

Matière: Projet L3

Thème 01	
Intitulé	Outil d'aide à l'adaptation posologique de médicaments
Domaine	Programmation mobile
Mots clés	Programmation mobile, Android, Bases de données
Objectifs et démarche	L'objectif de ce projet est la conception et le développement d'une application mobile qui d'aide à l'adaptation posologique de médicaments utilisant UML pour la conception, Java et Fxml pour le développement des interfaces et SQLite pour la gestion de base de données.
Outils et méthodes	SQLite, StartUML, JAVA, Android Studio, Flutter, vsCode.
Fonctionnalités attendues	<p>Pour administrer certains types de médicaments, le médecin demande aux patients de faire des bilans médicaux (analyses au laboratoire) parmi les trois paramètres : la clairance rénale, la bilirubine, la tgo/tgp (selon le médicament).</p> <p>Selon les valeurs de ces paramètres, le médecin décidera de :</p> <ul style="list-style-type: none"> - Administre la dose complète du médicament, si les bilans sont normaux - Ne pas administrer le médicament parce qu'il est contre-indiqué dans le cas du patient - Adapter la dose selon les valeurs des bilans. <p>Exemples de médicaments : (les valeurs suivantes ne sont pas toutes correctes mais sont prises pour exemple)</p> <ol style="list-style-type: none"> Médicament : cisplatine <ul style="list-style-type: none"> - Bilan demandé : clairance rénale - Dose à administrer : <ul style="list-style-type: none"> o si la clairance rénale ≤ 30 ml/min → la cisplatine est contre-indiquée o si la clairance rénale ≥ 60 ml/min → on administre la dose complète (la valeur de la dose complète est connue par les médecins, on n'a pas à la préciser) o si $30 \text{ ml/min} < \text{la clairance rénale} < 60 \text{ ml/min}$ → on administre 50% de la dose complète Médicament : acide zolidronique <ul style="list-style-type: none"> - Bilan demandé : clairance rénale - Dose à administrer : <ul style="list-style-type: none"> o si la clairance rénale ≤ 30 ml/min → l'acide zolidronique est contre-indiqué o si la clairance rénale ≥ 60 ml/min → on administre la dose complète o si $30 \text{ ml/min} < \text{la clairance rénale} < 40 \text{ ml/min}$ → on administre 3 mg d'acide zolidronique o si $40 \text{ ml/min} < \text{la clairance rénale} < 50 \text{ ml/min}$ → on administre 3.3 mg d'acide zolidronique o si $50 \text{ ml/min} < \text{la clairance rénale} < 60 \text{ ml/min}$ → on administre 3.5 mg d'acide zolidronique Médicament : capecitabine <ul style="list-style-type: none"> - Bilan demandé : clairance rénale, bilirubine, tgo/tgp - Dose à administrer : <ul style="list-style-type: none"> o si la bilirubine ≥ 60 ml/min → la capecitabine est contre-indiquée o si la bilirubine < 60 ml/min → on administre la dose complète o si tgo/tgp ≥ 55 ml/min → la capecitabine est contre-indiquée o si tgo/tgp < 55 ml/min → on administre la dose complète o si $30 \text{ ml/min} < \text{la clairance rénale} < 50 \text{ ml/min}$ → on administre 25% de

	<p>la dose complète</p> <ul style="list-style-type: none"> ○ si la clairance rénale ≥ 60 ml/min → on administre la dose complète ○ si la clairance rénale ≤ 30 ml/min → la capecitabine est contre-indiquée <p>Exemples de cas :</p> <ol style="list-style-type: none"> Médicament à administrer : cisplatine <ul style="list-style-type: none"> - Bilan réalisé : clairance rénale = 25 ml/min → le médicament est contre-indiqué - Bilan réalisé : clairance rénale = 80 ml/min → administrer la dose complète - Bilan réalisé : clairance rénale = 55 ml/min → administrer 50% de la dose complète Médicament : acide zolidronique <ul style="list-style-type: none"> - Bilan réalisé : clairance rénale = 25 ml/min → le médicament est contre-indiqué - Bilan réalisé : clairance rénale = 80 ml/min → administrer la dose complète - Bilan réalisé : clairance rénale = 55 ml/min → administrer 3.5 mg d'acide zolidronique Médicament : capecitabine <ul style="list-style-type: none"> - Bilan réalisé : clairance rénale + bilirubine + tgo/tgp <p><u>Cas 1 :</u></p> <ul style="list-style-type: none"> - clairance rénale = 25 ml/min → contre-indication - bilirubine = 70 ml/min → contre-indication - tgo/tgp = 45 ml/min → réduction de 25% <p>→ le médicament est contre-indiqué</p> <p><u>Cas 2 :</u></p> <ul style="list-style-type: none"> - clairance rénale = 75 ml/min → dose complète - bilirubine = 40 ml/min → dose complète - tgo/tgp = 45 ml/min → réduction de 25% <p>→ la dose à administrer est 25% de la dose complète</p> <p>L'application doit avoir :</p> <ul style="list-style-type: none"> ✓ Gestion de la BD de médicaments (ajout, recherche) ✓ Sélection d'un médicament dans la base des médicaments, puis déduire la dose à administrer, suivant le bilan du malade.
Étapes à suivre	<ol style="list-style-type: none"> 1. Dessiner les maquettes des interfaces. 2. Entamer la partie conception par UML. <ul style="list-style-type: none"> ○ Construire un diagramme des cas d'utilisation. ○ Continuer les diagrammes des séquences. ○ Construire le diagramme de classes. 3. Entamer la programmation.

