L3-New Novembre 2019

Projet de Bases de données-2

Application de gestion de RDV (Karim LAHLOU – Safia OUERDANE)

Instructions générales

Ce programme de gestion de rendez-vous devra être réalisé entièrement en Java ou tout autre langage en accord avec votre chargé de TP (C#, PHP, Python, etc…) et la base de données doit être obligatoirement Oracle.

Vous pouvez vous aider :

- ✓ Du support de cours (incontournable)
- ✓ Des ressources disponibles sur Internet, notamment sur le site d'Oracle
- ✓ De l'encadrement en TP/Projet.

Modules, API et concepts concernés par ce programme (à titre d'information)

- Concepts généraux de Java (héritage, polymorphisme, encapsulation, surcharge, méthodes...)
- Constructeurs, interfaces, classes abstraites,
- Collections
- Threads
- JDBC et Gestion de la persistance par ORM (*Object-Relational Mapping*)
- Interfaces graphiques et Swing

Critères d'évaluation

- Ergonomie (vous avez carte blanche sur le choix du *'look and feel'* de l'IHM). Mais votre interface doit être visuellement agréable, "*user friendly*" et intuitive en fenêtre graphique ou en console en lignes de commande.
- Qualité du code (commentaires, algorithmes utilisés)
- Respect du pattern MVC souhaité
- Gestion des exceptions
- Gestion optimale, flexible et simple de la persistance
- Richesse des concepts mis en œuvre et des API utilisées.
- Respect du cahier des charges

Etapes de mise en œuvre

1) Avant tout, imprégnez-vous des exigences du cahier des charges
2) Générez des maquettes de l'IHM. Pour ce, vous pouvez par exemple utiliser au choix l'un des sites suivants :
   https://gomockingbird.com/mockingbird/
3) Rédigez un mini document de conception technique avec le dictionnaire de données, le MCD, MLD et/ou MPD, un diagramme de classes souhaité pour les classes Java voir aussi un diagramme de cas d'utilisation et/ou séquences élémentaire. Le but est d'illustrer le choix de vos entités amenées à être implémentées sous forme de classes et les interactions entre ces entités. Pensez à utiliser des AGLS comme JMerise, StarUML, ….

Cahier des charges

I – Introduction

Cette application est destinée à ma psychologue qui souhaite gérer ses rendez-vous avec ses patients.
Il permettra deux types d'actions :

- ✓ La psychologue enregistre un rendez-vous de consultation pour un patient.
- ✓ Le patient consulte ses rendez-vous passés et futurs.

La psychologue et les patients doivent être authentifiés pour se connecter à l'application.

II – Contraintes techniques

- ✓ N'oubliez pas de créer votre package spécifique (ne pas utiliser, par exemple, le package par défaut de NetBeans ou Eclipse).
- ✓ Lors de la fermeture du programme, un message de confirmation doit être proposé à l'utilisateur.
- ✓ Un couplage faible avec le SGBD est impératif. Ceci permettra, entre autres, d'interchanger de SGBD avec un minimum de codage et de paramétrage.
- ✓ Pour la gestion de la persistance, l'utilisation de Framework Hibernate ou JPA est autorisée. Toutefois nous attirons votre attention sur le point concernant le couplage faible.
- ✓ Toutes les ressources ouvertes doivent être libérées au pire à la fermeture du programme.

III – Description de l'application Gestion des rendez-vous de consultation

Ma psy oublie tout : l'heure de mon rendez-vous, mon deuxième prénom, même la date de ma prochaine consultation ! En cherchant sans ses lunettes dans ses fiches manuscrites, elle a confondu ma fiche avec un autre patient.
Pour sa fête, elle a reçu un ordinateur d'un patient passionné. Connaissant ma névrose du rangement et de l'organisation des données, elle m'a demandé de l'aider à mettre un peu d'ordre dans ses fiches patients.
Voici les données à conserver :

- *Qui est venu la voir et à quelle heure. La classification qui l'intéresse est la suivante : enfant, ado, femme, homme ou couple. Notez que les séances d'un couple peuvent se poursuivre individuellement et qu'un patient ado peut revenir quelques années plus tard en tant qu'adulte.*
- *Par quel moyen (autre patient, docteur, bouche à oreille, Pages Jaunes, Internet, etc.) le patient l'a-t-elle connue (elle est curieuse !)*
- *La date de la première consultation et si le patient est en retard à son rendez-vous.*
- *La date et l'heure, le prix et le mode de règlement de la séance de consultation.*
- *Pour les consultations pour anxiété, l'indicateur de 0 à 10 que le patient s'attribue (enfin s'il peut parler encore ...)*
- *Certains critères (La profession actuelle du patient, voir la liste de ses professions antérieures, etc.)*
- *Certains mots-clés (expressions utilisées à l'oral par le patient), d'éventuelles postures affichées et certains comportements observés au cours de la séance notés après le rendez-vous.*
- *Notez que la psy prend au maximum 3 patients pour une même consultation (de toute manière, il n'y a que deux sièges et un fauteuil !)*

Ma psy ne fait pas un boulot facile, alors ne ratez pas sa base de données !

