**Projet : Cabinet Médical**

par Boris Nzaramba (8079777),

Matthew Watters (3841418)

et Caleb St-Denis (7256131)

Soumis à

Professeure Amal Zouaq et Mme Joyce Habbouche

dans le cadre du cours CSI2532A – Base de données I

Université d’Ottawa

École de science informatique et de génie électrique

le 11 avril 2016

Table des matières

[1 Modèle conceptuel EER 1](#_Toc448103485)

[2 Modèle relationnel 2](#_Toc448103486)

[3 Requêtes SQL 3](#_Toc448103487)

[4 Objectif d’application 7](#_Toc448103488)

[4.1 Défi 7](#_Toc448103489)

[5 Guide d’utilisateur 7](#_Toc448103490)

[5.1 Installation 7](#_Toc448103491)

[5.2 Démonstration 8](#_Toc448103492)

[5.2.1 Interface Médecin 8](#_Toc448103493)

[5.2.2 Interface Secrétaire 10](#_Toc448103494)

# Modèle conceptuel EER

# Modèle relationnel

SECRETAIRE (SecID(PK), Prenom, Nom, NumTel, Adresse)

PK SecID

MEDECIN (MedID(PK), Prenom, Nom, NumTel, Adresse, Specialite, ecID(FK))

PK MedID

FK SecID references SECRETAIRE (SecID)

PATIENT (PatID(PK), Prenom, Nom, NumTel, Adresse, dateNaissance, sexe, SSN, RefMed(FK))

PK PatID

FK RefMed references MEDECIN (MedID)

CONSULTATION (PatID(PK,FK), MedID(PK,FK), CDate(PK), Heure, Duree, Objet)

PK PatID, MedID, CDate

FK PatID references PERSONNE (PerID)

FK MedID references PERSONNE (PerID)

SUBSTANCEACTIVE (SubID(PK), Nom)

PK SubID

INCOMPATIBLESUBSTANCEACTIVE (SubA(FK), SubB(FK))

PK SubA, SubB

FK SubA references SUBSTANCEACTIVE(SubID)

FK SubB references SUBSTANCEACTIVE(SubID)

MEDICAMENT (MedID(PK), Nom, Prix, estGenerique, SubID(FK))

PK MedID

FK SubID references SUBSTANCEACTIVE(SubID)

PRESCRIPTIONEXAM (ERXID(PK), Nom, PatID(FK), MedID(FK), CDate(FK))

PK ERXID

FK (PatId, MedId, CDate) references CONSULTATION(PatId, MedId, CDate)

PRESCRIPTIONMEDI (MRXID(PK), Duree, MediID, PatID(FK), MedID(FK), CDate(FK))

PK MRXID

FK MediID references MEDICAMENT (MediID)

FK (PatId, MedId, CDate) references CONSULTATION(PatId, MedId, CDate)

PATHOLOGIE (PathID(PK), Nom)

PK PathID

INCOMPATIBLESUBPATH (SubID(FK), PathID(FK))

PK SubID, PathID

FK SubID references SUBSTANCEACTIVE (SubID)

FK PathID references PATHOLOGIE (PathID)

PATHOLOGIEPATIENT (PatID(FK), PathID(FK), dateDebut(PK), dateFin)

PK PatID, PathID, dateDebut

FK PatID references PATIENT (PatID)

FK PathID references PATHOLOGIE (PathID)

# Requêtes SQL

SET search\_path = cabinetMD;

1. SELECT \*

FROM Medecin

ORDER BY nom ASC;

1. SELECT \*

FROM Patient

WHERE refmed = 'M001';

1. SELECT \*

FROM Patient

WHERE refMed IN

(SELECT medID

FROM Medecin

WHERE LOWER(nom) LIKE '%blay%');

1. SELECT \*

FROM Patient

WHERE refMed IN

(SELECT medID

FROM Medecin

WHERE nom = 'Tremblay' OR nom = 'Solo');

1. (SELECT patID

FROM PathologiePatient

WHERE pathID = (SELECT pathID FROM Pathologie WHERE nom = 'rhume'))

