

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: Modèles Intelligents et de Décision (M.I.D)

Thème

Réalisation d'une application de gestion dans un centre d'imagerie médicale

Réalisé par :

- BENABADJI Abdelkarim

Présenté le 24 Juin 2014 devant le jury composé de MM.

- HADJILA Feth-Allah (Président)
- BRIKCI-NIGASSA Amine (Encadreur)
- KHITRI Souad (Examineur)
- KAZI-TANI Adila (Examineur)

Année universitaire : 2013-2014

Remerciements

Je remercie Dieu de m'avoir donné la chance d'étudier dans un domaine qui m'est cher et de m'avoir permis de rencontrer des gens tout à fait intéressants pendant ma Licence et mon Master, j'adresse aussi mes remerciements à ma famille et à mon entourage pour leur soutien, leur patience et leur persévérance. Je remercie aussi les gens du centre d'imagerie médicale qui m'ont donné la chance de les assister dans leur travail et de contribuer à faire avancer les choses. Je remercie aussi les professeurs chez qui j'ai pu assister durant mon cursus universitaire, je remercie tout particulièrement mon encadreur, M. BRIKCI-NIGASSA Amine, pour ses conseils avisés et ses remarques pertinentes. Je remercie aussi les membres du jury, M. HADJILA Feth-Allah, M^{me} KHITRI Souad et M^{me} KAZI-TANI Adila pour avoir accepté de juger le présent travail. Enfin, je remercie vraiment tout le monde car si un simple battement d'ailes d'un papillon peut déclencher une tornade à l'autre bout du monde, je pense que pour la réalisation de ce mémoire, j'ai aussi eu droit à un battement d'aile mais d'un autre type.

Dédicaces

Je dédie mon travail à tous ceux qui ont participé de près ou de loin à sa concrétisation, de mes parents aux gens qui ont créé les outils qui ont permis l'implémentation de l'application en passant par ceux qui inventé et ceux qui ont fabriqué nos ordinateurs, à ceux qui inventent de nouvelles choses chaque jour, à ceux qui améliorent nos vies, à ceux dont personne n'entend parler, à ces gens cachés dans l'ombre mais qui jouent aussi leurs rôles dans l'évolution du monde. Je ne dédie pas seulement ce mémoire aux gens mais aussi à tous les êtres et à toutes les sciences permettant de faire avancer le monde vers des jours meilleurs.

Table des matières

Liste des figures	3
Introduction générale	5
I. Chapitre I : Etude de l'existant.....	7
I.1. Présentation du client.....	7
I.2. Etapes par lesquelles passe chaque patient	8
I.3. Proposition d'une solution	11
II. Chapitre II : Les outils utilisés.....	13
II.1. Base de données	13
a. Les langages SQL associés aux produits	13
b. Choix du Système de gestion de bases de données.....	15
c. Oracle MySQL Workbench	16
II.2. Le framework Qt	17
a. Présentation	17
b. Qt Creator.....	21
III. Chapitre III : Réalisation de l'application.....	26
III.1. Conception de la base de données.....	26
III.2. Interface utilisateur de l'application.....	31
III.3. Les différentes classes de l'application	39
Conclusion et perspectives.....	49
Références Bibliographiques	52
Annexe A : Installation des outils utilisés pour l'implémentation de l'application sous GNU/Linux Debian Wheezy	57
1. Le framework Qt	57
2. MariaDB.....	57
3. MySQL Workbench	57

Annexe B : Installation des outils nécessaires pour le fonctionnement de l'application sous GNU/Linux Debian Wheezy	58
Annexe C : Installation des outils utilisés pour l'implémentation de l'application sous Windows	59
1. Le framework Qt	59
2. MariaDB et MySQL Workbench	59
Annexe D : Les DLL nécessaires pour fonctionnement de l'application sous Windows	60
Annexe E: Génération des DLL de MySQL.....	61
Glossaire	62

Liste des figures

Figure I.1 Organigramme récapitulatif des étapes par lesquelles passe chaque patient .	10
Figure II.1 Capture d'écran de SQL Workbench	16
Figure II.2 Schéma montrant les différents composants de bureau des principaux systèmes d'exploitation que Qt utilise ^[40]	18
Figure II.3 Apparence de l'application réalisée pour le mémoire sous Windows.....	18
Figure II.4 Apparence de l'application réalisée pour le mémoire sous GNU/Linux sous l'environnement de bureau KDE.....	19
Figure II.5 Principaux éléments de Qt Creator	22
Figure II.6 Le mode Design dans Qt Creator.....	23
Figure II.7 Choix du mode de compilation.....	24
Figure III.1 Modèle relationnel de la base de données (obtenu par MySQL Workbench)	26
Figure III.2 Liste des tables de la base de données.....	27

Introduction générale

Introduction générale

Avec l'expansion des tailles des disques durs et l'accélération du traitement des informations, les bases de données sont devenues bien plus utiles qu'avant. La démocratisation de l'accès à Internet et donc la familiarisation avec les moteurs de recherche sur le Web ont accentué le vœu de disposer de sa propre base de données. Le besoin de journaliser les informations et de les retrouver ensuite en une fraction de seconde avec juste quelques mots clés n'est plus un problème et l'expression "chercher une aiguille dans une botte de foin" n'a plus lieu d'être. Dans ce mémoire, nous proposons une application de gestion utilisant le récent système de gestion de bases de données MariaDB, un SGBD dérivé du désormais célèbre MySQL, qui en prend les avantages tout en rajoutant de nouvelles fonctionnalités au fur des versions.

Le choix de ne pas obliger les utilisateurs à se cantonner à un seul système d'exploitation est une priorité. De plus, le logiciel doit avoir une interface unifiée. Le but est alors d'arriver à créer un logiciel portable et léger, et le framework Qt est tout destiné à la réalisation de cet objectif. De plus, les outils qu'il offre et sa documentation disponible et riche ont motivé notre choix.

Ce mémoire s'articule sur trois chapitres. Dans le chapitre I, nous présentons l'étude des éléments aidant à l'élaboration de l'application. Ensuite, le chapitre II explique nos motivations de choix des outils utilisés. Et enfin, dans le chapitre III, nous présentons à proprement parler l'application réalisée et son utilisation au quotidien.

CHAPITRE I :

Etude de l'existant

I. Chapitre I : Etude de l'existant

I.1. Présentation du client

Ayant travaillé 4 mois dans un centre d'imagerie médicale en tant que caissier, j'ai pu voir et étudier le fonctionnement de ce centre entre temps. Pour ce mémoire, j'ai donc proposé d'élaborer une application pour la gestion des patients.

Le client est donc un centre d'imagerie médicale, il contient 6 services différents :

1. Service de radiologie ;
2. Service de scanner ;
3. Service d'échographie ;
4. Service de mammographie ;
5. Service d'IRM ;
6. Service de cytoponction.

Le centre d'imagerie est composé de :

1. Une réceptionniste

La réceptionniste enregistre pour chaque patient, le nom, le prénom, le nom du mari éventuellement, la date de naissance et le type d'examen à pratiquer.

Elle oriente aussi les personnes désirant avoir des informations.

Elle enregistre aussi les rendez-vous demandés par téléphone.

2. Dix manipulateurs

Chaque manipulateur est assigné à un service. Il sait manipuler les machines de son service pour pouvoir pratiquer les examens.

3. Un caissier

Le caissier enregistre pour chaque patient, le nom, le prénom, le nom du mari éventuellement, la date de naissance, le type d'examen pratiqué, le montant compté pour l'examen, le montant versé et le montant restant ou juste le montant ajouté si c'est un complément, et la date de paiement.

4. Quatre médecins

Pour chaque examen, un des médecins interprète les résultats.

Chaque médecin sait aussi manipuler quelques machines, il peut donc endosser le rôle de manipulateur.

5. Trois secrétaires médicales

Chaque secrétaire a un ordinateur à sa disposition, il lui permet d'enregistrer l'interprétation des résultats pour chaque examen dictée par un des médecins.

I.2. Etapes par lesquelles passe chaque patient

Chacun a son rôle à jouer dans la clinique pour le bon déroulement d'un examen donné.

Quand un patient vient faire un examen :

1. La réceptionniste doit enregistrer son nom, prénom, nom du mari si c'est une femme mariée, date de naissance pour avoir l'âge, nom du médecin figurant sur l'ordonnance et enfin le type d'examen à passer. La réceptionniste lui donne un bon sur lequel figurent ces informations sauf le nom du médecin.
2. Le patient passe en salle d'attente ;
3. Quand le patient est appelé par la réceptionniste pour passer son examen, il est dirigé vers un des services d'examen selon le groupe de son type d'examen ;
4. A chaque service d'examen, un ou plusieurs manipulateurs sont attachés. Quand le patient passe chez un manipulateur, ce dernier ajoute sur son bon si le patient est conventionné ou non et avec quelle convention, et le montant compté pour l'examen ;
5. Une interprétation des résultats de l'examen est ensuite faite par un des médecins du centre d'imagerie et dictée à une des secrétaires, qui tape l'interprétation sur un ordinateur puis l'imprime et la met avec le cliché de l'examen dans une enveloppe ;
6. Le patient se dirige ensuite vers la caisse pour payer l'examen qu'il vient de passer. Après encaissement, et selon si le patient a payé la totalité du montant ou

une partie seulement, le bon est cacheté par le caissier ou ce dernier indique sur le bon le montant versé et le montant restant sans le cacheter jusqu'à ce que le patient complète entièrement le versement ;

7. Quand les résultats de l'examen sont prêts, le patient peut les retirer chez la réceptionniste via son bon préalablement cacheté.

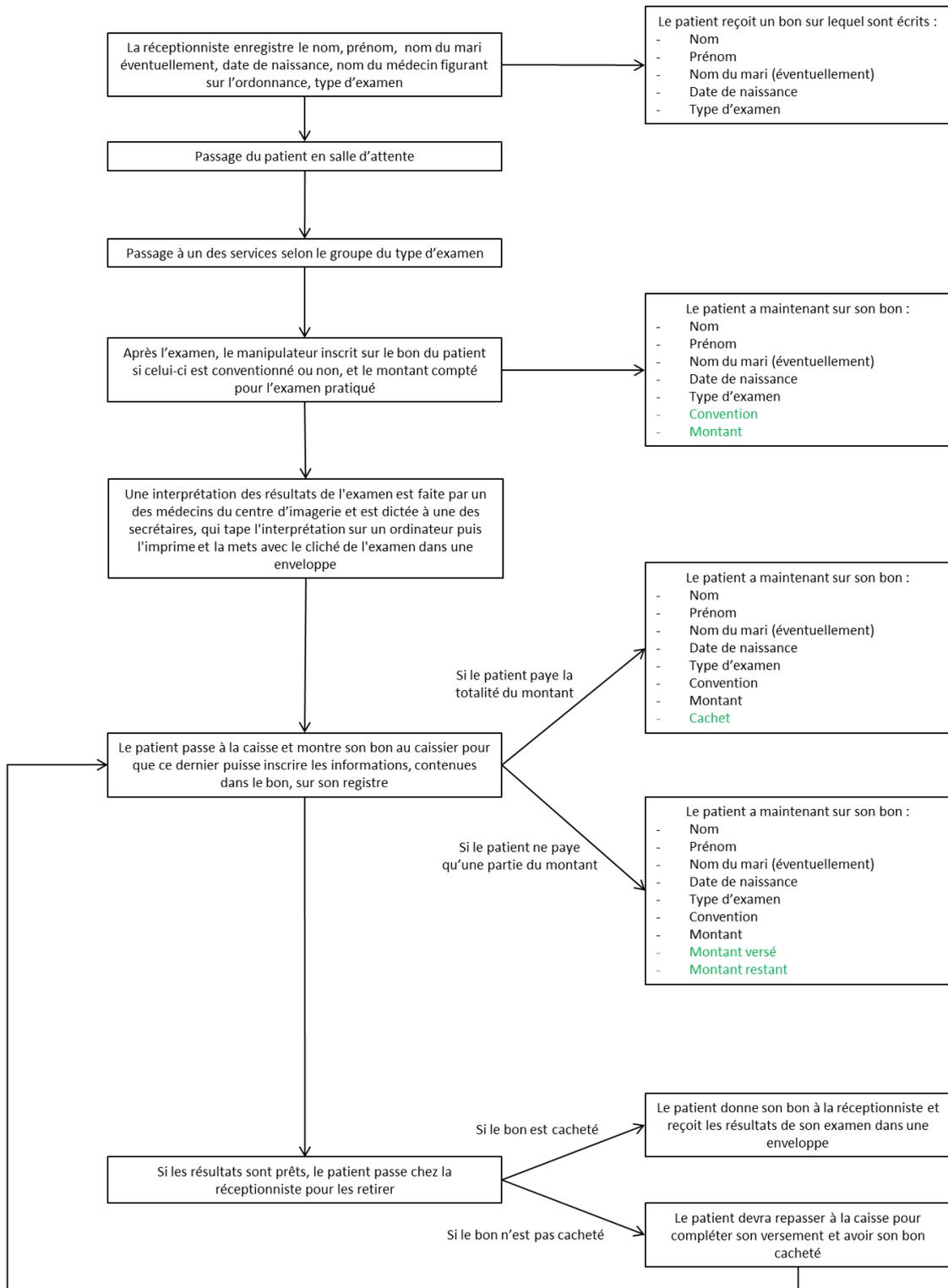


Figure I.1 Organigramme récapitulatif des étapes par lesquelles passe chaque patient

I.3. Proposition d'une solution

Le problème est que toutes ces étapes sont faites sur papier, sauf pour les interprétations des résultats qui sont enregistrées sur une base de données et, donc, demandent l'enregistrement du nom, prénom, type d'examen, etc. Si le nom d'un patient doit être retrouvé, la recherche peut prendre beaucoup de temps pour ceux qui n'ont pas accès à la base de données.

