Cahier des charges et sujet pédagogique définissant les objectifs fonctionnels et techniques.
- Base applicative existante côté backend (API REST Slim avec Medoo).
- Modèle de données relationnel initial.
- Contrat d'API formalisé via spécification OpenAPI.
- Environnement de développement configuré (Docker, PHP 8.2, MySQL/MariaDB, Java 17+, Maven).
- Outils de développement collaboratif (Git, GitHub, gestion de versions).
- Documentation technique partielle et supports pédagogiques.

Résultats attendus :

- Mise en œuvre d'un service applicatif sécurisé et opérationnel.
- Implémentation d'une authentification sécurisée (JWT) et gestion des habilitations (utilisateur / administrateur).
- Gestion complète des ressources numériques : fichiers, dossiers hiérarchiques, versionnage, partages sécurisés, quotas de stockage.
- Développement d'un client lourd JavaFX pour la gestion des ressources.
- Mise en place d'une interface web légère pour le téléchargement.
- Intégration complète entre les clients et l'API REST, mise en place de tests fonctionnels.
- Rédaction d'une documentation technique, d'un guide utilisateur et les README.md
- Présentation d'un MVP fonctionnel démontrant la cohérence globale de la solution.

Description des ressources documentaires, matérielles et logicielles utilisées[2]

Ressources documentaires

- cahier des charges et sujet pédagogique du projet *Coffre-fort numérique*,
- documentation officielle des technologies utilisées (PHP, Framework Slim, Medoo, JSON Web Tokens (JWT), JavaFX, MySQL),
- spécification de l'API via le contrat OpenAPI (Swagger),
- supports pédagogiques fournis en cours, échanges techniques et validations avec l'enseignant référent.

Ressources matérielles :

- Ordinateur personnel sous Windows 11,
- accès à un serveur SISR pour les tests de déploiement***,
- environnement de virtualisation et de conteneurisation via Docker.

Ressources logicielles :

- **back-end :** PHP 8.2, Framework Slim 4, ORM Medoo, base de données MySQL, Authentification JWT (HMAC-SHA256)
- **sécurité :** gestion des rôles (user/admin), Chiffrement AES-256-GCM des fichiers, Hachage des mots de passe,Middleware de sécurité
- **client lourd :** Java 17, JavaFX 21, Maven (gestion de build), FXML et SceneBuilder,
- **client léger :** Interface Web HTML/CSS/JavaScript, Bootstrap 5, Intégration avec l'API REST
- **outils de développement :** Docker/ Docker Compose, Git/GitHub, Visuel Studio Code, IntelliJ IDEA, Postman (test d'API).

Modalités d'accès aux productions[3] et à leur documentation[4] :

- Code source *du **back-end (API REST)*** disponible sur GitHub : https://github.com/PlumCreativ/coffreFort.git
- Code source du ***client lourd JavaFX*** disponible sur GitHub : https://github.com/PlumCreativ/coffreFortJava.git
- Application accessible en local via Docker : Site : http://localhost:9081 / PhpMyAdmin : http://localhost:8081
- La maquette de l'interface utilisateur a été réalisée avec Figma. Lien vers la maquette: https://www.figma.com/design/QMpU6uu0tGXpdInYixZ798/Mon_portfolio?node-id=366-2&p=f&t=AWQGb1XKf44sCC5L-0

[1] 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.

[2] Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

[3] 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.

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS **SESSION 2026**

Épreuve E5 - Conception et développement d'applications (option SLAM)

ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle (verso, éventuellement pages suivantes)

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

Objectif du projet est de concevoir et déployer un coffre-fort numérique sécurisé permettant :

- La gestion de fichiers par utilisateur
- Le stockage organisé en dossiers hiérarchiques
- La gestion des versions de fichiers
- Le contrôle des quotas de stockage
- Le partage sécurisé de fichiers ou dossiers via un client léger Web

Le projet s'inscrit dans une architecture client–serveur complète intégrant sécurité, gestion des données et interface utilisateur.

