

Compétences	D1.2 – Choix d’une solution – A1.2.4 Détermination des tests nécessaires à la validation d’un service D1.3 – Mise en production d’un service – A1.3.1 Tests d’intégration et d’acceptation d’un service – A1.3.3 Accompagnement de la mise en place d’un nouveau service – A4.1.10 Rédaction d’une documentation d’utilisation		
	A4.1.1 Proposition d’une solution applicative A4.1.2 Conception ou adaptation de l’interface utilisateur d’une solution applicative A4.1.3 Conception ou adaptation d’une base de données A4.1.4 Définition des caractéristiques d’une solution applicative A4.1.5 Prototypage de composants logiciels A4.1.6 Gestion d’environnements de développement et de test A4.1.7 Développement, utilisation ou adaptation de composants logiciels	Non Non Non Non Non Non Non	C4.1.1.1 Identifier les composants logiciels nécessaires à la conception de la solution C4.1.1.2 Estimer les éléments de coût et le délai de mise en œuvre de la solution C4.1.2.1 Définir les spécifications de l’interface utilisateur de la solution applicative C4.1.2.2 Maquetter un élément de la solution applicative C4.1.2.3 Concevoir et valider la maquette en collaboration avec des utilisateurs C4.1.3.1 Modéliser le schéma de données nécessaire à la mise en place de la solution applicative C4.1.3.2 Implémenter le schéma de données dans un SGBD C4.1.3.3 Programmer des éléments de la solution applicative dans le langage d’un SGBD C4.1.3.4 Manipuler les données liées à la solution applicative à travers un langage de requête C4.1.4.1 Recenser et caractériser les composants existants ou à développer utiles à la réalisation de la solution applicative dans le respect des budgets et planning prévisionnels C4.1.5.1 Choisir les éléments de la solution à prototyper C4.1.5.2 Développer un prototype C4.1.5.3 Valider un prototype C4.1.6.1 Mettre en place et exploiter un environnement de développement C4.1.6.2 Mettre en place et exploiter un environnement de test C4.1.7.1 Développer les éléments d’une solution C4.1.7.2 Créer un composant logiciel C4.1.7.3 Analyser et modifier le code d’un composant logiciel C4.1.7.4 Utiliser des composants d’accès aux données C4.1.7.5 Mettre en place des éléments de sécurité liés à l’utilisation d’un composant logiciel
Objectif principal		Construire une base de données gérée en Java pour suivre les stocks de l’association Web courses	
Objectifs intermédiaires		- Créer une base de données nommée Java stocks - Structurer cette base de données - Intégrer les différents éléments dans cette base de données à l’aide de java - Produire un ensemble cohérent pour l’examen E4 - Produire une documentation technique relative à l’ensemble des activités	
Outils utilisés		Looping, IntelliJ IDEA, PostGreSQL, MAMP	
Mode de travail	Seuls ou en groupe selon leur préférence		
Mode d’évaluation	- Validation des missions au fil de l’eau, en direct, sur une base personnelle et selon son rythme jusqu’à la fin de chaque semestre - Compte-rendu à rendre pour évaluation sommative à la fin de chaque semestre → coefficient 2 - Chaque compte-rendu est à poser dans le PFC - Épreuve E4 certificative		

Sommaire

Informations générales sur Java Stocks	2
Informations techniques sur Java Stocks	3
Installation de votre plateforme de travail	3
Structure du programme	3
Gestion des collections	3
La gestion du garbage collector et exceptions	4
Les tests unitaires	4
La gestion des bugs	4
Conception de l'application	5
Mission 1 Construire le MCD Java stocks et le diagramme de classe	5
Mission 2 Construire la base de données javastocks avec PostgreSQL	7
Mission 3 Construire la maquette de l'application de gestion de stock de Java stocks	9
Gestion de l'application	13
Mission 1 Gérer l'application de gestion de stock de Java stocks	13
Mission 2 Gérer le réapprovisionnement de Java stocks	15
Maintenance de l'application	19
Mission 1 maintenance évolutive	19
Mission 2 Compléter une application	21
Tester l'application	23
Mission 1 Gérer les tests de développement par flux	23
Mission 2 Gérer la planification des tâches avec Trello	24
Documenter l'application	25
Mission 1 Documenter l'application de gestion de stock de Java stocks	25

Informations générales sur Java Stocks

L'association Web courses gère ses stocks afin d'offrir le meilleur à ses participants. Dans un premier temps, l'administrateur de gestion des stocks utilise la réservation de boisson, textile et denrée sèche pour un coureur pour une épreuve donnée. Il peut ainsi afficher, modifier, consulter ou créer une nouvelle réservation.