Afin de faciliter, d'accélérer et de journaliser leurs travaux, une base de données centralisée peut être un bon choix. Il convient donc de mettre en place un serveur pour la base de données et chaque utilisateur devra avoir un ordinateur à disposition relié au serveur.

CHAPITRE II :

Les outils utilisés

II. Chapitre II : Les outils utilisés

II.1. Base de données

a. Les langages SQL associés aux produits

Chaque système de gestion de bases de données relationnelles (SGBDR) a son langage SQL propre. Parmi les solutions les plus connues, nous pouvons citer :

- *MySQL*

MySQL est un SGBD libre/open source. Il fait partie des SGBD les plus utilisés dans le monde, surtout dans le monde des bases de données Web ^[1]. Beaucoup de grandes sociétés telles que Twitter, Facebook, EADS et YouTube, réduisent leurs coûts de manière significative en utilisant MySQL pour leurs services. ^{[2][3]}

MySQL AB, la société produisant ce SGBDR, a été acquise en 2008 par Sun Microsystems ^[4]. En 2009, Sun Microsystems a été, à son tour, rachetée par Oracle Corporation, la société produisant une des solutions concurrentes à MySQL, Oracle Database ^[5].

MySQL est multiplateforme et distribué sous une double licence GPL et propriétaire ^[6].

- *MariaDB*

MariaDB est aussi un SGBDR libre. Il s'agit surtout d'un dérivé (*fork*) de MySQL géré par la fondation MariaDB.

Son créateur, Michael "Monty" Widenius est aussi l'un des co-auteurs du SGBDR originel : le nom de MySQL est en référence au nom de sa première fille "My", et celui de sa dernière fille "Maria" est représenté dans le nom de son nouveau projet.

Peu de temps après que MySQL AB ait été acquise par Sun Microsystems, Widenius démissionna. Il fondera plus tard sa propre société, Monty Program AB, une société qui appartient à ses employés et dont il n'est pas propriétaire. Le but étant de ne pas reproduire les erreurs du passé, en mettant le projet sous licence GPL v2 et en le maintenant par une société appartenant à ses employés pour éviter toute offre alléchante de rachat par une grande firme et pour "[ne] pas voir tout ce travail disparaître parce

qu'il était tombé dans de mauvaises mains" selon son instigateur.

L'objectif de MariaDB est d'arriver comme remplaçant binaire de MySQL sans nécessiter de trop grands travaux. Cela sous-entend une équivalence des noms des fichiers, des API, des protocoles, des sockets, des commandes, etc. ^{[7] [8] [9]}

Wikipedia a franchi le pas en décembre 2012 en adoptant en lieu et place d'une version modifiée de MySQL, la solution MariaDB sans aucune modification ^{[7] [10]}. D'autres sociétés ont également été séduites comme Google ^[11] et Mozilla ^[12]. De nombreuses distributions GNU/Linux comme Red Hat ^[13], Fedora et openSUSE ^[14] ont aussi franchi le pas.

MariaDB est aussi multiplateforme. ^{[15] [16]}

- *Oracle Database*

Oracle Database est un SGBDR propriétaire fourni par Oracle Corporation. Depuis l'introduction du support du modèle objet dans la version 8, il peut aussi être qualifié de système de gestion de bases de données relationnelles-objet (SGBDRO). ^[17]

Il est surtout ciblé pour la gestion de larges bases de données en entreprise ^[18]. Ce qui permet aussi à Oracle de continuer à proposer MySQL pour la gestion de bases de données Web ^[5].

C'est un outil multiplateforme ^[19].

- *Microsoft SQL Server*

Microsoft SQL Server est un SGBDR développé et commercialisé par la société Microsoft. La solution Microsoft Access, fournie dans la suite Office, gérant des bases de données qui n'excèdent pas le gigaoctet ^[1], SQL Server est destiné à gérer de plus larges bases de données mais si la base de données devient encore plus grande et que le nombre de clients augmente, Microsoft propose de passer à SQL Azure ^[20], une solution de Cloud Computing qui ne nous intéresse pas dans ce mémoire ^[21].

Ce SGBD n'est disponible que sous les plateformes Windows et avec une licence propriétaire ^[22].

- *PostgreSQL*

PostgreSQL est un SGBDR et un SGBDRO ^[23]. C'est un outil libre disponible selon les termes d'une licence de type BSD ou MIT qui autorise les utilisateurs à faire ce qu'ils veulent avec le code source à condition qu'ils mentionnent les droits d'auteur dans chaque copie du logiciel ^{[24] [25]}.

Il est beaucoup utilisé dans les bases de données web du fait de sa capacité à gérer de larges volumes. Parmi les solutions qui l'utilisent, nous pouvons citer Yahoo! ^[26], Sony Online ^[27] ou encore Instagram ^[28].

Il fonctionne sur tous les systèmes d'exploitation majeurs incluant GNU/Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), et Windows ^[23].

- *SQLite*

Contrairement aux systèmes de gestion de bases de données traditionnels, SQLite ne reproduit pas le schéma habituel « Client - Serveur » mais il est directement intégré aux programmes. L'intégralité de la base de données est stockée dans un fichier indépendant. SQLite est en fait une bibliothèque écrite en C qui propose un moteur de bases de données transactionnelles accessible par le langage SQL.

SQLite n'a pas de licence, il est dans le domaine public, ce qui rend cette solution particulièrement utilisée dans de nombreux projets qu'ils soient libres ou propriétaires tels que les systèmes embarqués, les navigateurs web ou les systèmes d'exploitation mobiles que sont iOS et Android. ^{[29] [30] [31] [32] [33]}

b. Choix du Système de gestion de bases de données

Voulant élaborer une application multiplateforme, fonctionnant principalement sur Windows et GNU/Linux, et utilisant un gestionnaire de bases de données gratuit, libre et travaillant en mode « Client – Serveur », notre choix s'est dirigé vers la solution MariaDB.

Pour les besoins de l'implémentation, ce SGBD est disponible en téléchargement pour Windows sur le site officiel ^[34] et pour GNU/Linux Debian Wheezy en ajoutant le dépôt officiel via les démarches décrites sur le site ^[35].

c. Oracle MySQL Workbench

L'interface en mode ligne de commande (CLI) est la seule interface possible avec MariaDB à son installation qui ne soit pas un outil tiers. MySQL Workbench est un outil à interface graphique (GUI) pour les architectes de bases de données, les développeurs et les administrateurs de bases de données. De plus, il permet la modélisation des données, le développement SQL et propose des outils d'administration complets pour la configuration de serveurs, la gestion des utilisateurs, la sauvegarde, etc. [36]

C'est un outil conçu pour travailler dans un environnement MySQL mais il est tout à fait possible d'importer des bases de données d'autres types tels que Microsoft SQL Server ou PostgreSQL via son service de migration de bases de données [37].

MySQL Workbench est édité par Oracle Corporation. Il est disponible sous Windows, GNU/Linux et Mac OS X [38]. La version Community Edition est disponible sous licence GPL et les versions Standard Edition et Enterprise Edition sont disponibles sous une licence commerciale propriétaire [39].

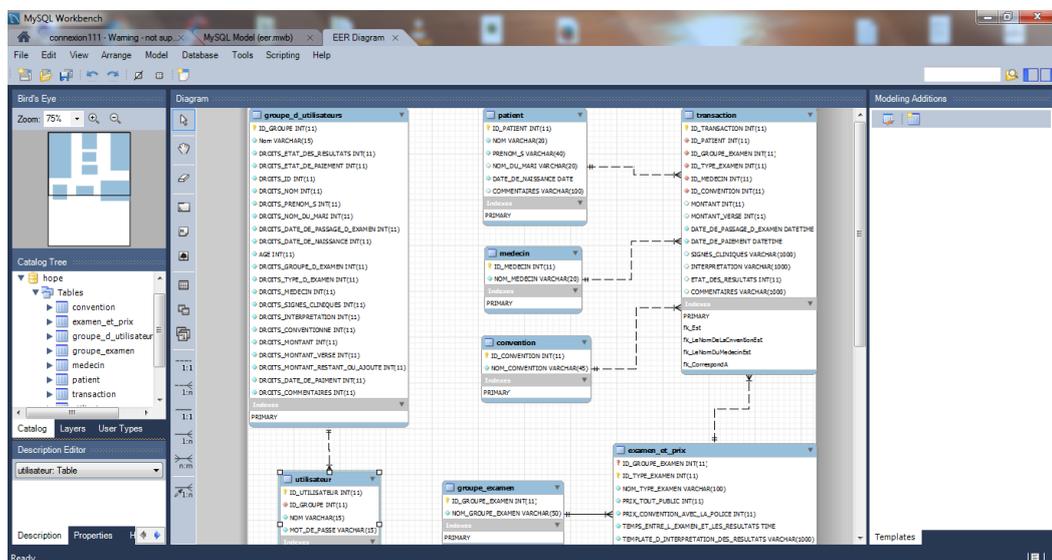


Figure II.1 Capture d'écran de SQL Workbench

II.2. Le framework Qt

a. Présentation

Qt (prononcé en anglais cute (/kju:t/), est un framework d'applications et d'interfaces utilisateur multiplateforme, orientée objet et développée en C++ par Qt Development Frameworks, filiale de Digia. Dans sa conception fondamentale, Qt offre des composants d'interface graphique (widgets), mais pas seulement, il offre aussi des modules de connexions réseaux, de script pour le JavaScript, d'analyse XML, d'accès aux bases de données, etc. ^[40] ^[41]

Il permet la portabilité des applications en fournissant des outils conçus pour simplifier la création d'applications et d'interfaces utilisateur pour le bureau, l'embarqué, et les plateformes mobiles. Avec Qt, on peut réutiliser le code efficacement pour cibler plusieurs plateformes avec un seul code de base ^[42].

Les environnements officiellement supportés sont :

- Pour les plateformes bureau :
 - GNU/Linux avec X11 ;
 - Windows ;
 - Mac OS X.
- Pour les plateformes de l'embarqué :
 - Embedded Android ;
 - Embedded Linux ;
 - Windows Embedded (Compact et Standard).
- Pour les plateformes mobiles :
 - Android ;
 - iOS ;
 - BlackBerry 10 ;
 - Sailfish OS ;
 - WinRT (Windows Phone inclus) ;
 - Tizen. ^[43]

Qt utilise directement les API de bas niveau des plateformes supportées, ce qui rend ce framework vraiment flexible et efficace, à l'opposé des autres outils qui utilisent un

schéma à plusieurs couches tels que Motif sous X11 qui souffre parfois de ralentissement à cause des nombreux appels de fonctions d'une couche à l'autre pour parvenir à ses fins [40].

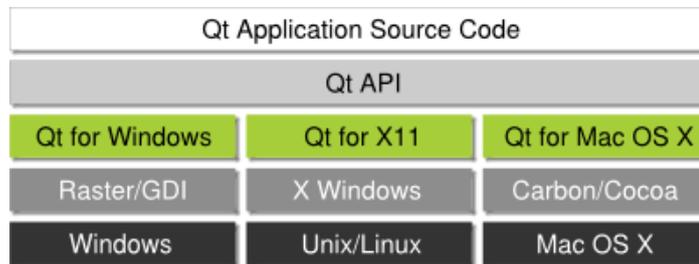


Figure II.2 Schéma montant les différents composants de bureau des principaux systèmes d'exploitation que Qt utilise [40]

Par défaut, Qt adapte le thème de la fenêtre et de ses composants à celui du système d'exploitation sur lequel l'application s'exécute, en tenant compte des directives d'IHM de chaque système d'exploitation. Grâce à cette méthode, les widgets codés ont une apparence adaptée à chaque plateforme [40] [41].

