

Projet Java ING3 2018

Gestion informatique d'un centre hospitalier



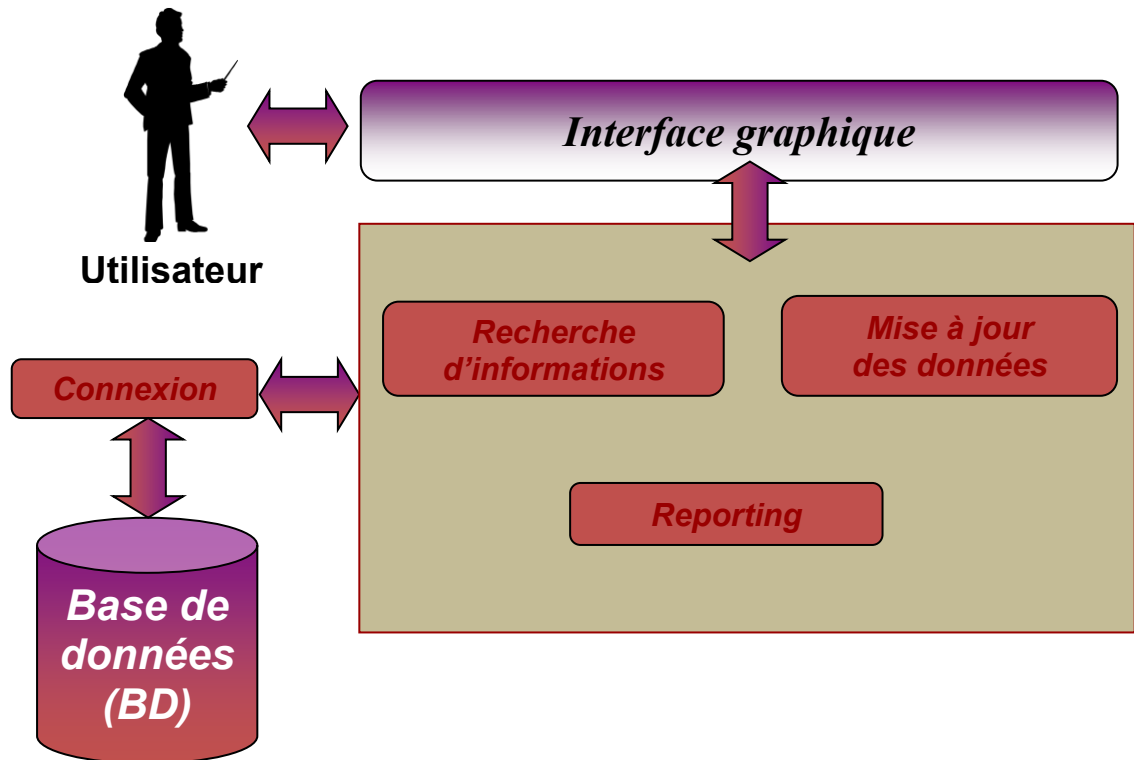
Sommaire

Introduction	3
Objectif.....	3
Architecture générale du système	3
Description de la base de données	4
Ressources à télécharger de campus.....	5
Création et importation de la base.....	5
Mise à jour des données via le système.....	5
Interrogation de la base de données via le système.....	5
L'interface graphique	5
Le module de Reporting avec JFreeChart	5
Enrichissement de la base existante.....	5
Travail à faire et planning.....	6
Livraison à déposer sur campus : PowerPoint et code Java	6
Déroulement de la soutenance	6
Critères d'évaluation du travail à réaliser	7
Recommandations	7
Annexe : exemples de requêtes.....	8

Introduction

Objectif

Ce projet vise à concevoir un système d'aide à la décision pour le personnel hospitalier. L'architecture de ce système est décrite dans le schéma ci-dessous :



Architecture générale du système

Dans cette section, nous présentons l'Architecture générale du système de gestion informatique d'un centre hospitalier. Ce système compte principalement 5 modules :

- Le module de **Recherche d'informations**
- Le module de **Mise à jour des données**
- Le module de **Reporting**
- Le module **Connexion** communique entre la BD et les 3 modules précédents
- L'**Interface graphique** communique avec les 3 premiers modules

L'interface graphique permet de mettre à jour (ajout, suppression, modification) toutes les données utilisées pour la gestion informatique du centre hospitalier, d'interagir avec les 3 autres modules, et d'afficher au fur et à mesure les résultats de chaque module du processus de gestion. Tous les modules communiquent via la base de données. Les résultats produits sont également stockés dans cette dernière.