Dans un second temps, cela lui permet d'avoir une visibilité sur ses stocks, et peut ainsi gérer les ruptures et demander le réapprovisionnement des produits en cas de besoin.

L'application est développée en Java, avec IntelliJ Idea.

Elle est gérée par un gestionnaire de stocks, employé à temps plein.

Informations techniques sur Java Stocks

Installation de votre plateforme de travail

Utiliser le TD Java Chap1b TD La plateforme java pour installer votre plateforme de travail

Le nom de votre projet : **esm.javastocks**

Nom du package : **JavaStocks**

Structure du programme

Vous devez dans un premier temps, gérer 4 classes et 3 classes héritières.

Les classes héritières utilisent le mot-clef **extends** pour étendre leur héritage par rapport à la classe mère.

Afin d'optimiser le programme, vous devrez créer 4 méthodes génériques :

- CRER
- MODIFIER
- CONSULTER
- SUPPRIMER

En utilisant des surcharges sur ces méthodes.

Pour vous aider, consulter la mission 2 du Chap3c TD création de classe java - méthodes et exceptions.

Pour vous aider dans la création de l'interface, consulter la mission 2 Java Chap3d TD corrigé création de classe java - interfaces et collections.

Penser à déclarer vos accesseurs et à instancier vos objets.

Dans le programme principal, utiliser do/while pour la création du menu principal mais aussi pour vos sous-menus.

Utiliser le case/break pour gérer vos différents cas.

Pour la saisie et l'exploitation des informations saisies par l'utilisateur, utiliser :

Integer.parseInt() pour transformer une chaîne de caractère en entier

```
Scanner sc = new Scanner(System.in);  
System.out.println("Veuillez saisir un mot :");  
String str = sc.nextLine();  
System.out.println("Vous avez saisi : " + str);
```

Pour récupérer et traiter une information saisie au clavier

Gestion des collections

Pour gérer votre code, vous aurez besoin de différentes collections. La gestion des imports à utilis ;er seront les suivantes :

Java.io.* → pour les accès sur la base de données

Java.util.*
Java.lang.*

Pour la gestion des formules mathématiques, la gestion de lecture console, la gestion des tris etc.....

Java.org.unit.* → pour la gestion des tests unitaires

Org.java.api.*
Javax.swing.JFrame
Java.awt.Color
Java.swing.*

Pour la gestion du mode graphique

La gestion du garbage collector et exceptions

Afin de pouvoir libérer les ressources utilisées en mémoire, il faudra gérer le **garbage collector**.

Vous devrez donc gérer pour la présence ou non des fichiers, des accès en lectures/écriture à la base de données.

Pour cela l'utilisation de try/catch/finally pour sera d'une grande aide.

Pour vous aider, consulter la mission 3 Java Chap3c TD création de classe java - méthodes et exceptions

Les tests unitaires

Afin de valider la solidité de votre application, vous serez amené à la tester dans des cas extrêmes.

Pour gagner cette fiabilité, vous allez imaginer les pires cas d'utilisation de vos objets car en développement la règle d'or est : « NE JAMAIS FAIRE CONFIANCE A UN UTILISATEUR ».

Pour vous aider dans cette tâche complexe, IntelliJ IDEA propose directement la préparation d'un ensemble de test avec lesquels vous pourrez jouer en totalité.

Les résultats des tests sont synthétisés dans une vue type arbre.

La gestion des bugs

Lors de la conception de votre application, vous serez confronté à des bugs.