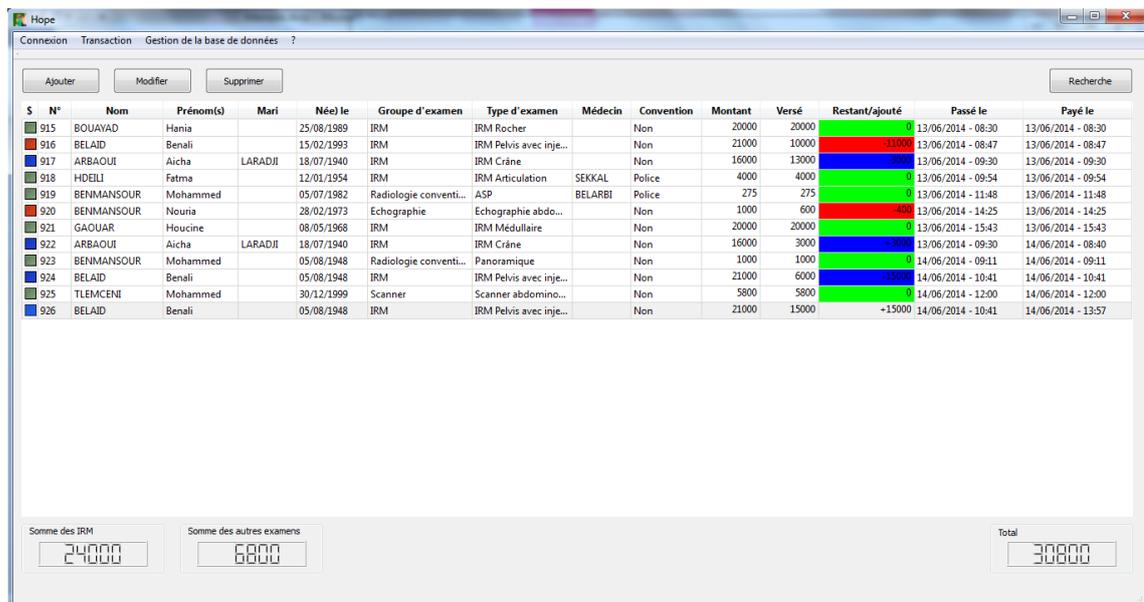
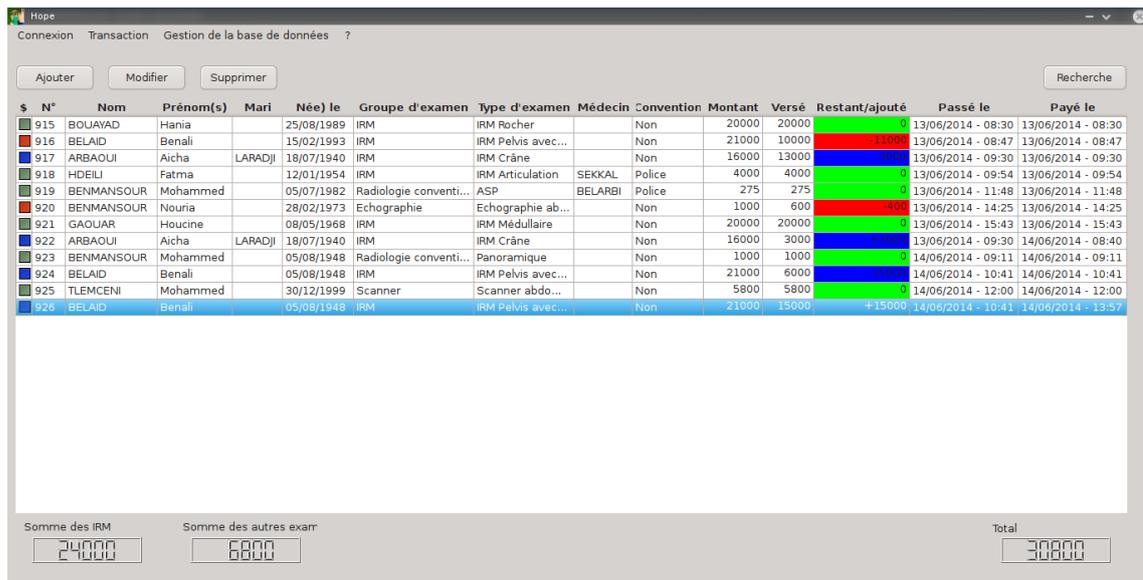


Figure II.3 Apparence de l'application réalisée pour le mémoire sous Windows



\$	N°	Nom	Prénom(s)	Mari	Née le	Groupe d'examen	Type d'examen	Médecin	Convention	Montant	Versé	Restant/ajouté	Passé le	Payé le
915	BOUAYAD	Hania			25/08/1989	IRM	IRM Rocher		Non	20000	20000	0	13/06/2014 - 08:30	13/06/2014 - 08:30
916	BELAID	Benali			15/02/1993	IRM	IRM Pelvis avec...		Non	21000	10000	-11000	13/06/2014 - 08:47	13/06/2014 - 08:47
917	ARBAOUI	Aicha	LARADJI		18/07/1940	IRM	IRM Crâne		Non	16000	13000	-3000	13/06/2014 - 09:30	13/06/2014 - 09:30
918	HDEILI	Fatma			12/01/1954	IRM	IRM Articulation	SEKKAL	Police	4000	4000	0	13/06/2014 - 09:54	13/06/2014 - 09:54
919	BENMANSOUR	Mohammed			05/07/1982	Radiologie conventi...	ASP	BELARBI	Police	275	275	0	13/06/2014 - 11:48	13/06/2014 - 11:48
920	BENMANSOUR	Nouria			28/02/1973	Echographie	Echographie ab...		Non	1000	600	-400	13/06/2014 - 14:25	13/06/2014 - 14:25
921	GAOUAR	Houcine			08/05/1968	IRM	IRM Médullaire		Non	20000	20000	0	13/06/2014 - 15:43	13/06/2014 - 15:43
922	ARBAOUI	Aicha	LARADJI		18/07/1940	IRM	IRM Crâne		Non	16000	3000	-13000	13/06/2014 - 09:30	14/06/2014 - 08:40
923	BENMANSOUR	Mohammed			05/08/1948	Radiologie conventi...	Panoramique		Non	1000	1000	0	14/06/2014 - 09:11	14/06/2014 - 09:11
924	BELAID	Benali			05/08/1948	IRM	IRM Pelvis avec...		Non	21000	6000	-15000	14/06/2014 - 10:41	14/06/2014 - 10:41
925	TLEMCENI	Mohammed			30/12/1999	Scanner	Scanner abdo...		Non	5800	5800	0	14/06/2014 - 12:00	14/06/2014 - 12:00
926	BELAID	Benali			05/08/1948	IRM	IRM Pelvis avec...		Non	21000	15000	+15000	14/06/2014 - 10:41	14/06/2014 - 13:57

Somme des IRM: 24000 Somme des autres exam: 6000 Total: 30000

Figure II.4 Apparence de l'application réalisée pour le mémoire sous GNU/Linux sous l'environnement de bureau KDE

De plus, Qt supporte des bindings avec plusieurs langages autres que le C++, comme Python, Ruby, Java, JavaScript, BASIC, Ada 2005, C#, Pascal, Lua, Perl. La plupart de ces bindings sont produits et maintenus par des équipes de développeurs open source [40].

Qt est notamment connu pour être l'API sur laquelle repose l'environnement de bureau KDE, l'un des environnements de bureau les plus utilisés dans le monde GNU/Linux [44]. D'autres entreprises et organismes comme Adobe, Google, IBM, Skype ou encore la NASA utilisent aussi ce framework pour l'élaboration de leurs produits.

Il est livré avec les différents outils de développement :

- **Qt Assistant :**

A l'instar de la javadoc, il permet de visualiser la documentation complète de Qt en mode hors ligne.

- **Qt Designer :**

C'est un outil graphique de conception de GUI, il permet d'accélérer le développement d'application ou de diviser le travail en deux en réservant cet outil aux concepteurs de GUI et le l'implémentation du code aux développeurs. Il enregistre les fichiers avec l'extension .ui.

- **Qt Linguist :**

C'est un outil dédié aux traducteurs, il leur permet d'entrer leur traduction dans les fichiers avec l'extension .ts pour créer des correspondances entre les différentes langues utilisées dans l'application développée.

- **Qt Creator :**

C'est l'IDE dédié à Qt. Son éditeur de texte offre les principales fonctions que sont la coloration syntaxique, le complètement automatique, l'indentation, etc. Qt Creator intègre en son sein les outils Qt Designer et Qt Assistant. Une explication plus complète de cet outil est donnée plus bas.

- **Qt Demo :**

Il présente un ensemble de petits programmes (sous forme d'exécutables et de sources) qui montrent les fonctionnalités offertes par le framework.

- **qmake :**

Il prend en entrée un fichier (avec l'extension .pro) décrivant le projet (liste des fichiers source, dépendances, paramètres passés au compilateur, etc.) et génère un fichier de projet spécifique à la plateforme. Ainsi, qmake produit un Makefile qui contient la liste des commandes à exécuter pour la génération d'un exécutable, à l'exception des étapes spécifiques à Qt (génération des classes C++ lors de la conception d'interface graphique avec Qt Designer, génération du code C++ pour lier les signaux et les slots avec le Meta Object Compiler, etc.)

Le fichier de projet est fait pour être très facilement éditable par un développeur. Il consiste en une série d'affectations de variables.

En voici un exemple pour un petit projet:

```
TARGET    = monApplication
SOURCES   += main.cpp\
          mainwindow.cpp
HEADERS   = mainwindow.h
FORMS     = mainwindow.ui
QT        += sql
```

Ces déclarations demandent que l'exécutable soit nommé "monApplication", donne la liste des fichiers source, en-têtes et fichiers d'interface graphique. La dernière ligne déclare que le projet requiert le module SQL de Qt. ^[40]

Le framework Qt et ses outils sont développés en tant que projet libre/open source gouverné par un modèle méritocratique. Il peut être utilisé sous une licence libre (GPL v3 et LGPL v2.1) ou sous une licence commerciale propriétaire ^[45].

Les produits de la licence propriétaire, Qt Entreprise et Qt Mobile, contiennent des fonctionnalités et des outils de productivité supplémentaires ^[46].

b. Qt Creator

Qt Creator est un IDE qui fournit des outils pour concevoir et développer des applications et des interfaces utilisateur avec le framework Qt et leur déploiement à travers différents systèmes d'exploitation bureau, embarqués et mobile à travers tout un cycle de vie de développement, de la création du projet au déploiement sur les plateformes cibles ^[47].

Pour le compilateur, avec l'installation par défaut,

- Sous les environnements UNIX (GNU/Linux et Mac OS X), il utilise G++, le compilateur C++ de la GNU Compiler Collection (GCC) ;
- Sous Windows, il peut utiliser MinGW, la version Windows de GCC, ou MSVC, le compilateur de Microsoft Visual C++ ^[48].

Pour le débogueur, Qt Creator utilise par défaut GDB (GNU Symbolic Debugger) ^[49].

Mais il reste tout à fait possible de changer de compilateur et/ou de débogueur dans les paramètres de configuration pour utiliser Clang comme compilateur, par exemple ^[48].

Il est disponible sous Windows, GNU/Linux et Mac OS X ^[50].

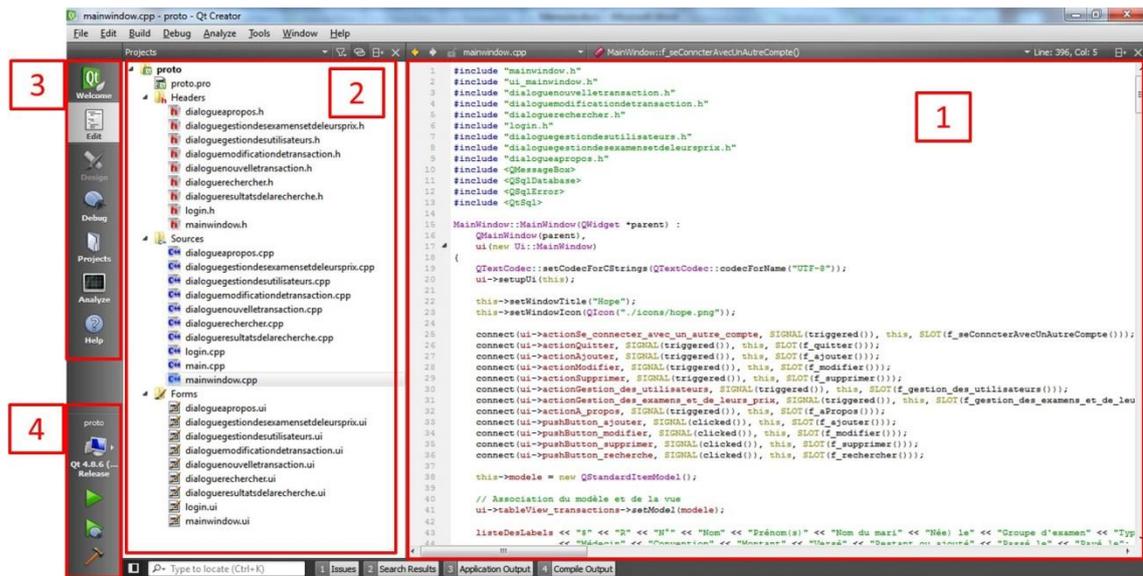


Figure II.5 Principaux éléments de Qt Creator

L'interface se présente principalement comme suit :

1. L'éditeur de texte

Il permet principalement : la coloration syntaxique ; la complémentation de code ; la recherche d'expressions ; à défaut d'avoir le système d'onglets, il permet le scindement en plusieurs parties pour gérer plusieurs fichiers source en même temps ; l'explication de chaque classe ou méthode en positionnant le curseur dessus puis en appuyant sur "F1".