Thème 02	
Intitulé	Réalisation d'une application de suivi des projets de fin d'étude licence.
Domaine	Programmation JAVA, Système d'information, Bases de données.
Mots clés	SQL, UML, JAVA.
Objectifs et démarche	L'objectif de ce projet est la conception et le développement d'une application de gestion des projets de fin d'étude licence en utilisant UML pour la conception, Java pour le développement des interfaces et MySQL pour la gestion de base de données.
Outils et méthodes	MySQL, StartUML, JAVA, NetBeans ou Eclipse.
Fonctionnalités attendues	<p>L'application doit avoir :</p> <ul style="list-style-type: none"> ✓ Gestion des groupes : ajout, suppression et modification. ✓ Gestion des sujets : ajout, suppression et modification. <ul style="list-style-type: none"> - Un sujet est connu par son titre, sa description, son domaine, ses mots clés, ses outils de développement, ... ✓ Gestion des étudiants : ajout, suppression et modification. <ul style="list-style-type: none"> - Un étudiant est connu par un numéro d'inscription, le nom, le prénom, la date de naissance. - Un étudiant appartient à un seule groupe et un seul binôme. ✓ Gestion des séances : ajout, suppression et modification. <ul style="list-style-type: none"> - Une séance est connue par sa date, son horaire, la salle et le groupe. ✓ Suivi des binômes: <ul style="list-style-type: none"> - Création des binômes : un binôme est appartient à un seul groupe et contient au maximum deux étudiants. - Affectation des sujets : un binôme ne peut avoir qu'un seul sujet. ✓ Suivi des étudiants : <ul style="list-style-type: none"> - Evaluation continue de chaque binôme par séance à savoir : <ul style="list-style-type: none"> ▪ L'état d'avancement de rapport ▪ L'état d'avancement de l'application ▪ Une note pour chaque séance. ▪ Une remarque pour chaque séance. - Evaluation de rapport final et de l'application finale de chaque binôme. <p>Etablir des statistiques semestrielles pour chaque étudiant, chaque sujet et chaque groupe.</p>
Étapes à suivre	<ol style="list-style-type: none"> 1. Introduire le sujet par une introduction générale. 2. Construire un cahier des charges. 3. Entamer la partie conception par UML. <ul style="list-style-type: none"> ○ Construire un diagramme des cas d'utilisation. ○ Continuer les diagrammes des séquences. ○ Construire le diagramme de classes. 4. Entamer la programmation.