L'implémentation du prototype de gestion informatique du centre hospitalier va être réalisée en Java (JDK 1.8) et la base de données décrite ci-dessous est stockée sous MySQL.

Description de la base de données

Dans cette section, nous vous donnons ci-dessous la description de la base de données existante et qui sera exploitée par le système de gestion de cet hôpital. Il s'agit d'une base décrivant de façon – très simplifiée – les données d'un hôpital :

SERVICE (**CODE**, NOM, BATIMENT, DIRECTEUR)
CHAMBRE (**NO_CHAMBRE**, **#CODE_SERVICE**, SURVEILLANT, NB_LITS)
EMPLOYE (**NUMERO**, *NOM*, *PRENOM*, *TEL*, ADRESSE)
DOCTEUR (**#NUMERO**, SPECIALITE)
INFIRMIER (**#NUMERO**, #CODE_SERVICE, ROTATION, SALAIRE)
MALADE (**NUMERO**, *NOM*, *PRENOM*, *TEL*, ADRESSE, MUTUELLE)
HOSPITALISATION (**NO_MALADE**, **#CODE_SERVICE**, **#NO_CHAMBRE**, **LIT**)
SOIGNE (**#NO_DOCTEUR**, **#NO_MALADE**)

Remarques :

- Les clés primaires sont en gras et soulignées
- Les clés candidates importantes sont en italique et soulignées
- Les clés secondaires sont précédées par #
- Etant dans un contexte hétérogène, deux concepts identiques peuvent ne pas avoir la même terminologie (exemple : la clé primaire CODE dans la table SERVICE est équivalent à la clé secondaire CODE_SERVICE dans les tables CHAMBRE, INFIRMIER et HOSPITALISATION)
- La clé primaire NUMERO des tables DOCTEUR et INFIRMIER est aussi clé secondaire référençant la clé primaire NUMERO de la table EMPLOYE

En définissant le schéma, les hypothèses suivantes ont été faites :

- Un service se trouve logé dans un seul bâtiment, mais plusieurs services peuvent partager un bâtiment
- Le directeur d'un service est un docteur identifié par son NUMERO
- Le numéro de CHAMBRE est local à un bâtiment (i.e. chaque bâtiment possède une CHAMBRE numéro 1)
- Une chambre n'identifie pas directement le bâtiment qui la contient, mais indirectement par le code de son service
- Un(e) surveillant(e) est un(e) infirmier(ère) identifié(e) par son NUMERO
- Les informations communes à tous les employés sont dans la relation EMPLOYE qui contient donc docteurs, infirmiers(ères) ainsi que d'autres employés
- Le triplet (NOM, PRENOM, TEL) est considéré comme une clé de la relation EMPLOYE
- Docteurs et infirmiers(ères) sont donc des employés identifiés par un unique numéro
- La relation DOCTEUR comporte les attributs spécifiques des docteurs
- La spécialité des docteurs est définie sur une liste de valeurs permises, 'Anesthésiste', 'Cardiologue', 'Generaliste', 'Orthopédiste', ...
- La relation INFIRMIER comporte les attributs spécifiques des infirmiers(ères)
- La rotation (pour simplifier) prend les valeurs 'JOUR' et 'NUIT'
- Un(e) infirmier(ère) est affecté(e) à un service
- Les docteurs ne sont pas affectés à un service particulier mais sont souvent amenés à exercer de façon transversale
- Le concepteur de la base de données a choisi de maintenir les malades dans une relation séparée qui comporte tous les attributs nécessaires
- Le triplet (NOM, PRENOM, TEL) est également une clé de la relation MALADE
- La relation HOSPITALISATION ne concerne que les malades hospitalisés à l'état courant de la BD

Ressources à télécharger de campus

Le fichier [Ressources Projet Java.zip](#), à télécharger de la page campus du « Projet Java », contient les fichiers suivants :