Toutes les entités recensées doivent avoir des identifiants sous forme d'un entier et de préférence auto-incrémenté. Certains champs doivent être obligatoires comme le nom et le prénom du patient et d'autres optionnels comme leurs professions actuelles ou antérieures. Certains champs sont modifiables dans les formulaires comme la classification ou l'adresse du patient et d'autres non modifiables comme le nom et le prénom du patient.

IV – Fonctionnalités de base de la Gestion des RDVs

Attention à bien gérer les différents rendez-vous de chaque patient !

- Inscription d'un nouveau patient par la psy.
- Ajout, annulation ou modification d'un rendez-vous d'un patient par la psy.
- Consultation des rendez-vous sur une journée donnée ou une semaine donnée par la psy.
- Consultation des rendez-vous antérieurs et futurs d'un patient par le patient ou la psy.
- Login avec authentification par email et mot de passe pour le patient et admin et mot de passe pour la psy.
- Les rendez-vous peuvent être pris du lundi au samedi de 8h00 à 20h00 en créneaux de 30 minutes avec les contraintes suivantes :
    - Un patient ne peut pas avoir plus de 3 rendez-vous par jour.
    - La psy ne peut pas travailler plus de 10 heures par jour.
    - Un rendez-vous ne peut concerner que 3 personnes à la fois au maximum.

Ce défi rapportera jusqu'à **2 pts sur la note finale** aux membres du groupe
Seule condition : **Toutes** les fonctionnalités de base doivent être mises en œuvre.

- ✓ Possibilité à un patient de modifier ses rendez-vous selon la disponibilité de la psy. Attention à la gestion de la concurrence d'accès à la base de données.

V – Fonctionnalité bonus

Génération de fichier des RDVs d'une journée ou d'une semaine à préciser en format pdf ou json ou ical (Ceci peut rapporter un bonus jusqu'à **2 pts sur la note finale** aux membres du groupe).

Consignes d'organisation

- S'organiser en binômes (Pas de monôme ni de trinôme sauf si groupe de TD impair et remettre la liste des groupes à la fin de la séance de lancement du projet)
- Utiliser un AGL (Atelier de Génie Logiciel) pour élaborer votre schéma conceptuel de données (MCD) et générer le schéma logique puis le schéma physique de votre base de données ainsi que le code SQL pour la création de votre base de données (JMerise ou AnalyseSi par exemple).
- Intégrer à votre script toutes les contraintes d'intégrité nécessaires pour le bon fonctionnement de votre application.
- Proposer un jeu d'essai en remplissant quelques tables pour valider votre bonne conception.
- Créer et testez votre base de données.
- Développez votre application en utilisant un EDI (Environnement de Développement Intégré) en testant d'abord la communication avec votre base de données puis en implémentant les cas d'utilisation que vous avez choisis. Vous pouvez utiliser aussi un AGL comme StarUML pour vous faciliter la conception de votre application.
- Pensez à définir des priorités pour garantir la livraison de votre projet avec un minimum de fonctionnalités dans les délais.

- Séance de validation du travail fini à la dernière séance de Tp/Projet durant la semaine du 16/12/2019.
- Un rapport décrivant votre projet en intégrant le dictionnaire de données, MCD, MLD ou MPD et voir aussi le diagramme de classes et/ou cas d'utilisation ainsi qu'un scenario d'exécution avec des prises d'écran et en mettant en évidence les fonctionnalités implémentées, celles envisagées et non terminées et aussi les difficultés rencontrées et les points forts et faibles du projet.
- Le rendu exclusivement sur campus doit se faire avant le dimanche 22/12/2019 à 23h55.

Rendus attendus

1) Au plus tard, le **22/12/2019** à **23h55**, vous devez déposer vos livrables dans un fichier zip, sur Campus (Projet source Java ou autre langage + scripts SQL + Rapport).
2) Tous les fichiers sources, scripts SQL et documents doivent contenir les noms des membres du groupe en en-tête.
3) Les livrables seront importés sur votre projet dans Eclipse. Objectif "zéro erreur" à ce niveau.
4) La documentation comptera pour 25% de la note du projet et sera évaluée minutieusement avant même le lancement de votre programme.
5) Au départ, chaque groupe disposera d'un capital maximum de 20 pts. Au fur et à mesure que des erreurs, des "oublis" ou des imperfections seront rencontrées, un ou des points seront décomptés.