IntelliJ IDEA met à votre disposition

- Le debug dans la console
- Le mot-clef assert

Pour vous aider dans l'utilisation du mot-clef assert, consulter la mission 3 Java Chap2 TD

Les types java

Conception de l'application

Mission 1 Construire le MCD Java stocks et le diagramme de classe

Compétences	• C1.1.1.2 Identifier les fonctionnalités attendues du service à produire • C1.1.1.1 Recenser et caractériser les contextes d'utilisation, les processus et les acteurs sur lesquels le service à produire aura un impact • C1.1.2.1 Analyser les interactions entre services
Vocabulaire à connaître	
Evaluation	Production d'écrit DST E4

Objectif principal

- **Créer** le MDC et le MLD de Java Stock

Objectifs intermédiaires

- **Réinvestir** les notions de cours en base de données

Outils nécessaires

- Suite Microsoft ou Libre Office
- IntelliJ IDEA, Visual Code ou Visual Studio 2022
- VirtualBox ou réel
- PostGréSQL

Pré-requis

- Cours de base de données

Votre mission :

- **Construire** le **MCD** de Java Stock selon les informations ci-dessous
- **Construire** le **MLD** de Java Stock selon les informations ci-dessous
- **Construire** le **diagramme de classe** de Java Stock selon les informations ci-dessous

Informations

Afin de gérer au mieux les événements sportifs, l'association Web courses doit gérer un stock de produits pour aider les coureurs dans leurs efforts. Les articles principaux sont les maillots, les boissons et de la nourriture sèche.

Voici les règles et contraintes du système :

Les articles de Java stocks sont classifiés en 3 catégories :

- Textile
- Boisson
- Denrée sèche

Pour chaque catégorie, on gère les informations suivantes :

- Pour le textile → la taille et la couleur
- Pour la boisson → le volume de la bouteille
- Pour la denrée sèche → le poids

Un article ne peut appartenir qu'à une seule catégorie. Il possède un libellé et une quantité permettant de connaître son stock disponible.

Lorsqu'un article ne convient, il est supprimé logiquement de la base. Pour cela, un indicateur SL est associé à l'article.

Afin de préparer les épreuves, le gestionnaire de stock réserve à une date donnée, pour un coureur et un type d'épreuve donnés, une liste d'articles déterminée.

Le coureur dispose d'un nom et d'un prénom. Pour le type d'épreuve, un libellé est précisé.

Exemple de réservation

Numéro de réservation : 123456789

Date : 23/03/2024

Coureur :

id 58 Nom Berthollo Prénom Pierre

Type épreuve :

id 45 La courses folle junior

Liste des articles réservés

Id 5	teeshirt	vert	taille	L		qté 1
Id 3	boissons				50cl	qté 6
Id 8	Barre énergétique					250g qté 12

Mission 2 Construire la base de données javastocks avec PostgreSQL

Compétences	A4.1.5 Prototypage de composants logiciels	Non	C4.1.5.1 Choisir les éléments de la solution à prototyper
			C4.1.5.2 Développer un prototype
			C4.1.5.3 Valider un prototype
	A4.1.6 Gestion d'environnements de développement et de test	Non	C4.1.6.1 Mettre en place et exploiter un environnement de développement
			C4.1.6.2 Mettre en place et exploiter un environnement de test
	A4.1.7 Développement, utilisation ou adaptation de composants logiciels	Non	C4.1.7.1 Développer les éléments d'une solution
			C4.1.7.2 Créer un composant logiciel
			C4.1.7.3 Analyser et modifier le code d'un composant logiciel
			C4.1.7.4 Utiliser des composants d'accès aux données
			C4.1.7.5 Mettre en place des éléments de sécurité liés à l'utilisation d'un composant logiciel
Vocabulaire à connaître			
Evaluation	Production d'écrit DST E4		

Objectif principal

- **Gérer** la maquette de l'application Java Stock

Objectifs intermédiaires

- **Réinvestir** les notions de cours de java
- **Insérer** des notions de calculs pour la gestion de stock

Outils nécessaires

- Suite Microsoft ou Libre Office
- IntelliJ IDEA, Visual Code ou Visual Studio 2022
- VirtualBox ou réel
- PostGréSQL

Pré-requis

- Cours de base de données
- Cours sur Java

Votre mission :

Créer la base de données javastocks à partir de PostGreSQL sur votre serveur srvjava.
Charger les données de votre base de données afin que cette dernière soit utilisable.
Tester votre base de données à l'aide de requêtes SQL

Guide d'implémentation

Utiliser le lien suivant <https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/26258-jdbc-decouvrez-la-porte-dacces-aux-bases-de-donnees>

Informations utiles

Nom de la base de données : javastocks

Nom des tables :