Description technique de la solution :

La solution repose sur une architecture client–serveur organisée autour d'une API REST centralisant la logique métier et la sécurité.
Chaque utilisateur dispose d'un espace personnel structuré en dossiers hiérarchiques, permettant le stockage et le versionnage des fichiers.
L'accès aux données est sécurisé par authentification et contrôle des habilitations.
Le système intègre une gestion dynamique des quotas ainsi qu'un mécanisme de partage sécurisé via liens temporaires avec contrôle d'expiration et de téléchargements.
L'ensemble garantit la cohérence des données et la traçabilité des actions.

Fonctionnalités principales :

- **Gestion des utilisateurs :** inscription, authentification JWT, rôles, gestion des quotas (admin), suppression de comptes (admin).
- **Gestion des fichiers et dossiers :** création, renommage, suppression, upload multipart, versionnage.
- **Gestion des quotas :** calcul dynamique de l'espace utilisé, blocage des uploads en cas de dépassement, indicateur visuel dans le client.
- **Partages sécurisés :** génération de liens uniques, expiration, limitation du nombre de téléchargements, révocation, téléchargement via interface Web.
- **Journalisation :** traçabilité des téléchargements et des actions sensibles.

Organisation du code

Backend (exemples)

- public/index.php : point d'entrée de l'API Slim.
- src/Controller/ : contrôleurs REST (authentification, fichiers, dossiers, partages, administration).
- src/Model/ : accès aux données via Medoo.
- sql/init.sql : création des tables (users, folders, files, file_versions, shares, downloads_log, audit_logs).
- docker-compose.yml : orchestration des services.

Client lourd JavaFX

- Interfaces graphiques définies en FXML (login, inscription, tableau de bord, gestion des partages, gestion des versions).
- Contrôleurs JavaFX pour la gestion des événements.
- Arborescence des dossiers (TreeView).
- Liste des fichiers (TableView).
- Upload avec barre de progression.
- Client HTTP dédié pour les appels REST

Client légPage Web de consultation d'un lien de partage

- Affichage des informations (nom, taille, expiration, téléchargements restants)
- Téléchargement direct du fichier
- Téléchargement d'un dossier sous forme d'archive ZIP
- Vérification des contraintes côté serveur

Productions livrées

- API REST complète et sécurisée
- Client JavaFX connecté à l'API, client Web de téléchargement public
- Base de données relationnelle normalisée, scripts SQL
- Environnement Docker prêt à l'exécution, code versionné sur GitHub
- Documentation technique, guide utilisateur
- Captures d'écran et maquette Figma

Des schémas explicatifs accompagnent cette réalisation :

- MCD de la base de données.
- Schéma des flux principaux (authentification, upload, téléchargement).

Représentation des flux typiques de l'application *:*

Authentification

Client JavaFX → ***POST /auth/login*** → API Slim → vérification BDD → génération JWT → accès aux ressources protégées.

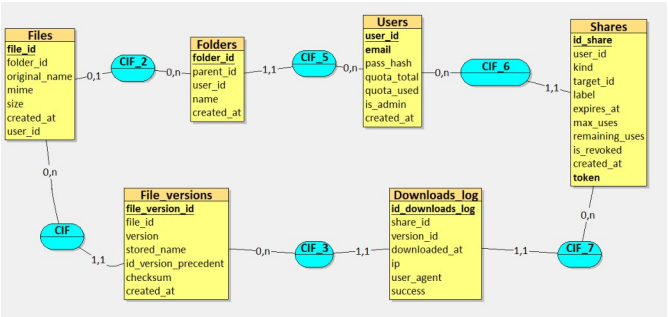
Navigation et upload

Client JavaFX → ***GET /folders*** *→ affichage arborescence*

Client JavaFX → ***POST /files*** *(multipart) → chiffrement et stockage fichier → mise à jour quota → confirmation.*

Téléchargement via client léger

Navigateur → accès via lien sécurisé → API REST → Vérification validité (expiration / usages / révocation) → téléchargement du fichier.



[4] 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.