2. La barre des fichiers du(des) projet(s) ouvert(s)

Dans un projet Qt, il existe 4 types de fichiers :

- Le fichier .pro : c'est le fichier de configuration du projet. Ce sont les directives que devra suivre qmake pour la compilation du projet.
- Les fichiers .h : ce sont les habituels fichiers de headers des projets C++ ;
- Les fichiers .cpp : ce sont les habituels fichiers source des projets C++ ;
- Les fichiers .ui : Comme ces initiales l'indiquent, UI pour User Interface, ils représentent les fichiers de gestion des interfaces utilisateur. Ils sont au format XML. Pour être utilisés dans

l'application, ils peuvent être traités de deux façons :

- Au moment de la compilation : ils sont convertis via les outils qmake et uic en code C++ qui peut être compilé. Exemple : le fichier "fenetre.ui" est converti en "ui_fenetre.h".
- Au moment de l'exécution : ils sont traités par la classe QUiLoader qui construit dynamiquement l'arbre de widgets en parcourant les fichiers XML. [51]

3. La barre des options

Elle contient 7 éléments :

- Welcome : c'est la page d'accueil de l'IDE. Elle permet d'ouvrir directement les projets récemment ouverts.
- Edit : permet d'afficher l'éditeur de texte pour éditer le code source.
- Design : permet de lancer le concepteur d'interfaces utilisateur si c'est un fichier .ui. C'est en fait Qt Designer qui est entièrement intégré dans Qt Creator.

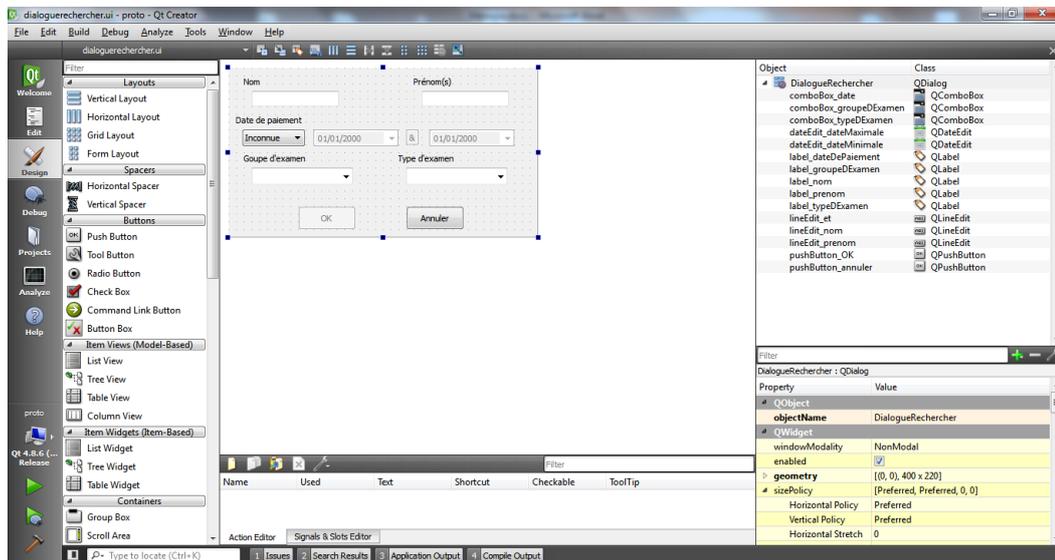


Figure II.6 Le mode Design dans Qt Creator

- Debug : permet d'inspecter l'état de l'application en débogant.
- Projects : permet de gérer les paramètres de compilation et

d'exécution du projet. Il n'est disponible que pour les fichiers .pro.

- Analyze : permet d'utiliser les outils d'analyse de code pour détecter les fuites de mémoire et profiler le code C++.
- Help : permet d'accéder à la documentation de Qt en mode hors ligne. ^[52]

4. La barre des outils de compilation

Elle contient 4 éléments :

- Le sélecteur de la cible

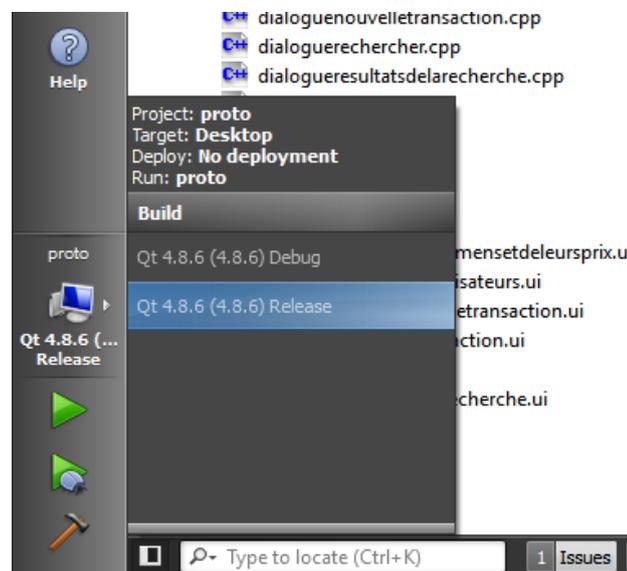


Figure II.7 Choix du mode de compilation

Il permet de sélectionner le mode de compilation : soit "Debug", soit "Release". Pour faire les tests, "Debug" est adapté. Les DLL à joindre changent en fonction du mode de compilation utilisé. En mode Release, les DLL sont beaucoup plus légères qu'en mode Debug (une DLL "Release" fait 3-4 Mo tandis qu'une DLL "Debug" fait parfois plus de 100 Mo) ^[53].

- Exécution
- Débogage
- Compilation

CHAPITRE III :

Réalisation de l'application

III. Chapitre III : Réalisation de l'application

III.1. Conception de la base de données

Notre base de données est composée de 8 tables.

Le schéma suivant présente le modèle de données que nous avons élaboré lors de la conception de notre application :

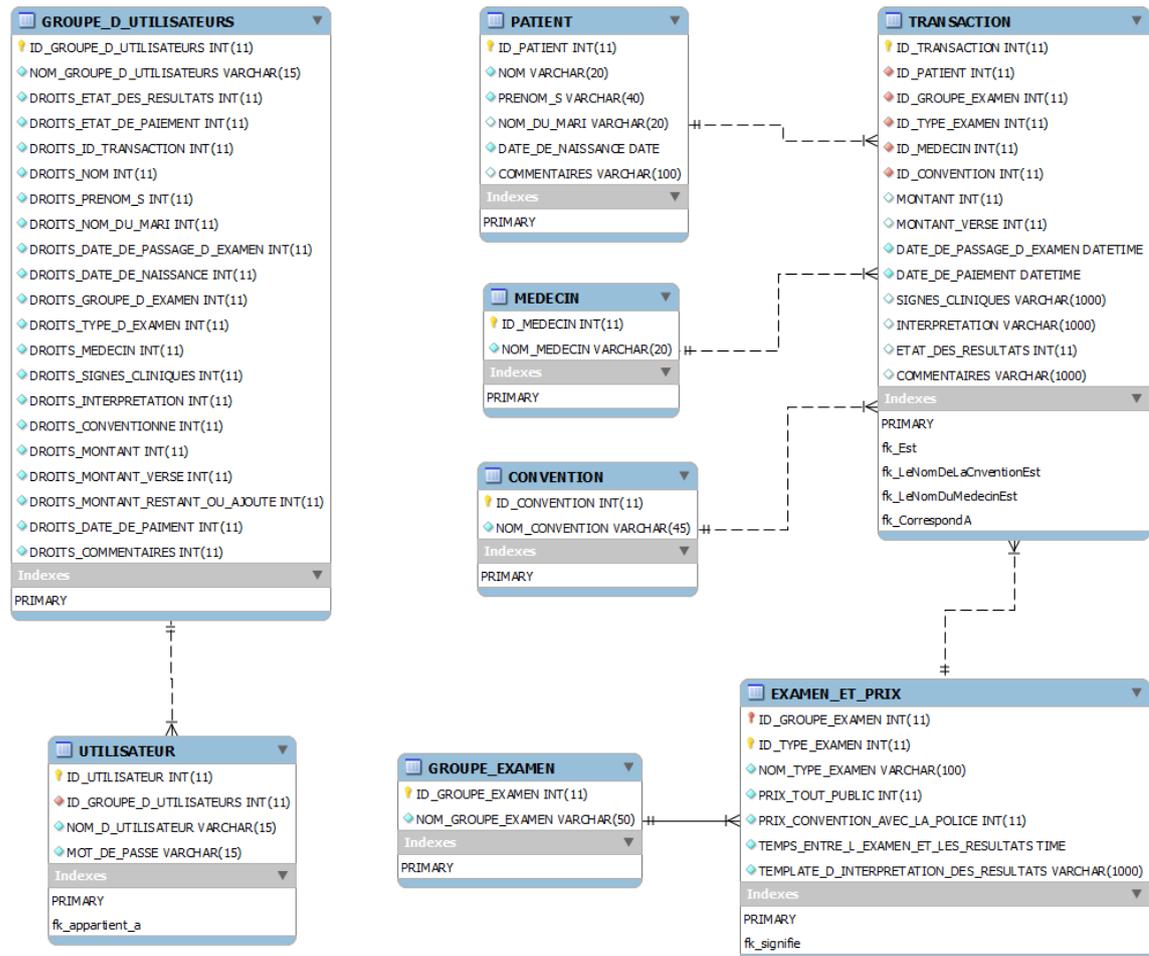


Figure III.1 Modèle relationnel de la base de données (obtenu par MySQL Workbench)

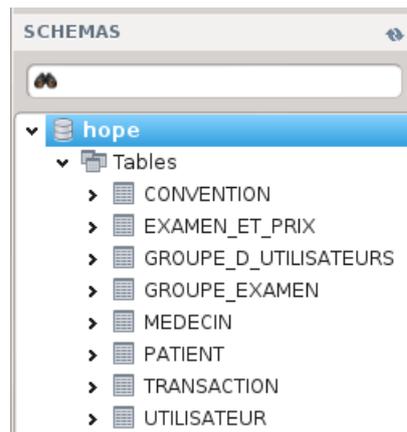


Figure III.2 Liste des tables de la base de données

La table « TRANSACTION »

C'est la table principale de notre diagramme.

Elle contient une clé primaire "ID_TRANSACTION", un entier permettant d'identifier la transaction courante.

Les autres champs sont :

- "ID_PATIENT" : un entier servant comme clé étrangère de la table « PATIENT » pour avoir les informations sur le patient de la transaction courante ;
- "ID_GROUPE_EXAMEN" et "ID_TYPE_EXAMEN" : deux entiers servant comme un couple de clés étrangères de la table « EXAMEN_ET_PRIX » pour identifier le groupe et le type d'examen compté dans la transaction courante ;
- "ID_MEDECIN" : un entier servant comme clé étrangère de la table « MEDECIN » pour l'identification du nom du médecin ;
- "ID_CONVENTION" : un entier servant comme clé étrangère de la table « CONVENTION » pour identifier le type de convention ;
- "MONTANT" et "MONTANT_VERSE" : deux entiers servant à enregistrer le montant et le montant versé du patient pour la transaction courante ;
- "DATE_DE_PASSAGE_D_EXAMEN" : de type DATETIME pour enregistrer la date et l'heure de passage d'examen ;

- "DATE_DE_PAIEMENT" : de type DATETIME pour enregistrer la date et l'heure de règlement de l'examen ;
- "SIGNES_CLINIQUES" : une chaîne de caractères pour enregistrer les signes cliniques du patient lors du passage de l'examen ;
- "INTERPRETATION" : une chaîne de caractères contenant l'interprétation des résultats de l'examen faite par le médecin ;
- "COMMENTAIRES" : une chaîne de caractères contenant les commentaires faits sur la transaction courante.

La table « PATIENT »

Cette table contient les informations sur le patient et a pour identifiant la clé primaire "ID_PATIENT".