- Coureur
- Type_epreuve
- Réserver
- Articles

Liste des requêtes à réaliser

- Afficher la liste des produits
- Afficher la liste des coureurs
- Afficher la liste des types_épreuves
- Afficher la liste des réservations
- Afficher la liste des réservations en date du jour courant, pour le coureur Brillat Savarin

Mission 3 Construire la maquette de l'application de gestion de stock de Java stocks

Compétences	A4.1.5 Prototypage de composants logiciels	Non	C4.1.5.1 Choisir les éléments de la solution à prototyper
			C4.1.5.2 Développer un prototype
			C4.1.5.3 Valider un prototype
	A4.1.6 Gestion d'environnements de développement et de test	Non	C4.1.6.1 Mettre en place et exploiter un environnement de développement
			C4.1.6.2 Mettre en place et exploiter un environnement de test
	A4.1.7 Développement, utilisation ou adaptation de composants logiciels	Non	C4.1.7.1 Développer les éléments d'une solution
			C4.1.7.2 Créer un composant logiciel
			C4.1.7.3 Analyser et modifier le code d'un composant logiciel
			C4.1.7.4 Utiliser des composants d'accès aux données
			C4.1.7.5 Mettre en place des éléments de sécurité liés à l'utilisation d'un composant logiciel
Vocabulaire à connaître			
Evaluation	Production d'écrit DST E4		

Objectif principal

- **Créer** la maquette de l'application Java Stock

Objectifs intermédiaires

- **Réinvestir** les notions de cours de java

Outils nécessaires

- Suite Microsoft ou Libre Office
- IntelliJ IDEA, Visual Code ou Visual Studio 2022
- VirtualBox ou réel
- PostGréSQL

Pré-requis

- Cours de base de données
- Cours sur Java

Votre mission :

Créer la maquette de l'application Java stocks à partir des informations ci-dessous

Informations

Menu Principal

- 1- Gestion des articles
- 2- Gestion des coureurs
- 3- Gestion des types d'épreuve

- | |
|---|
| <ul style="list-style-type: none">4- Gestion des réservations5- Article en rupture / réservation en attente6- Consulter l'historique7- Quitter |
|---|

Menu Article

- | |
|---|
| <ul style="list-style-type: none">1- Ajouter un article2- Modifier un article3- Consulter un article4- Suppression logique5- Revenir au menu6- Quitter |
|---|

Menu coureur

- | |
|---|
| <ul style="list-style-type: none">1- Ajouter un coureur |
|---|

Menu types d'épreuve

- | |
|--|
| <ul style="list-style-type: none">1- Ajouter un type d'épreuve |
|--|

Menu réservation

- | |
|--|
| <ul style="list-style-type: none">1- Créer une réservation2- Modifier une réservation3- Consulter une réservation4- Annuler une réservation5- Revenir au menu6- Quitter |
|--|

Menu Article en rupture / réservation en attente
--

- | |
|---|
| <ul style="list-style-type: none">1- Consulter les produits en rupture2- Consulter les réservations en attente de validation3- Revenir au menu4- Quitter |
|---|

Menu historique

- | |
|--|
| <ul style="list-style-type: none">1- Consulter le nombre de réservation à une date donnée2- Consulter le nombre de réservation pour un coureur donné3- Consulter le nombre de réservation pour un type d'épreuve donné |
|--|

- 4- Consulter le nombre de réservation à une date donnée et un type d'épreuve donné
- 5- Consulter les quantités réservées par article à une date donnée
- 6- Consulter les quantités réservées pour un type d'épreuve donné
- 7- Consulter les quantités réservées à une date donnée et un type d'épreuve donné
- 8- Revenir au menu
- 9- Quitter

Création article

- Libellé
- Catégorie : T pour textile, B pour boisson, DS pour denrée sèche
- Taille à choisir dans une liste déroulante
- Couleur à choisir dans une liste déroulante
- Volume
- Poids

AJOUTER

Afficher un message de confirmation de bonne création article avec son ID

Modification article

Choisir l'article sur une liste déroulante et afficher l'article

- ID article
- Libellé
- Catégorie : T pour textile, B pour boisson, DS pour denrée sèche
- Taille
- Couleur
- Volume
- Poids

MODIFIER

Afficher un message de confirmation de modification article avec son ID

Consultation article

Choisir l'article sur une liste déroulante

AFFICHER (l'ensemble des informations)