- Le fichier **hopital.sql** : script SQL de la base de données à créer et importer en local sur wamp et à distance dans le serveur de l'ECE.
- Le fichier source **SSHTunnel.java** : sa classe *SSHTunnel* établit une connexion au serveur de la base de données (BDD) via un tunnel SSH
- Le fichier source **Connexion.java** : sa classe *Connexion* permet de se connecter, grâce à ses 2 constructeurs, à la BDD locale (*localhost*) et distante du serveur de l'ECE <https://sql.ece.fr/> via le tunnel SSH. Elle récupère ou met à jour les données à la demande des modules de Recherche, de Mise à jour et de Génération et Historisation des Activités. Ses méthodes permettent chacune d'exécuter une requête SQL de type *String* passée en paramètre.
- Le fichier **SSH.pdf** : la procédure à suivre sur la connexion SSH vers le serveur de base de données de l'ECE.
- Le dossier **javadoc** : la documentation en ligne des 2 classes *SSHTunnel* et *Connexion*, en cliquant sur **index.html**.
- le dossier **lib** : les 2 librairies **jsch-0.1.51.jar** (pour la connexion SSH) et **mysql-connector-java-5.1.18-bin.jar** (pour l'accès au serveur mysql de la base données), à intégrer dans votre projet sur NetBeans pour son bon fonctionnement.

Création et importation de la base

Pour créer la base et importer ses tables à partir du fichier **hopital.sql** téléchargé : consulter le [Guide SGBD MySQL](#) sur cette même page campus.

Mise à jour des données via le système

Ce module permet de mettre à jour (modifier, insérer, supprimer) les données de la BDD via l'interface graphique de votre système.

Interrogation de la base de données via le système

Vous trouverez des exemples non exhaustifs de requêtes de recherche d'informations en **ANNEXE** de ce document

L'interface graphique

Votre interface graphique doit contenir un menu avec des items. Chaque item est associé à un module de votre architecture globale. Par exemple, pour mettre à jour les données d'une table particulière, il faut tout d'abord sélectionner votre item de mise à jour qui va ouvrir une fenêtre de mise à jour et mettre à jour la table.

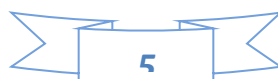
Votre interface graphique constitue la *Vue* dépendante du *Contrôleur* (modules de Recherche, de Mise à jour et de Génération) qui interroge les données (*Modèle*) de la base, via le module de Connexion, pour transmettre ces données à l'interface graphique qui les affiche. Votre interface graphique sera événementielle (touches clavier ou/et clics souris). Une fenêtre d'accueil permettra **vous connecter à la BDD localement ou à distance** en saisissant le nom de la base, votre login et votre password. Votre interface graphique affichera de manière ergonomique, claire et fluide toutes les informations pertinentes de la BDD exploitées par le système.

Le module de Reporting avec JFreeChart

Ce module permet de générer au fur et à mesure des statistiques (camemberts, histogrammes...), comme par exemple le nombre d'hospitalisés par type de service, les maladies les plus contractées... : consulter [The JFreeChart Class Library](#)

Enrichissement de la base existante

Afin de pouvoir répondre à d'autres requêtes telles que la gestion des rendez-vous... il serait nécessaire de revisiter la structure de la base existante et donc d'ajouter les tables et attributs pertinents. A vous d'imaginer les requêtes utiles.



Travail à faire et planning**Livrable à déposer sur campus : PowerPoint et code Java**

Le livrable sera un fichier zip contenant les comptes rendus suivants :

1) Un PowerPoint avec les slides suivants (environ 10 slides) :

- Page de garde avec titre, noms des coéquipiers et groupe de TD
- Sommaire
- **Diagramme de classes** avec l'outil Draw.io ou équivalent présentant les attributs, les méthodes et les cardinalités, sans constructeurs ni getters/setters.
- Conception technique présentant la modularité avec Entrées et Sorties, selon l'approche **Modèle-Vue-Contrôleur**
- **Maquettes de votre interface** graphique en commentant les composants graphiques Swing (ou autre librairie Java) utilisés et leur mise en page (layouts).
- **Versioning GIT** : screenshot et lien avec login et passwd montrant la bonne utilisation et répartition des tâches du code entre coéquipiers
- **Bilan individuel et collectif** de l'état du travail effectué.
- **Bibliographie de toutes vos sources** (web, livres etc.). **Toute source non citée est considérée de facto comme un plagiat.**

2) Le code Java :

Tous les dossiers et fichiers du projet développé sur Eclipse ou Netbeans avec les sources, l'exécutable **.jar** en mode graphique et la documentation **Javadoc** avec commentaires sur les classes et les méthodes.

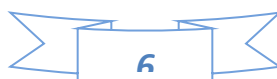
Votre livrable sera à déposer sur campus le **dimanche 22 avril 2018 à 23h55 dernier délai** en respect des consignes ci-dessus dans la rubrique [Livrable à déposer](#).