EXCEPT

(SELECT patID

FROM PathologiePatient

WHERE pathID = (SELECT pathID FROM Pathologie WHERE nom = 'grippe'));

1. Hypothèse: Pour une maladie donnée, un patient est "traité" par le medecin qui a diagnostiqué la maladie. (Alors il suffit de lister les medecins qui ont diagnostiqué les maladies rhume et grippe)

(SELECT medID

FROM PathologiePatient

WHERE PathID IN (SELECT pathID FROM Pathologie WHERE nom = 'rhume'))

INTERSECT

(SELECT medID

FROM PathologiePatient

WHERE PathID IN (SELECT pathID FROM Pathologie WHERE nom = 'grippe'));

1. SELECT COUNT(\*) AS myCount

FROM

(SELECT patID

FROM Consultation

WHERE medID IN (SELECT medID FROM Medecin WHERE nom = 'Tremblay')

INTERSECT

(SELECT patID

FROM PathologiePatient

WHERE pathID = (SELECT pathID FROM Pathologie WHERE nom = 'cancer du foie'))) AS liverCancerPatientsTreatedByTremblay;

1. SELECT COUNT(\*) AS myCount

FROM

(SELECT patID

FROM Consultation

WHERE medID IN (SELECT medID FROM Medecin WHERE nom = 'Tremblay')

EXCEPT

(SELECT patID

FROM PathologiePatient

WHERE pathID = (SELECT pathID FROM Pathologie WHERE nom = 'cancer du foie'))) AS R;

1. SELECT M.medID, M.nom, Count(P.patID)

FROM Medecin M INNER JOIN Patient P

ON P.refMed = M.medID

GROUP BY medID;

1. En se basant sur la même hypothèse que pour la requête (6), il suffit il suffit de trouver les médecins qui ont diagnostiqués le cancer et pour quels patients.

SELECT medID, COUNT(patID) AS myCount

FROM PathologiePatient

WHERE pathID IN (SELECT PathID FROM Pathologie WHERE nom LIKE '%cancer%')

GROUP BY medID;

-La table temporaire suivante est utilisée pour les requêtes 11 à 13.

-Elle consiste de la table IncompatibleSubstanceActive augmentée avec toutes les paires inverses.

-Une contrainte sur la table IncompatibleSubstanceActive prévient l'ajout de paires inverses puisque c'est une redondance qui peut causer des anomalies. Cependant, cette redondance facilite la recherche de substances incompatibles pour les requêtes.

DROP TABLE IF EXISTS Incompatibilite;

CREATE TEMP TABLE Incompatibilite AS

(SELECT S.SubID, IA.SubB AS SubIncompatible

FROM SubstanceActive S INNER JOIN IncompatibleSubstanceActive IA

ON S.SubID = IA.SubA)

UNION ALL

(SELECT S.SubID, IB.SubA

FROM SubstanceActive S INNER JOIN IncompatibleSubstanceActive IB

ON S.SubID = IB.SubB);

1. Il suffit de trouver les médicaments couramment prescrits au patient P001 qui comportent des substances actives incompatibles avec celle du médicament M003.

SELECT MediID

FROM PrescriptionMedi P NATURAL JOIN Medicament M

WHERE P.patID = 'P001' AND P.Duree > CURRENT\_DATE --On veut uniquement les medicaments du patient P001 en cours de prise.

AND M.SubID IN

(SELECT I.SubIncompatible

FROM Medicament NATURAL JOIN Incompatibilite I

WHERE MediID = 'M003');

1. SELECT S.SubID, I.SubIncompatible

FROM SubstanceActive S LEFT JOIN Incompatibilite I

ON S.SubID = I.SubID

ORDER BY SubID,SubIncompatible;

1. CREATE OR REPLACE FUNCTION reject\_incompatible\_prescription() RETURNS TRIGGER AS $reject\_incompatibility$

DECLARE

patient CHAR(4) := NEW.PatID;

newMed CHAR(4) := NEW.MedID;

BEGIN

* Obtenir les substances contre-indiquées pour le patient à qui on souhaite prescrire un nouveau médicament