CONSULTER

Suppression logique article

Choisir l'article sur une liste déroulante

Rajouter l'indicateur SL pour suppression logique

SUPPRIMER

AFFICHER l'ensemble des informations

Créer une réservation

- Date à choisir sur un calendrier ou une liste déroulante
- Coureur à choisir dans une liste déroulante
- Type épreuve à choisir dans une liste déroulante
- ID article + libellé à choisir dans une liste déroulante
- Afficher les caractéristiques de l'article
- Quantité réservée à contrôler pour disponibilité

CREER

Afficher un message de confirmation de bonne création de réservation avec son ID

Modifier une réservationChoisir l'ID sur une liste déroulante et afficher la réservation
AFFICHER (l'ensemble des informations)

- Date à choisir sur un calendrier ou une liste déroulante
- Coureur à choisir dans une liste déroulante
- Type épreuve à choisir dans une liste déroulante
- ID article + libellé à choisir dans une liste déroulante
- Quantité réservée à contrôler pour disponibilité

MODIFIER

Afficher un message de confirmation de bonne modification de réservation avec son ID

Consulter une réservationChoisir l'ID sur une liste déroulante et afficher la réservation
AFFICHER (l'ensemble des informations)**CONSULTER****Supprimer une réservation**Choisir l'ID sur une liste déroulante et afficher la réservation
AFFICHER (l'ensemble des informations)**SUPPRIMER**Supprimer physiquement la réservation dans la base
Réactualiser les quantités des articles
Afficher les nouveaux stocks

Gestion de l'application

Mission 1 Gérer l'application de gestion de stock de Java stocks

Compétences	A4.1.5 Prototypage de composants logiciels	Non	C4.1.5.1 Choisir les éléments de la solution à prototyper
			C4.1.5.2 Développer un prototype
			C4.1.5.3 Valider un prototype
	A4.1.6 Gestion d'environnements de développement et de test	Non	C4.1.6.1 Mettre en place et exploiter un environnement de développement
			C4.1.6.2 Mettre en place et exploiter un environnement de test
	A4.1.7 Développement, utilisation ou adaptation de composants logiciels	Non	C4.1.7.1 Développer les éléments d'une solution
			C4.1.7.2 Créer un composant logiciel
			C4.1.7.3 Analyser et modifier le code d'un composant logiciel
			C4.1.7.4 Utiliser des composants d'accès aux données
			C4.1.7.5 Mettre en place des éléments de sécurité liés à l'utilisation d'un composant logiciel
Vocabulaire à connaître			
Evaluation	Production d'écrit DST E4		

Objectif principal

- **Gérer** la maquette de l'application Java Stock

Objectifs intermédiaires

- **Réinvestir** les notions de cours de java
- **Insérer** des notions de calculs pour la gestion de stock

Outils nécessaires

- Suite Microsoft ou Libre Office
- IntelliJ IDEA, Visual Code ou Visual Studio 2022
- VirtualBox ou réel
- PostGréSQL

Pré-requis

- Cours de base de données
- Cours sur Java

Votre mission :

Gérer l'application Java stocks en lien avec votre base de données javastocks

Commenter votre code

Liens utiles

JDBC (Java DataBase Connectivity) : <https://www.jmdoudoux.fr/java/dej/chap-jdbc.htm>

Informations

La gestion de stocks est principalement axée sur la gestion des articles et des réservations.

Lors d'une réservation, les quantités réservées sont vérifiées pour chaque article par rapport à la quantité disponible pour cet article.

Dans le cas où la quantité disponible est suffisante, la réservation est créée et les stocks de chaque article réajusté.

Dans le cas où l'article risque de passer en rupture, un message d'alerte est affiché pour prévenir le gestionnaire de stocks. Dans ce cas, la réservation n'est pas effectuée mais elle est conservée dans le menu « réservation en attente de validation », sous le même format que la réservation.

Lorsque le produit a été réapprovisionné, le gestionnaire de stock valide la réservation en attente et les stocks des produits sont réajustés.