Elle contient aussi les champs suivant :

- "NOM" : une chaîne de caractère pour le nom du patient ;
- "PRENOM_S" : une chaîne de caractère pour le ou les prénoms ;
- "NOM_DU_MARI" : une chaîne de caractère pour le nom du mari si le patient est une femme mariée ;
- "DATE_DE_NAISSANCE" : de type DATE pour enregistrer la date de naissance du patient ;
- "COMMENTAIRES" : une chaîne de caractères pour les commentaires sur le patient.

La table « MEDECIN »

Cette table contient le nom du médecin et a pour identifiant la clé primaire "ID_MEDECIN".

Elle a aussi le champ suivant :

- "NOM_MEDECIN" : une chaîne de caractères contenant le nom du médecin.

La table « CONVENTION »

Cette table contient le nom de la convention et a pour identifiant la clé primaire "ID_CONVENTION".

Elle a aussi le champ suivant :

- "NOM_CONVENTION" : une chaîne de caractères contenant le nom de la convention.

La table « EXAMEN_ET_PRIX »

Cette table contient pour chaque couple de groupe et type d'examen, les informations relatives à ce couple. Elle a donc comme clé primaire un couple de clés primaires qui est "ID_GROUPE_EXAMEN" et "ID_TYPE_EXAMEN", deux entiers représentant respectivement l'ID du groupe et l'ID du type d'examen.

Elle contient aussi les champs suivants :

- "NOM_TYPE_EXAMEN" : une chaîne de caractère donnant le nom du type d'examen.

- "PRIX_TOUT_PUBLIC" : un entier représentant le prix sans convention de l'examen courant.

- "PRIX_CONVENTION_AVEC_LA_POLICE" : un entier représentant le prix de l'examen courant dans le cas de la convention avec la police.

- "TEMPS_ENTRE_L_EXAMEN_ET_LES_RESULTATS" : de type TIME, il permet d'afficher le temps que le patient devra attendre entre son passage d'examen et la disponibilité des résultats.

- "TEMPLATE_D_INTERPRETATION_DES_RESULTATS" : une chaîne de caractères contenant un modèle d'interprétation des résultats pour l'examen courant. La secrétaire n'a ainsi plus qu'à modifier les caractères voulu pour le patient ciblé.

Il est à noter que "ID_GROUPE_EXAMEN" est aussi une clé étrangère de la table « GROUPE_EXAMEN ».

La table « GROUPE_EXAMEN »

Cette table contient le nom du groupe d'examens et a pour identifiant la clé primaire "ID_GROUPE_EXAMEN".

Elle a aussi le champ suivant :

- "NOM_GROUPE_EXAMEN" : une chaîne de caractères représentant le nom du groupe d'examens.

La table « UTILISATEURS »

Cette table contient les informations sur l'utilisateur et a pour identifiant la clé primaire "ID_UTILISATEUR".

Elle contient aussi les champs suivant :

- "ID_GROUPE_D_UTILISATEURS" : un entier représentant une clé étrangère de la table « GROUPE_D_UTILISATEURS ».

- "NOM_D_UTILISATEUR" : une chaîne de caractère pour le nom de l'utilisateur.

- "MOT_DE_PASSE" : une chaîne de caractère pour le mot de passe de l'utilisateur.

La table « GROUPE_D_UTILISATEURS »

Cette table contient le nom du groupe d'utilisateur et les droits d'utilisation du groupe et a pour identifiant la clé primaire "ID_GROUPE_D_UTILISATEURS".

Elle contient aussi les champs suivants :

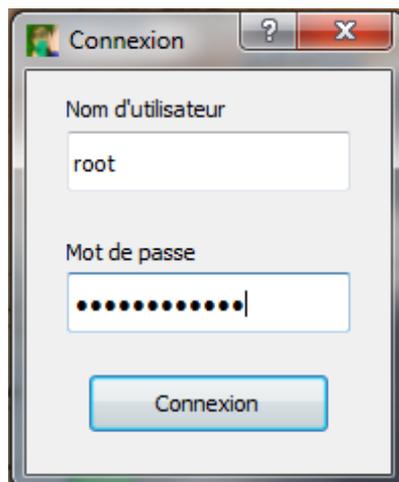
- "NOM_GROUPE_D_UTILISATEURS" : une chaîne de caractères représentant le nom du groupe d'utilisateurs

- les champs des droits sur chaque élément qui sont des entiers, par exemple : le champ "DROITS_ETAT_DE_PAIEMENT", s'il est égal à 0 l'état de paiement est invisible pour le groupe, s'il est égal à 1 il est en lecture seule, et s'il est égal à 2 il est disponible en lecture et écriture.

III.2. Interface utilisateur de l'application

Au lancement de l'application, l'utilisateur doit taper son nom d'utilisateur et son mot de passe.

Pour le moment, l'application se concentre sur les utilisateurs gérés par le SGBD et non les utilisateurs de la table « UTILISATEUR ». Et le seul utilisateur créé à l'installation de MariaDB est le "root".



Donc, si le nom d'utilisateur correspond à "root" et que le mot de passe soit celui du "root", la fenêtre principale s'ouvre.

S	N°	Nom	Prénom(s)	Mari	Née le	Groupe d'examen	Type d'examen	Médecin	Convention	Montant	Versé	Restant/ajouté	Passé le	Payé le
915		BOUAYAD	Hania		25/08/1989	IRM	IRM Rocher		Non	20000	20000		13/06/2014 - 08:30	13/06/2014 - 08:30
916		BELAID	Benali		15/02/1993	IRM	IRM Pelvis avec inje...		Non	21000	10000	-11000	13/06/2014 - 08:47	13/06/2014 - 08:47
917		ARBAOUI	Aicha	LARADJI	18/07/1940	IRM	IRM Crâne		Non	16000	13000	-3000	13/06/2014 - 09:30	13/06/2014 - 09:30
918		HDELLI	Fatma		12/01/1954	IRM	IRM Articulation	SEKAL	Police	4000	4000		13/06/2014 - 09:54	13/06/2014 - 09:54
919		BENMANSOUR	Mohammed		05/07/1982	Radiologie conventi...	ASP	BELARBI	Police	275	275		13/06/2014 - 11:48	13/06/2014 - 11:48
920		BENMANSOUR	Nouria		28/02/1973	Echographie	Echographie abdo...		Non	1000	600	-400	13/06/2014 - 14:25	13/06/2014 - 14:25
921		GAOUAR	Houcine		08/05/1968	IRM	IRM Médullaire		Non	20000	20000		13/06/2014 - 15:43	13/06/2014 - 15:43
922		ARBAOUI	Aicha	LARADJI	18/07/1940	IRM	IRM Crâne		Non	16000	3000	-13000	13/06/2014 - 09:30	14/06/2014 - 08:40
923		BENMANSOUR	Mohammed		05/08/1948	Radiologie conventi...	Panoramique		Non	1000	1000		14/06/2014 - 09:11	14/06/2014 - 09:11
924		BELAID	Benali		05/08/1948	IRM	IRM Pelvis avec inje...		Non	21000	6000	-15000	14/06/2014 - 10:41	14/06/2014 - 10:41
925		TLEMCENI	Mohammed		30/12/1999	Scanner	Scanner abdomino...		Non	5800	5800		14/06/2014 - 12:00	14/06/2014 - 12:00
926		BELAID	Benali		05/08/1948	IRM	IRM Pelvis avec inje...		Non	21000	15000	+15000	14/06/2014 - 10:41	14/06/2014 - 13:57

Somme des IRM: 24000
Somme des autres examens: 6000
Total: 30000

Il convient ici de parler des codes couleur utilisés. Ils représentent les différents états de paiement. Il existe 3 états :

- **Transaction payée en une fois**

Exemple :

915	BOUAYAD	Hania	IRM Rocher	20000	20000	0
-----	---------	-------	------------	-------	-------	---

La patiente BOUAYAD Hania a payé la totalité du prix de son examen en un seul coup. Il lui reste donc 0 DA.

La transaction est en état de "transaction payée en une fois" et porte la couleur verte.

Le "montant restant ou ajouté" est égal à 0.

- **Transaction partiellement payée**

Exemple :

924	BELAID	Benali	IRM Pelvis avec injection	21000	6000	-15000
-----	--------	--------	---------------------------	-------	------	--------

Le patient BELAID Benali n'a versé que 6.000 DA sur les 21.000 DA que compte son examen, il lui reste donc 15.000 DA.

La transaction est en état de "transaction partiellement payée" et porte la couleur rouge.

Le "montant restant ou ajouté" est un montant restant et a une valeur négative.

- **Transaction complètement payée**

Exemple :

926	BELAID	Benali	IRM Pelvis avec injection	21000	15000	+15000
-----	--------	--------	---------------------------	-------	-------	--------

Si maintenant le patient précédent verse le montant qui restait, c'est-à-dire les 15.000 DA, la nouvelle transaction prend la couleur bleue, cette transaction est en état "transaction complètement payée".

Le "montant restant ou ajouté" est un montant ajouté et a une valeur positive.

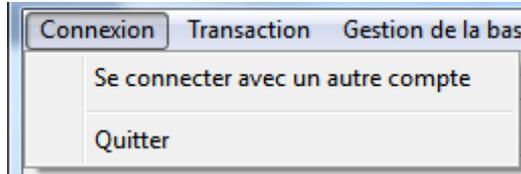
La transaction précédente, la n° 924, devient aussi en état de "transaction complètement payée" mais la valeur du "montant restant ou ajouté" demeure négative pour montrer que ce n'est pas cette transaction qui a complété le paiement mais bien la n° 926.

924	BELAID	Benali	IRM Pelvis avec injection	21000	6000	-15000
-----	--------	--------	---------------------------	-------	------	--------

Sur la fenêtre principale, sont présents :

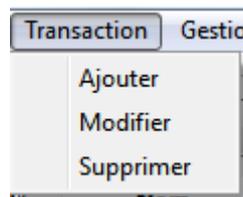
1. Sur la barre d'outils :

- Connexion



- "Se connecter avec un autre compte" permet de relancer la fenêtre de login pour se connecter avec un autre compte justement.
- "Quitter" permet de quitter l'application, ce qui est équivalent à appuyer sur "ALT + F4" ou à fermer la fenêtre de l'application.

- Transaction



- "Ajouter" permet d'ajouter une nouvelle transaction en ouvrant la fenêtre suivante :

Nouvelle transaction

Informations sur le patient | Commentaires sur la transaction

Nom:

Prénom:

Nom du mari:

Date de passage d'examen:

Date de naissance:

Âge:

Groupe d'examen:

Type d'examen:

Nom du médecin:

Montant:

Conventionné:

Montant versé:

Montant restant ou ajouté:

Date de paiement:

OK | Annuler

- "Modifier" permet de modifier la transaction sélectionnée en ouvrant une fenêtre semblable à celle de "Ajouter" mais en tenant compte de la transaction courante :

- "Supprimer" permet de supprimer la transaction sélectionnée en affichant une boîte de dialogue demandant de confirmer la suppression ou de l'annuler comme suit :

- Gestion de la base de données

- "Gestion des utilisateurs" permettra dans la version finale d'afficher et de gérer les utilisateurs, leurs mots de passe et le groupe auquel chacun appartient. Pour le moment, elle ne permet que d'afficher les

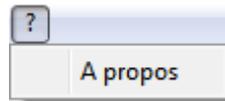
noms des utilisateurs, l'ID de chacun et le nom du groupe auquel chacun appartient.

N°	Groupe	Nom
0	Administrateur	Mohammed1
1	Réceptionniste	Zakia
2	Manipulateur	Mohammed2
3	Manipulateur	Soumia
4	Caissier	Abdelkarim
5	Secrétaire	Leila

- "Gestion des examens et de leurs prix" permettra elle aussi dans la version finale de gérer les types d'examens, le groupe de chacun et le prix de chacun selon chaque convention. Pour le moment, elle n'affiche que les types d'examens et leurs prix selon le groupe choisi.

Type d'examen	Prix tout public	Prix police
Téléthorax de face	1100	275
Téléthorax de face + profil	1300	325
Gril costal	1000	250
Epaule: face + profil + 3/4	1600	400
Sternum profil	1000	250
Clavicule face	900	225
ASP	1100	275
Crâne de face	1000	250
Crâne de face + profil	1300	325
Schuller bilatéral	1300	325
Os propre du nez	1000	250
Selle turcique de face + profil	1400	350
N.M.P	1000	250
N.F.P	1000	250
N.M.P + N.F.P	1400	350
Rachis cervical	900	225
Rachis cervical de face + profil	1200	300
Rachis cervical de face + profil / 1 3/4	1600	400
Rachis cervical de face + profil / 2 3/4	1800	450

- "?"



- "A propos" qui affiche une simple fenêtre sur l'auteur et les éventuelles contributions.