Thème 03	
Intitulé	Gestion de conférence
Domaine	Développement Web, Bases de données.
Mots clés	Développement Web, SQL, UML, JavaScript, Laravel.
Objectifs et démarche	<p>L'objectif de ce projet est la conception et le développement d'une application de gestion de conférence en utilisant UML pour la conception. Cette application permet de maîtriser les différentes étapes du développement d'applications web en utilisant le Framework Laravel qui combine les meilleures pratiques du domaine :</p> <ul style="list-style-type: none"> - Programmation objet PHP et design patterns - Architecture MVC - Intégration de services authentification, génération des URLs, etc.
Outils et méthodes	Laravel, Bootstrap, Editeur code PHP, Outil d'administration de bases de données.
Fonctionnalités attendues	<ul style="list-style-type: none"> - Fournir une plate-forme en ligne et interactive pour les chercheurs et les organisateurs de la conférence. - Un système de gestion de conférence, est une application en ligne destinée à la gestion complète d'une conférence, à savoir: <ul style="list-style-type: none"> o La création du site web officiel, o Réception des articles o L'évaluation des soumissions, o L'inscription des candidats, o L'indexation et le partage des résultats des soumissions - Authentification et Autorisations des utilisateurs : <ul style="list-style-type: none"> o Connexion sécurisée aux organisateurs et chercheurs pour visualiser et manipuler leurs ressources. (articles) - Administration : <ul style="list-style-type: none"> o Gestion des comptes organisateurs et chercheurs des inscriptions
Étapes à suivre	<ol style="list-style-type: none"> 1. Préparation de l'environnement <ul style="list-style-type: none"> ✓ Installation Laravel ✓ Installation Homestead ou Valet ✓ Configurations de l'environnement 2. Etude du système "E-conference" <ul style="list-style-type: none"> ✓ Expression des besoins fonctionnels et non-fonctionnels ✓ Analyse et Conception UML <ul style="list-style-type: none"> o Construire un diagramme des cas d'utilisation. o Continuer les diagrammes des séquences. o Construire le diagramme de classes. 3. Création de l'application "E-conference" <ul style="list-style-type: none"> ✓ Développement des modèles et migrations, des vues et des contrôleurs ✓ Mise en forme des vues avec Bootstrap ✓ Génération des migrations ✓ Tests unitaires ✓ Production et Déploiement. 4. Préparation d'un rapport détaillé couvrant les points suivants : <ul style="list-style-type: none"> ✓ Introduction du Framework Laravel (avantages, caractéristiques, installation, etc.) ✓ Les détails du système "E-conference" (description textuelle et modèles UML) ✓ Les détails d'implémentation et les résultats obtenus ✓ Conclusion (les principaux constats)

Thème 04	
Intitulé	Développement d'un éditeur pour les réseaux PERT.
Domaine	Programmation JAVA, Gestion de projets.
Mots clés	UML, JAVA.
Objectifs et démarche	<p>L'objectif de ce projet est la conception et le développement d'une d'un éditeur pour les réseaux PERT (Program Evaluation Review Technique) en utilisant UML pour la conception, Java pour le développement des interfaces.</p> <p>Un réseau PERT respecte les caractéristiques suivantes :</p> <ul style="list-style-type: none"> ✓ Un graph no cyclique. ✓ Une tâche est représentée par un arc. ✓ Une étape (ou jalon) est représenté par un cercle. ✓ Une seule étape initiale (étape qui n'a pas de entrées). ✓ Une seule étape finale (étape qui n'a pas de sorties). ✓ Une étape est connue par le nom, la date au plus tôt et la date au plus tard. ✓ Une tâche est connue par le nom, la durée, la marge totale et la marge libre. ✓ La tâche fictive est sans nom et de durée zéro. ✓ Un chemin est la succession de tâches de l'étape initiale jusqu'à l'étape finale ✓ Un ou plusieurs chemins critiques (un chemin est dite critique si la somme des durées de ses tâches le maximum). ✓ Une tâche (ou étape) est dite critique si elle appartient à un chemin critique. ✓ La date au plus tôt d'une étape = max (date au plus tôt de l'étape précédente + durée de la tache comprise entre les 2 étapes). La date au plus tôt de l'étape initiale = 0. ✓ La date au plus tard d'une étape = min (date au plus tard de l'étape suivante - durée de la tache comprise entre les 2 étapes). La date au plus tard de l'étape finale = la date au plus tôt de l'étape finale. ✓ La date au plus tôt d'une étape ≤ la date au plus tard de cette étape. ✓ La marge totale d'une tâche = la date au plus tard d'une étape de fin de tache - (la durée + la date au plus tôt d'une étape de début de tache). ✓ La marge libre d'une tâche = la date au plus tôt d'une étape de fin de tache - (la durée + la date au plus tôt d'une étape de début de tache). ✓ La marge totale d'une tâche ≤ la marge libre de cette tâche. ✓ Si la marge totale d'une tâche = 0 → la tâche est critique.
Outils et méthodes	StartUML, JAVA, NetBeans ou Eclipse.

Fonctionnalités attendues	<p>L'application doit avoir :</p> <ul style="list-style-type: none"> - Une barre de menus. - Une ou plusieurs barres d'outils. - Une palette d'objets pour le dessin de réseau PERT. - Des raccourcis pour le calcul des dates et des marges. - Une zone de dessin sous forme de fenêtres internes.
Étapes à suivre	<ol style="list-style-type: none"> 1. Introduire le sujet par une introduction générale. 2. Construire un cahier des charges. 3. Entamer la partie conception par UML. <ul style="list-style-type: none"> ○ Construire un diagramme des cas d'utilisation. ○ Continuer les diagrammes des séquences. ○ Construire le diagramme de classes. 4. Entamer la programmation.