

Système de Gestion de Banque de Sang

Contexte :

Suite aux défis rencontrés par les centres de transfusion sanguine dans la gestion manuelle des donneurs et receveurs, l'établissement hospitalier souhaite développer une solution web moderne et efficace. Votre mission est de créer une application web monolithique JEE qui intègre la gestion complète des donneurs, des receveurs, et l'automatisation du processus de matching selon les compatibilités sanguines et les urgences médicales.

Pages Principales de l'Application :

Dans cette application on trouve trois pages principales :

- Page création : 2 Formulaires distincts donneur/receveur avec des validations automatiques
- Page liste donneurs : Tableau avec informations personnelles + receveurs associés + actions (modifier/supprimer) + d'autres critères que vous voyez peut être pertinent de les ajouter
- Page liste receveurs : Tableau avec informations personnelles + donneurs associés + actions (modifier/supprimer) + d'autres critères que vous voyez peut être pertinent de les ajouter

Relations Métier entre les Entités :

La compatibilité sanguine suit une matrice stricte : O- est donneur universel (compatible avec tous), AB+ est receveur universel (peut recevoir de tous), tandis que les autres groupes suivent des règles spécifiques. Voici la matrice sur laquelle il faut se baser

		DONNEUR							
		O-	O+	B-	B+	A-	A+	AB-	AB+
RECEVEUR	AB+	●	●	●	●	●	●	●	●
	AB-	●		●		●		●	
	A+	●	●			●	●		
	A-	●				●			
	B+	●	●	●	●				
	B-	●		●					
	O+	●	●						
	O-	●							

Gestion des Donneurs :

Un donneur possède un groupe sanguin, des informations personnelles (nom, prénom, téléphone, CIN, Date de naissance, Poids, Sexe), et un statut de disponibilité (DISPONIBLE, NON_DISPONIBLE, NON_ELIGIBLE). D'autres attributs à ajouter si vous voyez pertinents de les mettre

Un donneur est NON_ELIGIBLE s'il ne respecte pas les critères suivants :

- Âge entre 18 et 65 ans
- Poids minimum de 50 kg
- Absence de contre-indications médicales : hépatite B/C, VIH, diabète insulino-dépendant, grossesse, allaitement

Un donneur peut donner au max 1 poche de sang

Un donneur peut être associé à 1 seul receveur à la fois

Après association, le donneur devient NON_DISPONIBLE

Pour le moment le donneur ne redevient pas DISPONIBLE après un don (voir la partie bonus si vous ajoutez la possibilité de le rendre DISPONIBLE).

Le statut de disponibilité est déterminé automatiquement

Gestion des Receveurs :

Un receveur possède un groupe sanguin, des informations personnelles (nom, prénom, téléphone, CIN, Date de naissance, Sexe). D'autres attributs à ajouter si vous voyez pertinents de les mettre

Le besoin en donneurs dépend de sa situation médicale : CRITIQUE (a besoin de 4 poches de sangs), URGENT (3 poches), NORMAL (1 poche)

Lors de l'affichage de la liste des receveurs, trier automatiquement par ordre de priorité décroissant (CRITIQUE → URGENT → NORMAL)

1 don = 1 poche MAXIMUM PAR RECEVEUR

L'état actuel d'un receveur est précisé automatiquement

L'état actuel d'un receveur est par défaut EN_ATTENTE et devient SATISFAIT quand il a le nombre requis de donneurs associés

Un receveur peut avoir plusieurs donneurs selon ses besoins

Un receveur SATISFAIT ne peut plus recevoir de nouveaux donneurs

Règles de Validation :

Lors de l'association, n'afficher que les entités compatibles :

- Pour un receveur : donneurs compatibles ET disponibles
- Pour un donneur : receveurs compatibles ET non encore satisfaits

Vous pouvez ajouter d'autres règles de validation que vous voyez pertinentes

Exigences Techniques

Stack Technologique Obligatoire :

- Langage : Java 8 ou version supérieure
- IDE : Au choix (Eclipse, IntelliJ IDEA, NetBeans)
- Vue : JSP avec JSTL pour les pages dynamiques
- Build : Maven pour la gestion des dépendances et le packaging
- Serveur : Apache Tomcat comme conteneur web
- Base de données : MySQL/PostgreSQL (au choix)
- Fonctionnalités Java : Collections API, Java Stream API, Optional, ... etc
- Tests Obligatoires : Minimum 2 tests unitaires avec JUnit
- Design UI : CSS natif ou framework (Bootstrap, Tailwind ...)
- Gestion des erreurs : message d'erreurs à afficher en cas d'erreur de validation ou autres

ORM via une bonne Configuration JPA/Hibernate :

- Présence du fichier persistence.xml dans src/main/resources/META-INF
- Configurer l'unité de persistance avec Hibernate
- Utiliser les annotations JPA pour mapper les entités (@Entity, @Id, @OneToMany, @ManyToOne, etc.)
- Utiliser EntityManagerFactory et EntityManager pour gérer la persistance
- **INTERDICTION d'utiliser application.properties (spécifique à Spring Boot)**

Architecture et Design Patterns

Implémenter une architecture MVC multicouches comprenant :

- Couche Présentation : JSP/JSTL pour les vues
- Couche Contrôleur : Servlets pour gérer les requêtes HTTP
- Couche Service : Logique métier et règles de compatibilité
- Couche DAO/Repository : Accès aux données avec JPA
- Autres si vous voulez

Design Patterns et principes à respecter :

- Repository Pattern / Singleton Pattern
- SOLID
- Eviter les antipatterns
- Logging : Implémenter un système de logs en cas de besoin

Configuration et Contraintes :

- Configuration obligatoire des path url via "web.xml" (mapping des servlets, filtres, listeners)
- **INTERDICTION d'utiliser les annotations de mapping (@WebServlet, @WebFilter)**

Gestion de Projet :

- Utiliser JIRA avec méthodologie Scrum
- Créer les user stories, sprints et backlogs ... etc

Bonus (à faire si vous voulez)

- 2 tests d'intégration à ajouter
- Ajouter la fonctionnalité permettant à un donneur de redevenir disponible automatiquement 1 mois après son dernier don
- Ajouter la Pagination : Pour les listes de donneurs/receveurs
- Ajouter la barre de recherche dynamique
- Ajouter le filtrage lors d'affichage de la liste des receveurs/donneurs
- Dockerisation : Dockerfile pour conteneuriser l'application
- Déploiement WAR : Déployer manuellement sur Tomcat sans IDE
- Dashboard Analytics : Une nouvelle page dédiée avec graphiques Chart.js affichant : Taux de donneurs par groupe sanguin, Répartition receveurs par urgence, Autres charts de votre choix

Livrables Attendus

- Code source complet avec structure Maven
- Fichier WAR déployable
- README.md
- Lien jira
- Diagramme de classe

Modalités pédagogiques :

- Projet individuel
- Début : 06/10/25, Deadline d'envoi des rendus : 14/10/25 avant minuit

Modalités d'évaluation :

- Présentation client et technique avec démonstration (strictement en français)
- Discussions avec des questions
- Mise en situation

Critères de performance :

- Respect des exigences fonctionnelles et règles métier
- Qualité de l'architecture et du code
- Maîtrise des technologies JEE
- Gestion de projet et livrables
- Qualité de la présentation et démonstration