2. Les boutons :

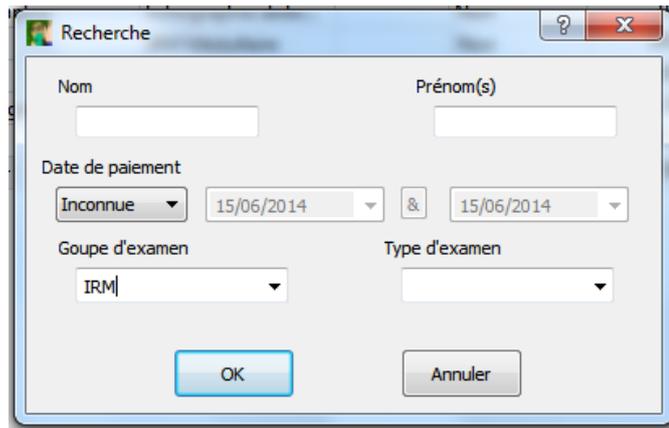


Les boutons sont au nombre de 3 à gauche + 1 à droite.

Le rôle de chacun des trois premiers est le même que celui des options de la barre d'outils.

Celui du bouton "Recherche" est nouveau par contre. Il permet de faire apparaître une petite fenêtre afin d'y énoncer les informations sur les transactions à rechercher.

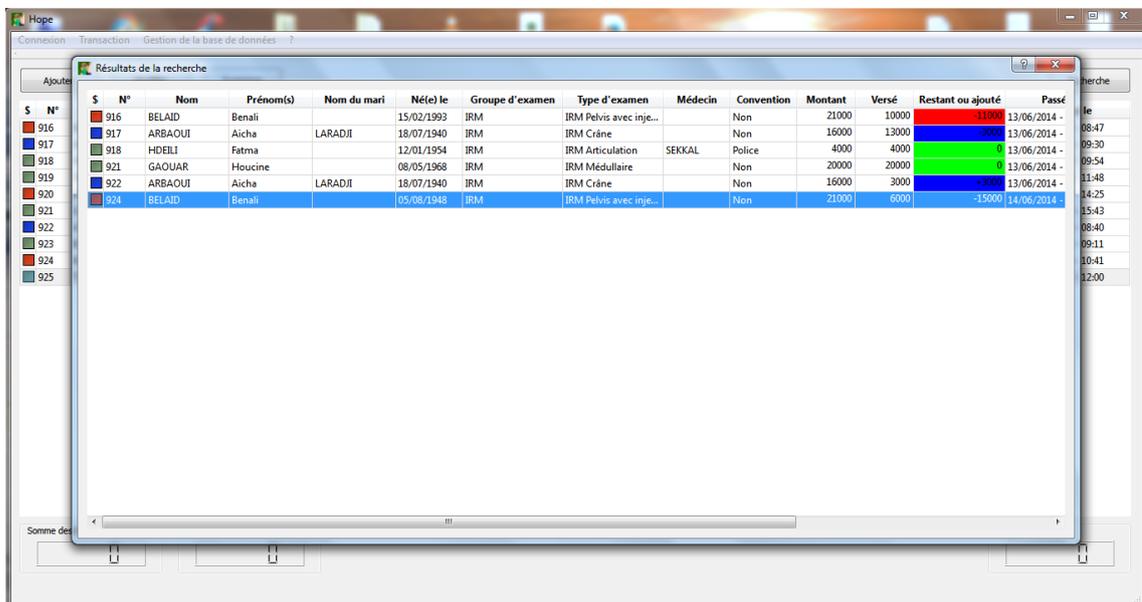
La fenêtre de recherche se présente comme suit :



Les champs disponibles pour la recherche sont :

- Le nom ;
- Le(s) prénom(s) ;
- La date de paiement ou l'intervalle de la date de paiement ;
- Le nom du groupe d'examen ;
- Le nom du type d'examen.

En cliquant sur le bouton "OK", la recherche est effectuée et la fenêtre des résultats de la recherche apparaît.



Cette fenêtre contient un seul élément, c'est le même tableau que dans la fenêtre principale. Ce tableau est expliqué plus bas.

3. Le tableau :

S	N°	Nom	Prénom(s)	Mari	Née le	Groupe d'examen	Type d'examen	Médecin	Convention	Montant	Versé	Restant/ajouté	Passé le	Payé le
	915	BOUYAYAD	Henia		25/08/1989	IRM	IRM Rocher		Non	20000	20000	0	13/06/2014 - 08:30	13/06/2014 - 08:30
	916	BELAID	Benali		15/02/1993	IRM	IRM Pelvis avec injec...		Non	21000	10000	-11000	13/06/2014 - 08:47	13/06/2014 - 08:47
	917	ARBAOUI	Aicha	LARADJI	18/07/1940	IRM	IRM Crâne		Non	16000	13000	-3000	13/06/2014 - 09:30	13/06/2014 - 09:30
	918	HDELI	Fatma		12/01/1954	IRM	IRM Articulation	SEKKAL	Police	4000	4000	0	13/06/2014 - 09:54	13/06/2014 - 09:54
	919	BENMANSOUR	Mohammed		05/07/1982	Radiologie conventi...	ASP	BELARBI	Police	275	275	0	13/06/2014 - 11:48	13/06/2014 - 11:48
	920	BENMANSOUR	Nouria		28/02/1973	Echographie	Echographie abdo...		Non	1000	600	-400	13/06/2014 - 14:25	13/06/2014 - 14:25
	921	GAOUAR	Houcine		08/05/1968	IRM	IRM Médullaire		Non	20000	20000	0	13/06/2014 - 15:43	13/06/2014 - 15:43
	922	ARBAOUI	Aicha	LARADJI	18/07/1940	IRM	IRM Crâne		Non	16000	3000	-13000	13/06/2014 - 09:30	14/06/2014 - 08:40
	923	BENMANSOUR	Mohammed		05/08/1948	Radiologie conventi...	Panoramique		Non	1000	1000	0	14/06/2014 - 09:11	14/06/2014 - 09:11
	924	BELAID	Benali		05/08/1948	IRM	IRM Pelvis avec injec...		Non	21000	6000	-15000	14/06/2014 - 10:41	14/06/2014 - 10:41
	925	TLEMCENI	Mohammed		30/12/1999	Scanner	Scanner abdomino...		Non	5800	5800	0	14/06/2014 - 12:00	14/06/2014 - 12:00
	926	BELAID	Benali		05/08/1948	IRM	IRM Pelvis avec injec...		Non	21000	15000	-15000	14/06/2014 - 10:41	14/06/2014 - 13:57

Elément central de l'application, il permet de jeter un rapide coup d'œil pour détecter une transaction voulue.

Il permet aussi de savoir quelle transaction a été complètement payée et laquelle ne l'a été que partiellement. Il contient 15 colonnes.

- \$: l'état de paiement ;
- N° : le numéro de la transaction ;
- Nom, Prénom(s), Mari : les noms du patient ;
- Né(s) le : la date de naissance du patient ;
- Groupe d'examen et Type d'examen : le nom du groupe et du type d'examen respectivement ;
- Médecin : le nom du médecin figurant sur l'ordonnance du patient ;
- Convention : le type de convention utilisée par patient ;
- Montant : le montant compté pour l'examen pratiqué ;
- Versé : le montant versé par le patient dans cette transaction ;
- Restant/ajouté : Si la transaction est complètement payée en une fois ou qu'elle est partiellement payée et n'a pas encore été entièrement complétée, c'est un montant restant, sinon c'est un montant ajouté pour le cas où cette transaction permet de compléter entièrement le paiement de l'examen ;
- Passé le : la date et l'heure du passage d'examen ;
- Payé le : la date de paiement de l'examen ;

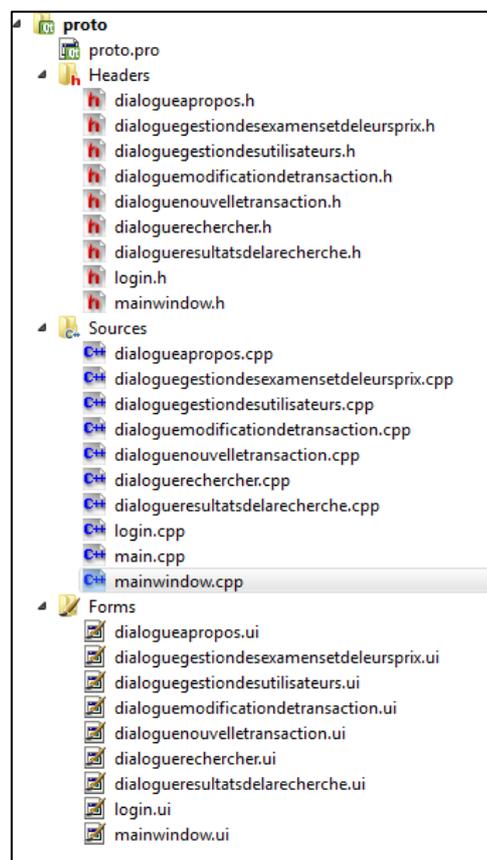
4. Les indicateurs des montants encaissés :

Chacun indique le montant encaissé du jour-même.



Les sommes se mettent à jour à chaque ajout, modification ou suppression de transaction.

III.3. Les différentes classes de l'application



L'application est composée de 9 classes chacune définie dans un fichier source (.cpp) :

1. Login

Elle hérite de la classe prédéfinie "QDialog". Elle permet de dessiner une petite boîte de dialogue à taille fixe qui permet de s'identifier pour pouvoir se connecter à la base de données. Elle a comme titre "Connexion".

La boîte de dialogue contient :

- **Nom d'utilisateur** : c'est un champ d'édition pour écrire le nom d'utilisateur ;
- **Mot de passe** : c'est un champ d'édition pour écrire le mot de passe et qui, donc, masque les caractères entrés ;
- **Un bouton "Annuler"** : toujours actif, qui une fois cliqué ferme l'application ;
- **Un bouton "OK"** : qui ne s'active que si les deux champs d'édition sont remplis.

Une fois le bouton cliqué, la classe instancie une connexion à la base de données en :

- Chargeant le pilote "QMYSQL" ;
- Initialisant l'adresse du serveur à "127.0.0.1" ;
- Initialisant le nom de la base de donnée à "hope" ;
- Initialisant le nom d'utilisateur et le mot de passe selon ce qui est inscrit dans les champs d'édition correspondants.

Si les données ne correspondent pas à ce qui est sur la base de données, une boîte de dialogue critique de type "QMessageBox" apparaît avec comme titre "Database Error", et ayant comme texte le message d'erreur envoyé par la base de donnée. Sinon, la boîte de dialogue se ferme et instancie la fenêtre principale "mainWindow" de la classe "MainWindow".

2. MainWindow

C'est une classe héritant de la classe prédéfinie "QMainWindow". Elle permet de dessiner une fenêtre contenant les différents éléments :

- **Une barre d'outils et quatre boutons**

Chacun des éléments de ces deux groupes, comme présentés dans la partie précédente, "Interface utilisateur de l'application", permet d'instancier la classe de la fenêtre adéquate, sauf pour l'élément "quitter" de l'option "Connexion" sur la barre d'outils qui permet de fermer la classe et donc de fermer l'application.

- **Un tableau**

Le tableau est rempli initialement avec les résultats d'une requête lancée au SGBD pour récupérer chaque transaction effectuée et les ordonne selon la date

de paiement de la plus ancienne en haut à la plus récente en bas, il dispose aussi d'une méthode permettant de calculer la différence entre le montant compté et le montant versé afin de déterminer l'état de chaque transaction et d'y adjoindre l'icône adéquate pour une bonne lisibilité. A chaque fermeture d'une instance de la classe "DialogueNouvelleTransaction" ou de la classe "DialogueModificationDeTransaction", ou de la fenêtre de confirmation de suppression d'une transaction, le tableau se réinitialise, ce qui permet de le garder tout le temps à jour.

- **Trois indicateurs numériques**

Chacun fonctionne comme le tableau, seul les requêtes diffèrent : Pour l'indicateur de la somme des IRM et celui de la somme des autres examens, c'est via une requête SQL qu'ils obtiennent chacun un résultat. Pour l'indicateur du total, c'est directement la somme des montants affichés dans les deux autres indicateurs qui est calculée pour avoir le résultat afin de ne pas alourdir encore plus le SGBD.

3. DialogueNouvelleTransaction

Elle hérite de la classe prédéfinie "QDialog". Elle dessine une boîte de dialogue contenant :

- **Deux onglets**

Le premier onglet se nomme "Informations sur le patient". Il contient 8 champs d'édition, 3 combobox et 3 champs d'édition de date :

- **Nom** : c'est un champ d'édition. Il représente le nom du patient. Il bénéficie d'une méthode qui permet de mettre toutes les lettres entrées en majuscule.
- **Prénom** : c'est un champ d'édition. Il représente le(s) prénom(s) du patient. Il bénéficie d'une méthode qui permet de mettre la première lettre entrée en majuscule et le reste en minuscule. Cette méthode met aussi chaque lettre après un espace ou un trait d'union en majuscule pour les patients ayant plusieurs prénoms (exemple : Mohammed Amine) ou un prénom composé (Nasr-Eddine).
- **Nom du mari** : c'est un champ d'édition. Il représente le nom du mari du patient si c'est une femme mariée. Il bénéficie aussi d'une méthode qui permet de mettre toutes les lettres entrées en majuscule.

