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 : 1**
Nom, prénom : Juhasz Klaudia	**N° candidat :**
Épreuve ponctuelle ☐ **Contrôle en cours de formation** ☒	**Date :** .20 / 06 /2025
Organisation support de la réalisation professionnelle	
Intitulé de la réalisation professionnelle **ClassCord - Client de messagerie interopérable**	
Période de réalisation : 16/06/2025 – 20/06/2025 **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 fonctionnel et technique détaillé.
- Serveur de test opérationnel mis à disposition par le formateur.
- Dépôt GitHub modèle à forker pour démarrer le projet.
- Exemples de messages au format JSON (connexion, messagerie, statut).
- Accompagnement pédagogique tout au long de la semaine.

Résultats attendus : Une application cliente fonctionnelle permettant :

- Connexion à un serveur de chat
- Envoi/réception de messages globaux et privés
- Gestion des statuts utilisateurs
- Interface Swing fluide et conviviale
- Communication en JSON avec le serveur

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

Ressources documentaires

- Cahier des charges fourni par le formateur et documentation Maven & org.json
- Cours Java (Sockets, Swing, Threads)
- Aide-mémoire Markdown pour le README et tutoriels en ligne (Java Swing, communication réseau)

Ressources matérielles

- Poste informatique personnel ou mis à disposition en salle BTS SIO
- Connexion au réseau local pour tests avec serveur et clients

Environnement logiciel ***(veuillez le trouver dans l'Annexe 1)***

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

Mon projet se trouve sur mon GitHub : https://github.com/juklau/classcord-client2.git où vous pouvez trouver un README.md en détaillant ma réalisation.

Pour lancer mon projet il faut aller dans la classe ConnectToServeurUI

voici son chemin :***classcord-client/src/main/java/fr/classcord/ui/ConnectToServeurUI.java***

---

[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

Contexte et objectifs

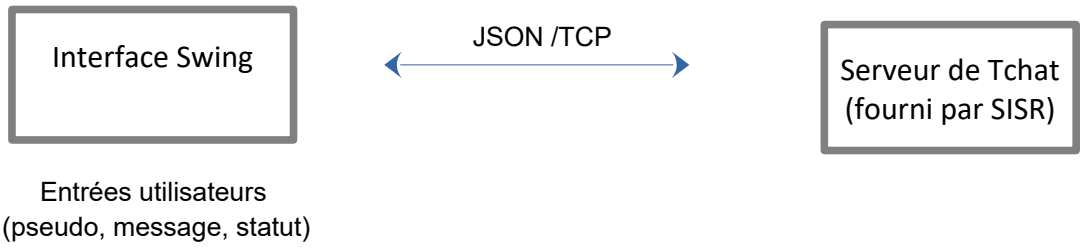
Dans le cadre de la semaine intensive SLAM (BTS SIO 2024), j'ai participé au développement de **ClassCord**, une application Java Swing de messagerie instantanée en réseau local. Cette application cliente interagit avec un serveur TCP (fourni ou géré par les étudiants SISR) via des échanges JSON pour offrir une interface fluide, sécurisée et réactive aux utilisateurs (étudiants et enseignants).

Architecture mise en œuvre ***(veuillez le trouver dans l'Annexe 2)***

Fonctionnalités développées

- Connexion à un serveur distant (saisie IP/port)
- Connexion en tant qu'invité ou avec authentification (login/mot de passe)
- Affichage des messages globaux en temps réel
- Envoi de messages privés avec distinction visuelle
- Affichage dynamique de la liste des utilisateurs connectés avec leur statut
- Modification de son propre statut (Disponible / Absent / Invisible)
- Interface graphique fluide avec gestion des threads réseau
- Gestion des erreurs (déconnexion propre, identifiants invalides, etc.)
-

Schéma fonctionnel simplifié



Productions réalisées

- Projet Maven complet avec structure professionnelle
- Classes Java métier (User, Message) + logique réseau encapsulée
- Interface Swing moderne, intuitive et modulaire
- Gestion multithreadée pour réception asynchrone
- README.md avec instructions de lancement et captures
- Documentation intégrée (commentaires, structure MVC)

Tests et validation

- Tests multi-clients avec interaction serveur réel
- Vérification des statuts, MP, messages globaux
- Démonstration complète : connexion, tchat, changement de statut, déconnexion propre

Projet mené de bout en bout dans un dépôt GitHub personnel avec commits réguliers et ce projet m'a permis de mobiliser concrètement mes compétences en Java, Swing, réseau, MVC et gestion de projet collaboratif.

Annexe 1

Environnement logiciel de l'application de Chat

Environnement logiciel

- Système d'exploitation : Windows 11
- IDE principal : Visual Studio Code (avec extensions Java) et IntelliJ IDEA
- Gestionnaire des dépendences : Maven
- Dépendance externe : org.json:json:20231013
- Contrôle de version : Git et plateforme GitHub
- Protocole de communication : Sockets TCP avec messages JSON (terminés par \n)
- Langage de programmation : Java 11 ou supérieur

Annexe 2

L'application repose sur l'architecture MVC (Modèle-Vue-Contrôleur) :

```
    ├── src/
    │   ├── main/
    │   │   ├── java/
    │   │   │   ├── fr/
    │   │   │   │   ├── classcord/
    │   │   │   │   │   ├── controller/
    │   │   │   │   │   │   ├── AuthController
    │   │   │   │   │   │   ├── ChatController
    │   │   │   │   │   │   ├── LoginController
    │   │   │   │   │   │   ├── SessionController
    │   │   │   │   │   ├── model/
    │   │   │   │   │   │   ├── ClientInvite
    │   │   │   │   │   │   ├── CurrentUser (inactif)
    │   │   │   │   │   │   ├── User.java (inactif)
    │   │   │   │   │   │   ├── Message.java (inactif)
    │   │   │   │   │   │   ├── UserColorManager
    │   │   │   │   │   ├── ui/
    │   │   │   │   │   │   ├── ChatPersoUI
    │   │   │   │   │   │   ├── ChatUI
    │   │   │   │   │   │   ├── ChoixModeUI
    │   │   │   │   │   │   ├── ConnectToServeurUI
    │   │   │   │   │   │   ├── GuestUI
    │   │   │   │   │   │   ├── LoginUI
    │   │   │   │   │   │   ├── UserStatusRenderer
    │   │   │   │   │   ├── app/
    │   │   │   │   │   │   ├── App (inactif)
    ├── pom.xml
```