Déroulement de la soutenance

Les soutenances auront lieu la semaine du 23 avril 2018. La soutenance sera de 15 minutes maximum par équipe. Elle permet au client (le prof ;-)) de tester votre programme et de vérifier l'adéquation de la solution que vous proposez avec le cahier des charges qui vous a été remis et en adéquation avec la version graphique du code déposé sur campus. Elle devra permettre (en très peu de temps pour le jury, faites attention ! C'est frustrant) de juger votre argumentation, l'esthétique (est-ce beau ?), la clarté (pertinence et précision des informations affichées ?), l'ergonomie (facile d'exécuter une action ?), l'originalité... de votre projet.

Chaque équipe présentera son projet par vidéoprojecteur devant les autres équipes de son groupe de TD. Pour preuve du bon fonctionnement de votre système, l'équipe exécutera son archive **jar** en mode graphique livré sur campus. Elle fera la démonstration, en connexion locale et distante, que leur exécutable respecte les spécifications du sujet. L'équipe présentera ensuite brièvement la **javadoc** et le **PowerPoint** livrés sur campus.

Chaque étudiant d'une équipe doit intervenir lors de la présentation orale et avoir une connaissance globale du projet de l'équipe. Le chargé de TP pourra interroger l'équipe et individuellement certains étudiants. Un étudiant trop hésitant sur une question le concernant ou n'ayant pas une vision globale du projet pourra se voir pénaliser par rapport à ses coéquipiers. **Tous les étudiants du groupe doivent être présents sous peine d'être pénalisés.**



Attention, cette soutenance vise à vérifier si votre travail répond aux exigences du client. Ce dernier (le correcteur) teste votre projet sur la version déposée sur campus dans une machine de l'école et en vérifie l'adéquation avec ses attentes. S'il « achète » votre projet, vous avez une bonne note, sinon... elle sera mauvaise. Si vous avez développé sur un système particulier, venez avec votre matériel. Un plantage le jour du test client est du plus mauvais effet alors préparez-vous un peu à l'avance et non à la dernière minute.

Critères d'évaluation du travail à réaliser

En respect des consignes spécifiées précédemment, chaque équipe sera évaluée lors de la soutenance sur les critères suivants :

- *La démonstration en lançant le **jar** de votre code en adéquation avec celle déposée sur campus. Elle est notée sur 20 avec un coefficient de 2, dont 2 points pour le **versioning GIT** et 3 points pour le **module de Reporting et l'enrichissement de la base existante**. Vous serez jugés sur les points suivants : la fiabilité de votre système, l'enrichissement de la base existante, l'interaction de votre programme Java avec la Base de données, la qualité de la démonstration (le soin particulier apporté à sa qualité visuelle, sa fluidité, la pertinence des informations affichées en mode graphique, son originalité particulière, ...), et la **Javadoc** (montrant la qualité et les commentaires de votre code).*
- *Le **Powerpoint** noté sur 20 avec un coefficient de 1 évaluera la qualité des slides et de la présentation orale (respect du temps, clarté dans les explications, cohérence de l'argumentation, cohésion de l'équipe)*
- *Des pénalités pourraient être appliquées sur la note de la soutenance dans les cas suivants : retard de dépôt du code ou/et du rapport (-2 points par jour de retard pour chaque dépôt), fichiers manquants dans le code déposé, plagiat, absence à une soutenance, prestation individuelle peu ou pas convaincante.*
- *La note finale du projet sera égale à (démonstration * 2 + Powerpoint) / 3.*

Recommandations

Le jour de la soutenance, vous devrez faire tourner le système que vous aurez déposé sur campus. Aucune amélioration ou nouvelle version développée ultérieurement ne sera acceptée.

Pensez, après le dépôt sur campus, à télécharger votre dépôt sur une machine de l'école pour le faire tourner dans les conditions de la soutenance et pallier les anomalies.

Ne vous laissez pas abuser par l'illusion d'avoir beaucoup de temps. Respectez l'organisation prévue pour avoir terminé avant les vacances. Commencez suffisamment tôt pour vous organiser au mieux et faire quelque chose de bien. Ne soyez pas trop ambitieux au départ. Préférez faire proprement un projet simple et bien testé avant de partir dans les extensions. Travaillez par version en utilisant un logiciel de **versioning GIT**. Faites des modifications seulement sur une nouvelle version en conservant précieusement la dernière qui marche. C'est toujours trop bête de planter le jour de la soutenance. Ne sous-estimez pas les phases de test qui doivent être régulières et sont presque aussi importantes que les phases de développement.