Mission 2 Gérer le réapprovisionnement de Java stocks

Compétences	• C1.1.1.2 Identifier les fonctionnalités attendues du service à produire • C1.1.1.1 Recenser et caractériser les contextes d'utilisation, les processus et les acteurs sur lesquels le service à produire aura un impact • C1.1.2.1 Analyser les interactions entre services
Vocabulaire à connaître	
Evaluation	Production d'écrit DST E4

Objectif principal

- **Compléter** le MDC, le MLD et l'applicatif de Java Stock

Objectifs intermédiaires

- **Réinvestir** les notions de cours en base de données
- **Réinvestir** les notions de cours de java

Outils nécessaires

- Suite Microsoft ou Libre Office
- IntelliJ IDEA, Visual Code ou Visual Studio 2022
- VirtualBox ou réel
- PostGréSQL

Pré-requis

- Cours de base de données
- Cours sur Java

Votre mission :

- **Compléter** le MCD et le MLD de Java Stock selon les informations ci-dessous
- **Créer** la partie réapprovisionnement de l'application Java stocks à partir des informations ci-dessous.
- **Commenter** votre code

Informations

Le seuil de réapprovisionnement est une quantité limite à partir de laquelle, pour un article, on déclenche une demande de réapprovisionnement auprès d'un fournisseur, pour un point de livraison.

Le seuil varie selon la catégorie de l'article :

- 10 pour un produit textile
- 100 pour un produit boisson
- 100 pour un produit denrée sèche

Chaque demande de réapprovisionnement contient un motif. La liste des motifs est la suivante :

- R réapprovisionnement
- NP nouveaux produits
- UR : urgence réapprovisionnement

Voici un exemple de demande de réapprovisionnement :

Point de livraison						
Nom : entrepôt Bergson						
Rue : ZI de la Marre						
Cp : 94000						
Ville : Créteil						
Tel : 06-12-78-89-11						
Email : entrepot.bergson@outlook.fr						
Fournisseur						
Nom : Sté du textile						
Rue : 15 rue Emile Zola						
Cp : 77000						
Ville : Maux						
Tel : 01-12-88-89-11						
Email : stedutextile@yahoo.fr						
Motif : R réapprovisionnement				Date : 23/03/2024		
Commande numéro : 152012200						
Code article	libellé	taille	couleur	contenance	poids	Qté
1245	Maillot	S	blanche			150
2416	bouteille			50cl		200
5689	barre énergétique				30g	400

La saisie des informations se présente comme suit :

Gestion Fournisseur	
	1- Créer un nouveau fournisseur 2- Modifier un fournisseur 3- Afficher un fournisseur
Gestion du point de livraison	
	1- Créer un nouveau point de livraison 2- Modifier un point de livraison 3- Afficher un point de livraison

Création d'un nouveau fournisseur

- Nom
- Rue
- Cp
- Ville
- Tel
- Email

Modifier un fournisseur

Choisir le fournisseur dans une liste déroulante

Afficher le fournisseur

- Code
- Nom
- Rue
- Cp
- Ville
- Tel
- Email

MODIFIER

Afficher un message de confirmation de modification fournisseur avec son ID

Consulter un fournisseur

Choisir le fournisseur dans une liste déroulante

Afficher le fournisseur

Création d'un nouveau point de livraison

- Nom
- Rue
- Cp
- Ville
- Tel
- Email

Modifier un point de livraison

Choisir le point de livraison dans une liste déroulante

Afficher le point de livraison

- Code

• Nom • Rue • Cp • Ville • Tel • Email	MODIFIER
Afficher un message de confirmation de modification point de livraison avec son ID	

Consulter un point de livraison
Choisir le point de livraison dans une liste déroulante
Afficher le point de livraison

Gestion des demandes de réapprovisionnement
1- Créer une nouvelle demande de réapprovisionnement 2- Modifier une demande de réapprovisionnement 3- Afficher une demande de réapprovisionnement

Créer une demande de réapprovisionnement
Se reporter à l'exemple de demande de réapprovisionnement ci-dessus

Modifier une demande de réapprovisionnement
Choisir le code de la demande de réapprovisionnement dans une liste déroulante
Afficher la demande de réapprovisionnement
MODIFIER
Afficher un message de confirmation de modification de la demande de réapprovisionnement avec son ID

Consulter une demande de réapprovisionnement
Choisir le code de la demande de réapprovisionnement dans une liste déroulante
Afficher la demande de réapprovisionnement