- **Date de passage d'examen** : c'est un champ d'édition de date. Il permet de préciser la date et l'heure de passage d'examen en faisant apparaître un calendrier miniature pour choisir la date ou simplement en entrant la date manuellement au clavier. Il est initialisé à la date et l'heure du moment où la boîte de dialogue apparaît en récupérant cette donnée à partir du serveur sur lequel se trouve le SGBD. Si c'est un complément de versement, ce champ est initialisé à la même date et heure que celles du premier versement.
- **Date de naissance** : c'est un champ d'édition de date. Il permet de préciser la date de naissance du patient en faisant apparaître un calendrier miniature pour choisir la date ou simplement en entrant la date manuellement au clavier. Il est initialisé à la date au "13/09/1948". La date ne dépasse pas la date du moment où la boîte de dialogue apparaît en récupérant cette donnée à partir du serveur sur lequel se trouve le SGBD.
- **Age** : c'est un champ d'édition en lecture seule. Il permet d'afficher l'âge du patient en le calculant à partir de la date de passage d'examen et de la date de naissance. La méthode qui calcule l'âge dont il bénéficie permet un affichage particulier pour les âges inférieurs à 5 ans en ajoutant l'information sur le nombre de mois. Si l'âge est même inférieur à 1 mois, la méthode permet d'affiner l'âge en affichant le nombre de jours.
- **Groupe d'examen** : c'est une combobox. Elle est remplie par les noms des groupes d'examen se trouvant sur la base de données. Elle permet de remplir la combobox suivante qui en dépend. L'index courant est initialisé à -1 ce qui permet de laisser l'élément affiché dans cette combobox vide. Elle est éditable, ce qui permet de facilement trouver le groupe voulu en tapant le début de son nom dans ce champ.
- **Type d'examen** : c'est une combobox. Elle est remplie par les noms des types d'examen se trouvant sur la base de données qui sont dans le groupe d'examen sélectionné dans la combobox précédente. Elle se vide à chaque changement de l'élément sélectionné dans la combobox précédente puis se remplit avec les types d'examens du groupe sélectionné. L'élément initial affiché est vide, et la combobox est remplie initialement avec tous les types d'examens, tout groupes d'examen confondus, mais à chaque changement du groupe d'examen, l'élément affiché est le nom du premier type d'examen du

groupe sélectionné. Elle permet d'afficher le montant du type d'examen sélectionné selon le type de convention sélectionné. Elle est éditable, ce qui permet de facilement trouver le type d'examen en tapant le début de son nom dans ce champ.

- **Nom du médecin**: c'est un champ d'édition. Il représente le nom médecin. Il bénéficie aussi d'une méthode qui permet de mettre toutes les lettres entrées en majuscule.
- **Montant** : c'est un champ d'édition. Il représente le montant compté pour l'examen pratiqué sur le patient. Il est initialisé selon le prix enregistré sur la base de données pour le type d'examen sélectionné et la convention choisie sauf si c'est un complément de versement auquel cas ce champ est initialisé à la valeur du montant compté dans le premier versement. Ce champ reste éditable pour que le caissier puisse modifier la valeur du montant si le centre d'imagerie fait une ristourne pour le patient.
- **Conventionné** : c'est une combobox. Elle permet de préciser si le patient est conventionné ou non, et avec quelle type de convention. A l'ouverture de la boîte de dialogue, cette combobox se remplit avec les types de conventions se trouvant dans la table « CONVENTION » avec comme élément affiché la première convention qui est "non" signifiant que le patient n'est pas conventionné. Cette combobox, permet de mettre à jour le montant du type d'examen sélectionné.
- **Montant versé** : c'est un champ d'édition. Il représente le montant versé dans cette transaction. Si c'est le premier versement pour l'examen, le montant versé est initialisé à la même valeur que celle du montant. Sinon, le montant versé est initialisé à la différence entre le montant et la somme des anciens montants versés. Ce champ reste éditable, donc le caissier peut très bien modifier la valeur si le patient n'a pas la somme complète à verser.
- **Montant restant ou ajouté** : c'est champ d'édition en lecture seule. Trois cas se présentent pour ce champ :
 1. La valeur inscrite est égale à zéro et la couleur est verte.
 2. La valeur inscrite est négative et la couleur est rouge.
 3. La valeur inscrite est positive et la couleur est bleue.

- **Date de paiement** : c'est un champ d'édition de date. Il permet de préciser la date et l'heure de paiement en faisant apparaître un calendrier miniature pour choisir la date ou simplement en entrant la date manuellement au clavier. Il est initialisé à la date et l'heure du moment où la boîte de dialogue apparaît en récupérant cette donnée à partir du serveur sur lequel se trouve le SGBD.

Le deuxième onglet se nomme "Commentaires sur la transaction". Cet onglet contient un unique widget où on peut éditer du texte afin d'y mettre les commentaires.

- Un bouton "OK"

Il est désactivé par défaut.

Il ne s'active que quand les champs du nom, du prénom, du groupe d'examen, du type d'examen, du montant, du montant versé, du montant restant ou ajouté ne sont pas vides.

Une fois cliqué, il récupère le contenu de chaque champ d'édition dans sa variable propre, puis lance trois opérations :

1. En faisant la correspondance du nom, prénom, nom du mari, date de naissance du patient avec ce qu'il y a dans la table « PATIENT », si celui-ci existe déjà, il n'y a pas d'insertion dans cette table ; Sinon, une requête permettra d'insérer le nouveau patient avec comme "ID_PATIENT" l'identifiant maximal de la table incrémenté de 1 ;
2. Même opération en ce qui concerne le nom du médecin dans la table « MEDECIN » ;
3. Une requête d'insertion de la nouvelle transaction dans la table « TRANSACTION » avec comme "ID_TRANSACTION" l'identifiant maximal de la table incrémenté de 1.

- Un bouton "Annuler"

Il est toujours activé.

Il ferme la fenêtre et permet de revenir à la classe parente qui est MainWindow.

4. DialogueModificationDeTransaction

Elle hérite de la classe prédéfinie "QDialog". Elle dessine une boîte de dialogue semblable à celle de la classe précédente. Seule l'initialisation des champs diffère. En effet, le constructeur de cette classe, en plus d'avoir la classe parente comme paramètre, il a aussi un entier qui est l'"ID_TRANSACTION" de la transaction sélectionnée dans le tableau de la classe parente MainWindow. Cet identifiant permet de lancer la requête de sélection des informations contenue dans la transaction sélectionnée afin d'initialiser chaque champ. Les traitements sont ensuite les mêmes que ceux de la classe précédente, la seule différence vient du fait qu'il y ait un 4^{ème} cas pour le champ du montant restant ou ajouté qui est que la valeur soit négative mais avec une couleur bleue dans le cas où cette transaction aurait été complétée plus tard.

5. DialogueRechercher

Elle hérite de la classe prédéfinie "QDialog". Elle dessine une boîte de dialogue contenant :

- **Nom** : c'est un champ d'édition qui permet d'inscrire le nom du patient à rechercher.
- **Prénom** : c'est un champ d'édition qui permet d'inscrire le prénom du patient à rechercher.
- **Date de paiement** : c'est une combobox contenant 3 éléments :
 1. "Inconnue" : si la date recherchée est quelconque. Les éléments "date minimale", "&" et "date maximale" sont désactivés.
 2. "Le" : si la date recherchée est spécifiée. L'élément "date minimale" devient activé, et les éléments "&" et "date maximale" sont désactivés.
 3. "Entre" : si la recherche est voulue dans un intervalle de deux dates. Les éléments "date minimale", "&" et "date maximale" sont activés.
- **Date minimale** : c'est un champ d'édition de date. . Il permet de préciser :
 1. Si l'élément "Le" est sélectionné dans la combobox précédente, la date de paiement des transactions à rechercher ;
 2. Si l'élément "Entre" est sélectionné dans la combobox précédente, la date de minimale des dates de paiement des transactions à rechercher

en faisant apparaître un calendrier miniature pour choisir la date ou simplement en entrant la date manuellement au clavier. Il est initialisé à la date du moment où la boîte de dialogue apparaît en récupérant cette donnée à partir du serveur sur lequel se trouve le SGBD.

- **&** : c'est un champ d'édition en lecture seule. Il permet simplement de montrer, quand l'élément "Entre" de la combobox "Date de paiement" est sélectionné, que c'est dans l'intervalle des dates contenues dans les dates minimale et maximale que sera effectuée la recherche.
- **Date maximale** : c'est un champ d'édition de date. Il fonctionne de la même façon que "Date minimale" à l'exception qu'il ne s'active que quand l'élément "Entre" de la combobox "Date de paiement" est sélectionné afin de représenter la date maximale de l'intervalle des dates de paiement des transactions à rechercher.
- **Un bouton "OK"** : ce bouton est désactivé par défaut. Il s'active au moment où la combobox de la date de paiement change. Il permet de fermer cette boîte de dialogue et d'instancier la boîte de dialogue "dialogueDesResultatsDeLaRecherche" de la classe DialogueDesResultatsDeLaRecherche avec dans le constructeur, en plus du parent, le nom, le prénom, la date de paiement, la date minimale, la date maximale, le groupe d'examen et le type d'examen.
- **Un bouton "Annuler"** : ce bouton est toujours activé. Il permet de fermer la boîte de dialogue est de revenir à la classe parente.

6. DialogueResultatsDeLaRecherche

Elle hérite de la classe prédéfinie "QDialog". Elle dessine une boîte de dialogue contenant un tableau donnant les transactions ayant comme informations les données demandé dans son constructeur.

7. DialogueGestionDesUtilisateurs

Elle hérite de la classe prédéfinie "QDialog". Elle dessine une boîte de dialogue contenant un tableau donnant l'ID de chaque utilisateur, son nom et son groupe à partir des tables « UTILISATEURS » et « GROUPE_D_UTILISATEURS ».

8. DialogueGestionDesExamensEtDeLeursPrix

Elle hérite de la classe prédéfinie "QDialog". Elle dessine une boîte de dialogue contenant :

- Un combobox : pour générer le tableau se trouvant en dessous selon le groupe d'examens sélectionné.
- Un bouton "Ajouter"
- Un bouton "Supprimer"
- Un tableau

9. DialogueAPropos

Elle hérite de la classe prédéfinie "QDialog". Elle dessine une boîte de dialogue contenant un champ d'édition de texte en lecture seule pour l'affichage du nom d'auteur et des éventuelles.

Conclusion et perspectives

Conclusion et perspectives

Dans ce travail, nous avons étudié le cas de gestion dans un centre d'imagerie médicale dans un premier temps et nous avons proposé une amélioration afin d'accélérer et de faciliter le travail des employés. Cette amélioration consistait en l'élaboration d'un logiciel de gestion des patients. La question du choix du langage de programmation et du système de gestion de bases de données se posait alors. Nous avons exposé quelques-uns des SGBD les plus connus et avons choisi MariaDB pour sa gratuité, la liberté de sa licence, sa disponibilité sur les principaux systèmes d'exploitation et son équivalence avec la solution désormais bien établie MySQL. Pour ce qui est du langage, voulant créer une application portable pour ne pas obliger les utilisateurs à choisir une plateforme unique, et une application qui ait aussi les avantages des applications natives, c'est-à-dire sans ralentissement ni lourdeur, nous avons opté pour utiliser le framework Qt, orienté objet et écrit en langage C++, de par aussi sa documentation assez fournie que ce soit via la documentation officielle ou celle trouvée sur le net.

Nous sommes arrivés à une réalisation partielle du but préalablement visé. Nous avons bien réussi à créer une application fonctionnelle de gestion mais le seul type d'utilisateur concerné est le caissier.

Il serait judicieux donc de continuer à développer l'application et à l'étendre aux autres types d'utilisateur avec les droits d'utilisations appropriés à chacun et d'ajouter des fonctionnalités facilitant son utilisation comme :

- L'ajout de l'auto-complétion pour les champs des noms et prénoms ;
- Le remplacement des combobox par des champs d'édition avec auto-complétion ;
- La possibilité pour l'utilisateur d'ajouter :
 - o De nouveaux examens avec leurs prix ;
 - o De nouveaux types de convention ;
 - o De nouveaux utilisateurs, de nouveaux groupes d'utilisateurs et de modification des droits d'utilisation ;
- Une meilleure gestion de la mémoire en réduisant le nombre de variables et le nombre de requête SQL ;

- L'utilisation des noms d'utilisateurs présents dans la table « UTILISATEUR » et le chiffrement des mots de passes sur la base de données ;
- La possibilité de changer l'adresse IP du serveur sur lequel se trouve le SGBD et le port avant la connexion.