DROP TABLE IF EXISTS MedicamentsContreIndiques;

CREATE TEMP TABLE MedicamentsContreIndiques AS

SELECT MediID

FROM Medicament

WHERE SubID IN

* Les subtances contre-indiqués par les medicaments du patient en cours de prise

(SELECT SubIncompatible

FROM Incompatibilite NATURAL JOIN Medicament

WHERE MediID IN

(SELECT MediID FROM PrescriptionMedi WHERE PatID = 'P001' AND Duree > CURRENT\_DATE))

OR SubID IN --les substances contre-indiquées par les pathologies du patient

(SELECT SubID

FROM IncompatibleSubPath

WHERE PathID IN (SELECT PathID FROM PathologiePatient WHERE PatID = 'P001'));

IF EXISTS (SELECT \* FROM MedicamentsContreIndiques WHERE MediID = newMed) THEN

RAISE EXCEPTION 'Le medicament % est contre-indiqué pour le patient %',newMed,patient;

END IF;

RETURN NEW;

END;

1. DELETE FROM Medecin

WHERE nom='Bos' AND prenom='Johan';

1. UPDATE Patient

SET adresse = '1000 Strawberry Lane'

WHERE nom='Valls' AND prenom = 'Manuel';

$reject\_incompatibility$ LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS reject\_incompatibility ON PrescriptionMedi;

CREATE TRIGGER reject\_incompatibility

BEFORE INSERT ON PrescriptionMedi

FOR EACH ROW EXECUTE PROCEDURE reject\_incompatible\_prescription();

# Objectif d’application

L’objectif de l’application, qu’on a nommé CabMD dans l’interface, est de permettre aux membres du cabinet de gérer des consultations tel qu’indiqué dans les spécifications. C’est-à-dire, l’application permet aux secrétaires de créer des consultations et permet aux docteurs d’en ajouter l’objet. Les consultations se font désormais pour des patients qui sont déjà enregistrés au cabinet. L’ajout de nouveaux patients demeure hors la portée de cette version de l’application.

## Défi

On s’est fixé le défi de rendre l’application le plus usable que possible pour ses utilisateurs. Pour ce faire, on a tenté de développer une interface simple et intuitive qui prend en compte le « workflow » des secrétaires et des médecins, afin que ces derniers puissent accomplir leurs tâches de manière efficace. Ceci se voit surtout dans l’interface des médecins où la navigation de consultation à consultation nécessite un seul clic (voir la section 2.2.1).

# Guide d’utilisateur

## Installation

Veuillez suivre les étapes suivantes pour rouler l’application localement :

1. S’assurer d’avoir installé et correctement configuré XAMPP localement et PostgreSQL sur votre serveur de base de données (ou localement).
2. Extraire le dossier CabinetMedical du .ZIP joint à ce rapport.
3. Placer se dossier à l’intérieur de votre installation XAMPP, dans le répertoire /xampp/htdocs
4. Créer une base de donnée locale et rouler le script /CabinetMedical/SQL/bdschema.sql (et optionnellement data.sql pour populer les tables avec des données échantillons).
5. Modifier le fichier /CabinetMedical/dbphp/Configuration.php avec les informations de la base de données que vous avez créée, à savoir, à savoir le host (l’adresse IP du serveur de base de données), le port, le dbname, et les credentials (nom d’usager et mot de passe).
6. Commencez le serveur Apache à partir du « XAMPP Control Panel »
7. Naviguer à <http://localhost/CabinetMedical/dbphp/main.php>

Vous devez maintenant voir l’écran de connexion de l’application :



Voir la section suivante pour un démonstration de l’application.