Maintenance de l'application

Mission 1 maintenance évolutive

Compétences	A4.1.5 Prototypage de composants logiciels	Non	C4.1.5.1 Choisir les éléments de la solution à prototyper
			C4.1.5.2 Développer un prototype
			C4.1.5.3 Valider un prototype
	A4.1.6 Gestion d'environnements de développement et de test	Non	C4.1.6.1 Mettre en place et exploiter un environnement de développement
			C4.1.6.2 Mettre en place et exploiter un environnement de test
	A4.1.7 Développement, utilisation ou adaptation de composants logiciels	Non	C4.1.7.1 Développer les éléments d'une solution
			C4.1.7.2 Créer un composant logiciel
			C4.1.7.3 Analyser et modifier le code d'un composant logiciel
			C4.1.7.4 Utiliser des composants d'accès aux données
			C4.1.7.5 Mettre en place des éléments de sécurité liés à l'utilisation d'un composant logiciel
Vocabulaire à connaître			
Evaluation	Production d'écrit DST E4		

Objectif principal

- **Créer** un ticket d'incident et le suivre jusqu'à sa clôture dans l'application Java Stock

Objectifs intermédiaires

- **Réinvestir** les notions de cours de java
- **Créer** un document technique de l'application Java Stock

Outils nécessaires

- Suite Microsoft ou Libre Office
- IntelliJ IDEA, Visual Code ou Visual Studio 2022
- VirtualBox ou réel
- PostgreSQL
- GLPI

Pré-requis

- Cours de base de données
- Cours sur Java

Votre mission

Créer un ticket d'incident auprès du Help Desk de Running Hightech
Affecter lui une priorité
Justifier votre choix
Montrer l'affectation de ce ticket à l'un des développeurs de Running Hightech
Corriger le problème → répondre au besoin du gestionnaire de stock
Tester la modification

Donner la preuve de la validation du test

Commenter le programme

Montrer la clôture du ticket et l'alimentation de ce dernier la base de connaissance

Informations

Le gestionnaire de stock cherche à valider les « réservations en attente de validation » mais ne trouve aucun menu ni moyen pour y parvenir.
Il contacte le Help Desk de Running Hightech.

Mission 2 Compléter une application

Compétences	A4.1.5 Prototypage de composants logiciels	Non	C4.1.5.1 Choisir les éléments de la solution à prototyper
			C4.1.5.2 Développer un prototype
			C4.1.5.3 Valider un prototype
	A4.1.6 Gestion d'environnements de développement et de test	Non	C4.1.6.1 Mettre en place et exploiter un environnement de développement
			C4.1.6.2 Mettre en place et exploiter un environnement de test
	A4.1.7 Développement, utilisation ou adaptation de composants logiciels	Non	C4.1.7.1 Développer les éléments d'une solution
			C4.1.7.2 Créer un composant logiciel
			C4.1.7.3 Analyser et modifier le code d'un composant logiciel
			C4.1.7.4 Utiliser des composants d'accès aux données
C4.1.7.5 Mettre en place des éléments de sécurité liés à l'utilisation d'un composant logiciel			
Vocabulaire à connaître			
Evaluation	Production d'écrit DST E4		

Objectif principal

- **Créer** un ticket d'incident et le suivre jusqu'à sa clôture dans l'application Java Stock

Objectifs intermédiaires

- **Réinvestir** les notions de cours de java
- **Créer** un document technique de l'application Java Stock

Outils nécessaires

- Suite Microsoft ou Libre Office
- IntelliJ IDEA
- VirtualBox
- GLPI

Pré-requis

- Cours de base de données
- Cours sur Java

Votre mission

Créer un ticket d'incident auprès du Help Desk de Running Hightech

Affecter lui une priorité

Justifier votre choix

Montrer l'affectation de ce ticket à l'un des développeurs de Running Hightech

Corriger le problème → répondre au besoin du gestionnaire de stock

Tester la modification

Donner la preuve de la validation du test

Commenter le programme

Montrer la clôture du ticket et l'alimentation de ce dernier la base de connaissance

Informations

Le gestionnaire de stock se rend compte que le menu CREER client et CREER type épreuve n'existe pas.

Il contacte le Help Desk de Running Hightech.

