BTS SERVICES INFORMATIQUES AUX ORGANISATIONS **SESSION 2025**

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 :** 19 / 12 /2025

Organisation support de la réalisation professionnelle

Intitulé de la réalisation professionnelle

MédiaStock – Application de gestion de matériel

Période de réalisation : 24/11/2025 – 19/12/2025 **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,
- base applicative existante (API REST Slim/Medoo),
- modèle de données relationnel,
- contrat d'API OpenAPI,
- environnement Docker et outils de développement collaboratif.

Résultats attendus :

- service informatique sécurisé et opérationnel,
- authentification JWT et gestion des habilitations,
- gestion des ressources numériques (fichiers, dossiers, quotas),
- client lourd JavaFX pour la gestion et client léger Web pour le téléchargement via liens sécurisés, interfacés avec l'API,
- tests, documentation et présentation d'un MVP fonctionnel.

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, Slim, Medoo, JWT, JavaFX),
- spécification de l'API via le contrat OpenAPI (Swagger),
- supports pédagogiques et échanges 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, framework Slim, ORM Medoo, base de données MySQL,
- sécurité : authentification JWT, gestion des rôles et habilitations,
- client lourd : JavaFX (Maven, FXML, SceneBuilder),
- client léger : application Web pour le téléchargement des fichiers partagés,
- outils : Docker, Git/GitHub, VS Code, IntelliJ IDEA, Postman.

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
- base de données initialisée automatiquement à partir du script ***sql/init.sql*** lors du build des conteneurs Docker

[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.

[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.

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS **SESSION 2025**

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 outil de gestion d'inventaire permettant :

Concevoir un coffre-fort numérique sécurisé permettant la gestion de fichiers par utilisateur, le stockage organisé en dossiers, le contrôle des quotas et le partage via un client léger Web.

Description technique

- Architecture client–serveur basée sur une API REST.
- Back-end : PHP 8 (Slim), ORM Medoo, base MySQL.
- Sécurité : authentification JWT.
- Client lourd : application JavaFX.
- Client léger Web : téléchargement de fichiers partagés.
- Déploiement conteneurisé avec Docker Compose.

Fonctionnalités principales :

- Authentification et gestion des utilisateurs.
- Gestion des dossiers et fichiers.
- Upload et suppression de fichiers.
- Gestion des quotas de stockage.
- Téléchargement sécurisé via client Web.

Organisation du code

Backend (exemples)

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

Client lourd JavaFX

- Interfaces graphiques définies en FXML (login, inscription, tableau de bord).
- Contrôleurs JavaFX pour la gestion des événements utilisateur.
- Navigation dans l'arborescence des dossiers (TreeView).
- Affichage des fichiers par dossier (TableView).
- Upload de fichiers avec indication de progression.
- Communication avec l'API REST via un client HTTP dédié.

Client léger Web

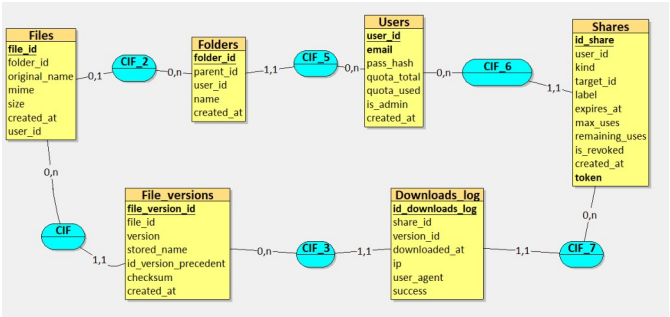
- Page Web simple permettant le téléchargement de fichiers partagés.
- Accès via lien sécurisé sans authentification utilisateur.
- Appels REST vers l'API pour la récupération des ressources.

Productions livrées

- API REST fonctionnelle.
- Client JavaFX connecté à l'API.
- Client Web de téléchargement.
- Scripts SQL, environnement Docker et code versionné.
- Documentation technique et captures d'écran

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.

Navigation et upload

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

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

Téléchargement via client léger

Navigateur → lien sécurisé Web → API → contrôle des droits → téléchargement du fichier.