Pensez à faire appel à vos amis ou votre famille pour tester et critiquer ce que, vous-même, ne pouvez critiquer objectivement.

Santé !!

Annexe : exemples de requêtes

La liste suivante non exhaustive est celle d'exemples de requêtes que vous devez interroger via votre module de recherche. Pour chaque requête vous trouverez le résultat que vous devez obtenir pour vous permettre de vérifier vous-même votre solution (il y a en général plusieurs solutions possibles, avec des variantes de syntaxe).

R1. Prénom et nom des malades affiliés à la mutuelle « MAAF ».

prenom nom

Edina Gallovits
Victor Hanesu
Paul Capdeville
Olga Poutchkova
Victoria Azarenka
Akiko Morigami
Mara Santangelo
Peter Luczak

R2. Prénom et nom des infirmier(ères) travaillant pendant la rotation de nuit.

prenom nom

Marion Bartoli
Tomas Berdych
James Blake
Guillermo Canas
Arnaud Clement
Alize Cornet
Florian Mayer
Alicia Molik
Virginie Razzano
Andy Roddick
Florent Serra
Gilles Simon
Ai Sugiyama
Serena Williams

R3. Donner pour chaque service, son nom, son bâtiment, ainsi que les prénom, nom et spécialité de son directeur.

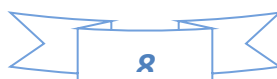
nom batiment prenom nom specialite

Cardiologie B Lleyton Hewitt Cardiologue
Chirurgie generale A Nathalie Dechy Pneumologue
Reanimation et Traumatologie A Marat Safin Traumatologue

R4. Donner pour chaque lit occupé du bâtiment « B » de l'hôpital occupé par un malade affilié à une mutuelle dont le nom commence par « MN... », le numéro du lit, le numéro de la chambre, le nom du service ainsi que le prénom, le nom et la mutuelle du malade l'occupant.

no_chambre lit nom prenom nom mutuelle

101 1 Cardiologie Milagros Sequera MNAM
101 3 Cardiologie Shuai Peng MNH
105 2 Cardiologie Jiri Vanek MNAM



R5. Quelle est la moyenne des salaires des infirmiers(ères) par service ?

code_service moyenne_des_salaires

CAR	1588.42
CHG	1472.38
REA	1370.58

R6. Pour chaque service du bâtiment « A » de l'hôpital, quel est le nombre moyen de lits par chambre ?

code_service nb_moyen_de_lits

CHG	2.6
REA	1.5

R7. Pour chaque malade soigné par plus de 3 médecins donner le nombre total de ses médecins ainsi que le nombre correspondant de spécialités médicales concernées.

nom prenom nb_soignants nb_specialites

Ascione	Thierry	4	3
Ditty	Julie	4	4
Dushevina	Vera	4	4
Govortsova	Olga	4	3
Karlovic	Ivo	4	3
Muller	Martina	4	4
Osterloh	Lilia	4	3
Vakulenko	Julia	4	4

R8. Pour chaque service quel est le rapport entre le nombre d'infirmier(ères) affecté(es) au service et le nombre de malades hospitalisés dans le service ?

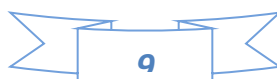
nom rapport_i_sur_m

Cardiologie 0.8889
 Chirurgie generale 0.6500

R9. Prénom et nom des docteurs ayant au moins un malade hospitalisé.

prenom nom

Igor Andreev
 Nathalie Dechy
 Elena Dementieva
 Novak Djokovic
 Gisela Dulko
 Roger Federer
 David Ferrer
 Richard Gasquet
 Tatiana Golovin
 Daniela Hantuchova
 Justine Henin
 Lleyton Hewitt
 Martina Hingis
 Ana Ivanovic
 Svetlana Kuznetsova
 Paul-Henri Mathieu
 Gael Monfils



Carlos Moya
Andy Murray
David Nalbandian
Olivier Rochus
Marat Safin
Fabrice Santoro
Radek Stepanek
Venus Williams
Vera Zvonareva

R10. Prénom et nom des docteurs n'ayant aucun malade hospitalisé.

prenom nom

Jonas Bjorkman
Sebastien Grosjean
Amelie Mauresmo
Rafael Nadal
Pauline Parmentier
Vincent Spadea