Tester l'application

Mission 1 Gérer les tests de développement par flux

Compétences	A4.1.5 Prototypage de composants logiciels	Non	C4.1.5.1 Choisir les éléments de la solution à prototyper
			C4.1.5.2 Développer un prototype
			C4.1.5.3 Valider un prototype
	A4.1.6 Gestion d'environnements de développement et de test	Non	C4.1.6.1 Mettre en place et exploiter un environnement de développement
			C4.1.6.2 Mettre en place et exploiter un environnement de test
	A4.1.7 Développement, utilisation ou adaptation de composants logiciels	Non	C4.1.7.1 Développer les éléments d'une solution
			C4.1.7.2 Créer un composant logiciel
			C4.1.7.3 Analyser et modifier le code d'un composant logiciel
			C4.1.7.4 Utiliser des composants d'accès aux données
			C4.1.7.5 Mettre en place des éléments de sécurité liés à l'utilisation d'un composant logiciel
Vocabulaire à connaître			
Evaluation	Production d'écrit DST E4		

Objectif principal

- **Créer** un jeu de test unitaire pour valider la solidité de l'application

Objectifs intermédiaires

- **Réinvestir** les notions de cours de java
- **Tester** différents cas d'utilisation sur l'application Java Stock

Outils nécessaires

- Suite Microsoft ou Libre Office
- IntelliJ IDEA
- SELENIUM

Pré-requis

- Cours de base de données
- Cours sur Java

Votre mission

Créer à l'aide du logiciel SELENIUM, les tests de développement par flux

Mission 2 Gérer la planification des tâches avec Trello

Compétences	• C1.1.1.2 Identifier les fonctionnalités attendues du service à produire • C1.1.1.1 Recenser et caractériser les contextes d'utilisation, les processus et les acteurs sur lesquels le service à produire aura un impact • C1.1.2.1 Analyser les interactions entre services
Vocabulaire à connaître	
Evaluation	Production d'écrit DST E4

Objectif principal

- **Utiliser** le logiciel TRELLO pour gérer la planification des tâches

Objectifs intermédiaires

- **Réinvestir** les notions de cours en gestion de projets

Outils nécessaires

- TRELLO

Votre mission

Créer à l'aide du logiciel Trello, la planification des tâches du projet

Documenter l'application

Mission 1 Documenter l'application Javastocks

Compétences	A4.1.5 Prototypage de composants logiciels	Non	C4.1.5.1 Choisir les éléments de la solution à prototyper
			C4.1.5.2 Développer un prototype
			C4.1.5.3 Valider un prototype
	A4.1.6 Gestion d'environnements de développement et de test	Non	C4.1.6.1 Mettre en place et exploiter un environnement de développement
			C4.1.6.2 Mettre en place et exploiter un environnement de test
	A4.1.7 Développement, utilisation ou adaptation de composants logiciels	Non	C4.1.7.1 Développer les éléments d'une solution
			C4.1.7.2 Créer un composant logiciel
			C4.1.7.3 Analyser et modifier le code d'un composant logiciel
			C4.1.7.4 Utiliser des composants d'accès aux données
			C4.1.7.5 Mettre en place des éléments de sécurité liés à l'utilisation d'un composant logiciel
Vocabulaire à connaître			
Evaluation	Production d'écrit DST E4		

Objectif principal

- **Documenter** l'application Javastocks

Objectifs intermédiaires

- **Réinvestir** les notions de cours de java
- **Créer** un document technique de l'application Javastocks

Outils nécessaires

- Suite Microsoft ou Libre Office

Pré-requis

- Cours de base de données
- Cours sur Java

Votre mission

Créer une fiche de synthèse de l'application Aventures alpines selon les indications ci-dessous

Sommaire ou table des matières

1. **Automatiser** votre sommaire

Introduction

1. **Présenter**
 - 1.1. le contexte sur lequel vous travaillez (Description de Javastocks)

Déroulé de la mission

1. **Donner** votre schéma UML associé au contexte
2. **Préciser** les contraintes du cahier des charges
- 1- **Présenter** Node JS
 - a. Composition
 - b. version
 - c. Paramétrage avec copie écran
 - d. Emplacement physique sur le serveur (chemin, dossier)
- 2- **Montrer** à l'aide de copie écran la structure de votre application
- 3- **Expliquer** les fonctionnalités de votre programme

Analyse de l'activité

1. **Spécifier** les difficultés rencontrées au cours des différentes phases de mise en place
2. **Lister** les avantages de sa mise en place pour une entreprise
3. **Mentionner** les apports professionnels acquis à travers cette expérience
4. **Préciser** les apports personnels acquis à travers cette expérience