## Démonstration

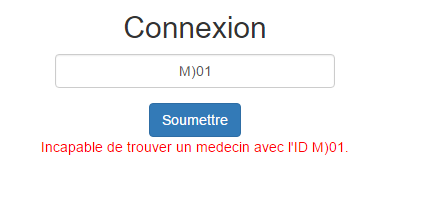
Note application « *CabMD »* a pour objectif principal de permettre aux membres du cabinet médical d’organiser dynamiquement et en toute simplicité leurs consultations. Il fallait que l’ergonomie d’utilisation de l’application soit la plus efficace possible pour permettre un gain de temps et un confort optimal aux usagers. Pour ce faire, nous avons divisé le logiciel en deux interfaces : l’une pour les médecins et l’autre pour les secrétaires. Ainsi, après sa connexion, chaque utilisateur est offert un design adapté pour son travail et ses spécifications. Tout ce dont un médecin a besoin est de s’informer de ses consultations tandis que le(a) secrétaire est chargé(e) de les créer, de les mettre à jour ou de les supprimer si besoin.

### Interface Médecin

Un médecin se connecte à l’application en entrant leur ID :



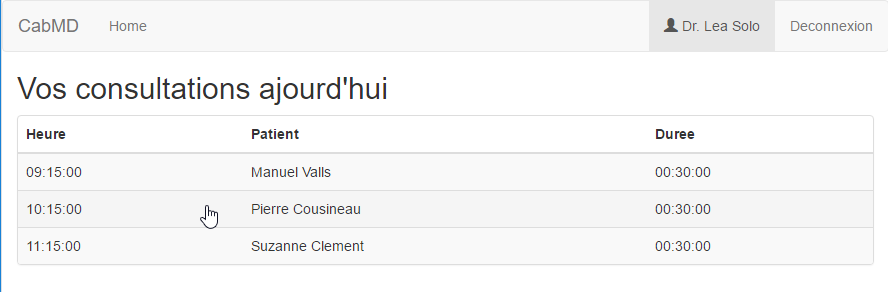
S’il ou elle arrive à faire une faute, l’application leur en informera avec un message d’erreur approprié.



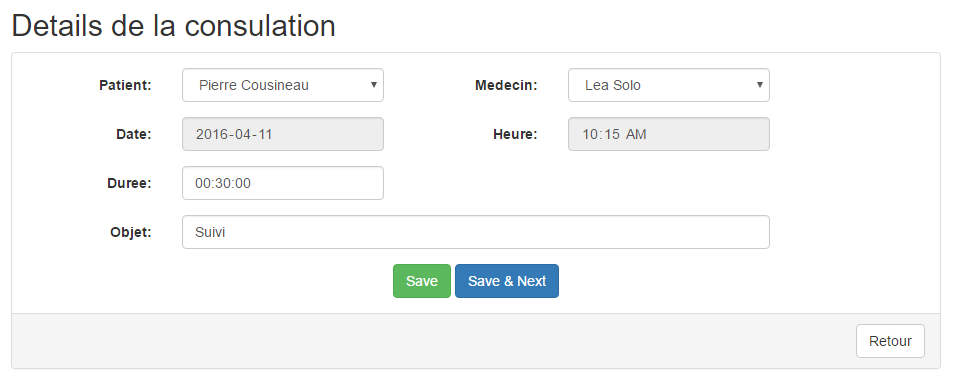
Une fois connecté, l’application affichera un menu principal où il ou elle peut survoler ses consultations de la journée, et aussi ses consultations prochaines s’il y en a.



Au début d’une consultation, le médecin sélectionne la rangée correspondante dans le menu pour l’explorer en détail.



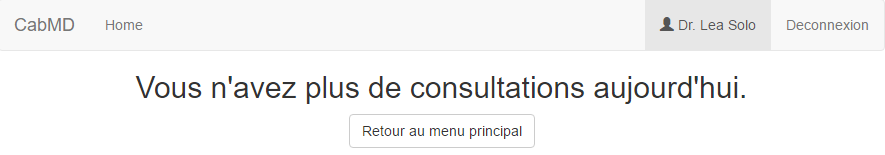
Le médecin peut maintenant modifier l’objectif de la consultation après avoir parlé au patient. Il ou elle peut aussi modifier la durée s’il y a un délai quelconque.



Une fois la consultation terminée, le médecin n’a qu’à cliquer le bouton « Save & Next » pour passer à la prochaine consultation et répéter le processus.

Ou bien, il ou elle peut cliquer « Save ». Dans ce cas, l’application affichera un message de succès ou d’erreur et retournera au menu principal. Le bouton « Retour » a aussi ce dernier effet (sans la sauvegarde).

Si le médecin clique « Save & Next » sur sa dernière consultation de la journée, il ou elle obtiendra le message suivant :



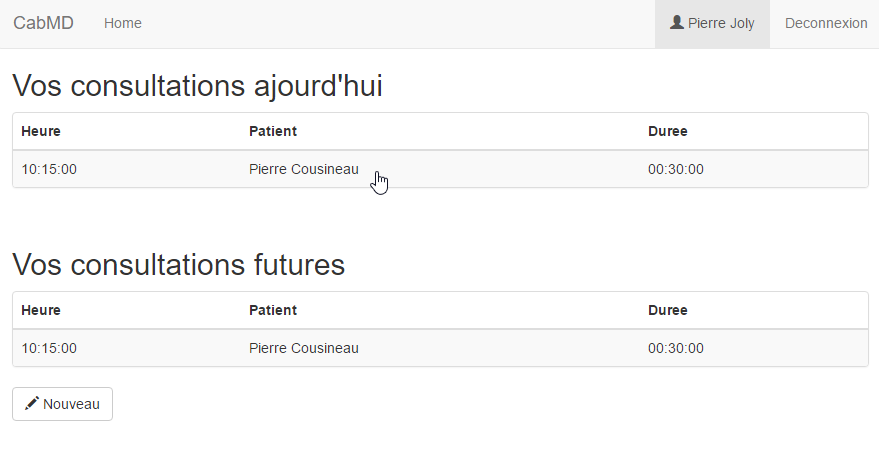
Le médecin peut de déconnecter n’importe quand avec le bouton « Déconnexion » en haut à droite. Ainsi, l’application retourne à la page de connexion.

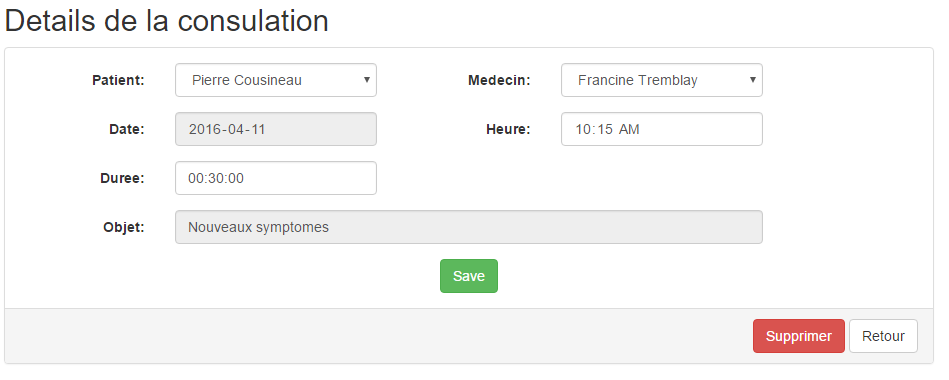
### Interface Secrétaire

Les secrétaires se connectent à travers la même page d’accueil. L’application déterminera selon l’ID entré s’il s’agit d’un médecin ou d’un(e) secrétaire.



Les secrétaires ont un menu principal fort similaire à celui des médecins. Ce menu affiche les consultations des médecins pour qui le ou la secrétaire travaille. On explore la consultation en détail en cliquant dessus.

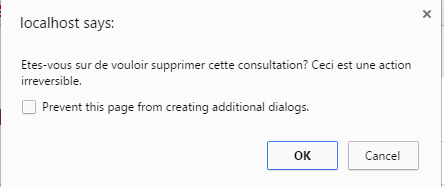




À ce niveau, il y a plus de différences avec l’interface des médecins.

Les secrétaires peuvent modifier l’heure et la durée d’une consultation, mais pas l’objet (car ceci est fait par le médecin). De plus, ils peuvent supprimer des consultations au cas où le patient ou le médecin doit annuler le rendez-vous, ou bien si une consultation fausse a été créée par erreur.

Les secrétaires doivent confirmer l’action de suppression car elle est irréversible.

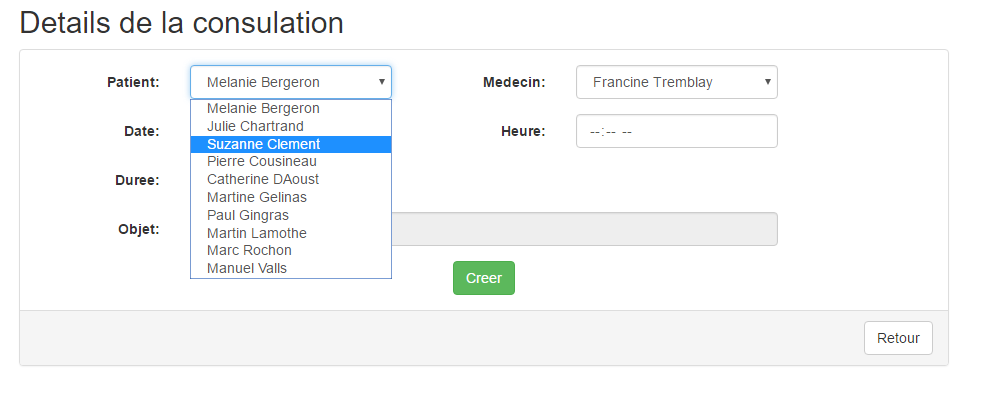


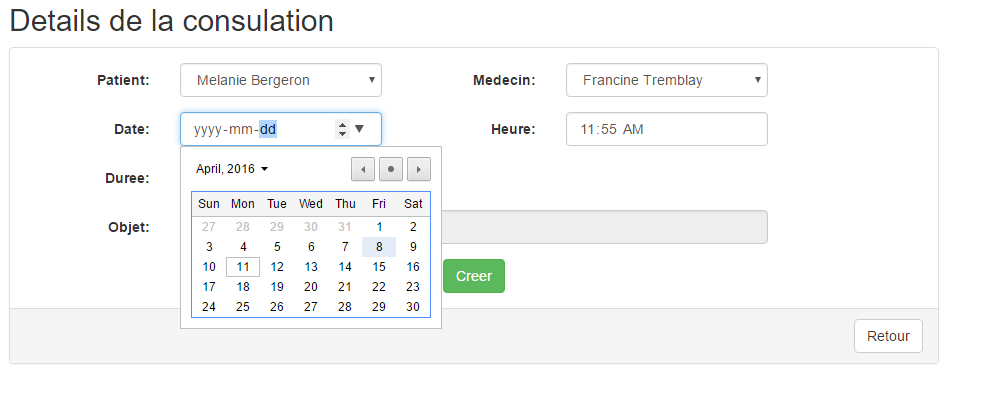
Ceci nous ramène au menu principal.

Les secrétaires peuvent créer de nouvelles consultations en cliquant le bouton « Nouveau » en bas de page.

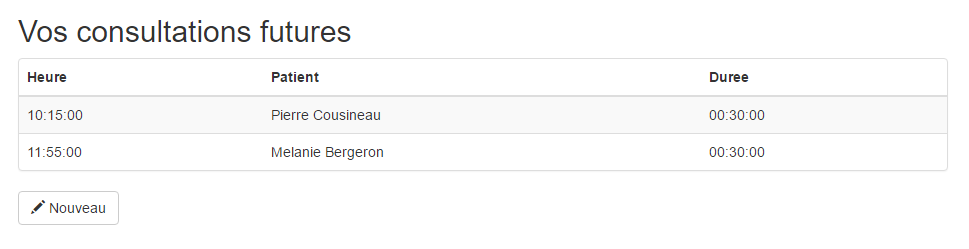


Ceci nous amène à un formulaire similaire à celle pour les consultations existantes. Les secrétaires choisissent un patient du cabinet, un de leurs docteurs, une date, et ensuite l’heure et la durée. Ils cliquent le bouton « Créer » pour finaliser le processus.





La nouvelle consultation apparaîtra sur le menu principal. Elle est aussi visible au médecin correspondant.



Les secrétaires se déconnectent de la même manière que les médecins.