Références bibliographiques

Références Bibliographiques

- [1] DMSOft Technologies, «Popular Database Management Systems Overview», 2014. [En ligne]. Disponible: <http://dbconvert.com/overview.php>. [Accès en Juin 2014].
- [2] Oracle Corporation et/ou ses filiales, «MySQL :: Pourquoi MySQL?», 2014. [En ligne]. Disponible: <https://www.mysql.fr/why-mysql/>. [Accès en Juin 2014].
- [3] Oracle Corporation et/ou ses filiales, «MySQL :: MySQL Customers», 2014. [En ligne]. Disponible: <http://www.mysql.com/customers/>. [Accès en Juin 2014].
- [4] Oracle Corporation et/ou ses filiales, «MySQL :: Sun acquiert MySQL», 2008. [En ligne]. Disponible: <http://www.mysql.fr/news-and-events/sun-to-acquire-mysql.html>. [Accès en Juin 2014].
- [5] Nil Sanyas, Next INpact, «Le rachat de Sun Microsystems par Oracle enfin effectif (MàJ)», 22 Mars 2010. [En ligne]. Disponible: <http://www.nextinpact.com/news/53958-europe-acquisition-sun-microsystems-oracle.htm>. [Accès en Juin 2014].
- [6] Oracle Corporation et/ou ses filiales, «MySQL :: MySQL Downloads», 2014. [En ligne]. Disponible: <http://www.mysql.fr/downloads/>. [Accès en Juin 2014].
- [7] Vincent Hermann, Next INpact, «Entretien avec Michael Widenius, fondateur de MySQL et MariaDB», 27 Mars 2013. [En ligne]. Disponible: <http://www.nextinpact.com/news/78598-entretien-avec-michael-widenius-fondateur-mysql-et-mariadb.htm>. [Accès en Juin 2014].
- [8] MariaDB Knowledge Base, «MariaDB versus MySQL - Compatibility - MariaDB Knowledge Base», 21 Mai 2013. [En ligne]. Disponible: <https://mariadb.com/kb/en/mariadb-vs-mysql-compatibility/>. [Accès en Juin 2014].
- [9] MariaDB Knowledge Base, «About MariaDB - MariaDB Knowledge Base», 29 Septembre 2013. [En ligne]. Disponible: <https://mariadb.com/kb/en/about-mariadb/>. [Accès en Juin 2014].
- [10] Feldman Asher, Wikimedia blog, «Wikipedia Adopts MariaDB - Wikimedia blog», 22 Avril 2013. [En ligne]. Disponible: <http://blog.wikimedia.org/2013/04/22/wikipedia-adopts-mariadb/>. [Accès en Juin 2014].
- [11] Jack Clark, The Register, «Google swaps out MySQL, moves to MariaDB - The Register», 12 Septembre 2013. [En ligne]. Disponible: http://www.theregister.co.uk/2013/09/12/google_mariadb_mysql_migration/. [Accès en Juin 2014].

- [12] Sheeri, Mozilla IT, «MySQL 5.1 vs. MySQL 5.5: Floats, Doubles, and Scientific Notation | Mozilla IT», 17 Janvier 2013. [En ligne]. Disponible: <http://blog.mozilla.org/it/2013/01/17/mysql-5-1-vs-mysql-5-5-floats-doubles-and-scientific-notation/>. [Accès en Juin 2014].
- [13] Sam Varghese, iTWire, «iTWire - Red Hat ditches MySQL, switches to MariaDB», 14 Juin 2013. [En ligne]. Disponible: <http://www.itwire.com/business-it-news/open-source/60292-red-hat-ditches-mysql-switches-to-mariadb>. [Accès en Juin 2014].
- [14] Steven J. Vaughan-Nichols, ZDNet - Linux and Open Source, «Oracle who? Fedora & openSUSE will replace MySQL with MariaDB | ZDNet», 31 Janvier 2013. [En ligne]. Disponible: <http://www.zdnet.com/oracle-who-fedora-and-opensuse-will-replace-mysql-with-mariadb-7000010640/>. [Accès en Juin 2014].
- [15] MariaDB Foundation, «MariaDB 5.5.38 Stable - MariaDB», 9 Juin 2014. [En ligne]. Disponible: <https://downloads.mariadb.org/mariadb/5.5.38/>. [Accès en Juin 2014].
- [16] MariaDB Foudation, «MariaDB 10.0.12 Stable - MariaDB», 16 Juin 2014. [En ligne]. Disponible: <https://downloads.mariadb.org/mariadb/10.0.12/>. [Accès en Juin 2014].
- [17] The Oracle FAQ, «Oracle 8 - Oracle FAQ», 23 Novembre 2008. [En ligne]. Disponible: http://www.orafaq.com/wiki/Oracle_8. [Accès en Juin 2014].
- [18] DMSoft Technologies, «Oracle migration and synchronization», 2014. [En ligne]. Disponible: <http://dbconvert.com/oracle.php>. [Accès en Juin 2014].
- [19] Oracle Corporation, «Oracle Database Software Downloads | Oracle Technology Network | Oracle», 2014. [En ligne]. Disponible: <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>. [Accès en Juin 2014].
- [20] DMSoft Technologies, «MS SQL migration and sync tools», 2014. [En ligne]. Disponible: <http://dbconvert.com/mssql.php>. [Accès en Juin 2014].
- [21] Microsoft, «SQL Azure : Développer et déployer des applications sur le Cloud | Microsoft Cloud et Serveurs d'Entreprises», 2014. [En ligne]. Disponible: <http://www.microsoft.com/france/serveur-cloud/sql/2012/sql-azure.aspx>. [Accès en Juin 2014].
- [22] Microsoft, «SQL Server 2012-2014-Buy | Server Cloud», 2014. [En ligne]. Disponible: <http://www.microsoft.com/en-us/server-cloud/products/sql-server/Buy.aspx#fbid=LYI64c5q7VA>. [Accès en Juin 2014].
- [23] PostgreSQL, «PostgreSQL: About», 2014. [En ligne]. Disponible: <http://www.postgresql.org/about/>. [Accès en Juin 2014].

- [24] PostgreSQL, «PostgreSQL: The world's most advanced open source database», 2014. [En ligne]. Disponible: <http://www.postgresql.org/>. [Accès en Juin 2014].
- [25] PostgreSQLwiki, «FAQ - PostgreSQL wiki - What is the license of PostgreSQL?», 22 Aout 2013. [En ligne]. Disponible: http://wiki.postgresql.org/wiki/FAQ#What_is_the_license_of_PostgreSQL.3F. [Accès en Juin 2014].
- [26] Eric Lai, Computerworld, «Size matters: Yahoo claims 2-petabyte database is world's biggest, busiest - Computerworld», 22 Mai 2008. [En ligne]. Disponible: http://www.computerworld.com/s/article/9087918/Size_matters_Yahoo_claims_2_peta_byte_database_is_world_s_biggest_busiest?taxonomyId=18&intsrc=hm_topic. [Accès en Juin 2014].
- [27] Eric Lai, Computerworld, «Sony Online opts for open-source database over Oracle - Computerworld», 20 Mars 2006. [En ligne]. Disponible: http://www.computerworld.com/s/article/109722/Sony_Online_opts_for_open_source_database_over_Oracle?taxonomyId=053. [Accès en Juin 2014].
- [28] Instagram, Inc., «Keeping Instagram up with over a million new users in twelve hours - Instagram Engineering», 2012. [En ligne]. Disponible: <http://instagram-engineering.tumblr.com/post/20541814340/keeping-instagram-up-with-over-a-million-new-users-in#replicationread-slaves>. [Accès en Juin 2014].
- [29] SQLite, «SQLite Home Page», 4 Juin 2014. [En ligne]. Disponible: <https://www.sqlite.org/>. [Accès en Juin 2014].
- [30] SQLite, «About SQLite», 4 Juin 2014. [En ligne]. Disponible: <http://www.sqlite.org/about.html>. [Accès en Juin 2014].
- [31] DMSoft Technologies, «SQLite migration and Synchronization», 2014. [En ligne]. Disponible: <http://dbconvert.com/sqlite.php>. [Accès en Juin 2014].
- [32] SQLite, «SQLite Copyright», 4 Juin 2014. [En ligne]. Disponible: <http://www.sqlite.org/copyright.html>. [Accès en Juin 2014].
- [33] SQLite, «Well-Known Users Of SQLite», 4 Juin 2014. [En ligne]. Disponible: <http://www.sqlite.org/famous.html>. [Accès en Juin 2014].
- [34] MariaDB Foundation, «MariaDB 5.5.38 Stable - MariaDB», 9 Juin 2014. [En ligne]. Disponible: https://downloads.mariadb.org/mariadb/5.5.38/#os_group=windows. [Accès en Juin 2014].

- [35] MariaDB Foundation, «MariaDB - Setting up MariaDB Repositories - MariaDB», 2014. [En ligne]. Disponible: https://downloads.mariadb.org/mariadb/repositories/#mirror=zol&distro=Debian&distro_release=wheezy&version=5.5. [Accès en Juin 2014].
- [36] Oracle Corporation et/ou ses filiales, «MySQL :: MySQL Workbench», 2014. [En ligne]. Disponible: <https://www.mysql.fr/products/workbench/>. [Accès en Juin 2014].
- [37] Oracle Corporation et/ou ses filiales, «MySQL :: New! MySQL Workbench: Database Migration», 2014. [En ligne]. Disponible: <https://www.mysql.fr/products/workbench/migrate/>. [Accès en Juin 2014].
- [38] Oracle Corporation et/ou ses filiales, «MySQL :: Download MySQL Workbench», 2014. [En ligne]. Disponible: <https://dev.mysql.com/downloads/workbench/>. [Accès en Juin 2014].
- [39] Oracle Corporation et/ou ses filiales, «MySQL :: MySQL Workbench :: 1.1 MySQL Workbench Editions», 2014. [En ligne]. Disponible: <https://dev.mysql.com/doc/workbench/en/wb-editions.html>. [Accès en Juin 2014].
- [40] Digia, «QtWhitepaper | Qt Wiki | Qt Project», 15 Juin 2013. [En ligne]. Disponible: <http://qt-project.org/wiki/QtWhitepaper>. [Accès en Juin 2014].
- [41] Mathieu Nebra et Matthieu Schaller, OpenClassrooms, «Introduction à Qt», 2014. [En ligne]. Disponible: <http://fr.openclassrooms.com/informatique/cours/programmez-avec-le-langage-c/introduction-a-qt>. [Accès en Juin 2014].
- [42] Digia, «Qt - Qt Framework», 2014. [En ligne]. Disponible: <http://qt.digia.com/Product/Qt-Framework/>. [Accès en Juin 2014].
- [43] Digia, «Supported Platforms | QtDoc 5.3 | Documentation | Qt Project», 2013. [En ligne]. Disponible: <http://qt-project.org/doc/qt-5/supported-platforms.html>. [Accès en Juin 2014].
- [44] KDE e.V., «KDE API Reference - API Reference Index», 2014. [En ligne]. Disponible: <http://api.kde.org/>. [Accès en Juin 2014].
- [45] Digia, «Qt Project», 2014. [En ligne]. Disponible: <http://qt-project.org/>. [Accès en Juin 2014].
- [46] Digia, «Qt 5 | Documentation | Qt Project», 2014. [En ligne]. Disponible: <http://qt-project.org/doc/qt-5/index.html>. [Accès en Juin 2014].
- [47] Digia, «Qt Creator : IDE Overview | Documentation | Qt Project», 2013. [En ligne]. Disponible: <http://qt-project.org/doc/qtcreator-2.5/creator-overview.html>. [Accès en Juin 2014].

- [48] Digia, «Qt Creator : Adding Tool Chains | Documentation | Qt Project», 2013. [En ligne]. Disponible: <http://qt-project.org/doc/qtcreator-2.5/creator-tool-chains.html>. [Accès en Juin 2014].
- [49] Digia, «Qt Creator : Debugging and Analyzing | Documentation | Qt Project», 2013. [En ligne]. Disponible: <http://qt-project.org/doc/qtcreator-2.5/creator-testing.html>. [Accès en Juin 2014].
- [50] Digia, «Qt Creator : Supported Platforms | Documentation | Qt Project», 2013. [En ligne]. Disponible: <http://qt-project.org/doc/qtcreator-2.5/creator-os-supported-platforms.html>. [Accès en Juin 2014].
- [51] Digia, «Using a Designer UI File in Your Application | Documentation | Qt Project», 2013. [En ligne]. Disponible: <http://qt-project.org/doc/qt-4.8/designer-using-a-ui-file.html>. [Accès en Juin 2014].
- [52] Digia, «Qt Creator : User Interface | Documentation | Qt Project», 2013. [En ligne]. Disponible: <http://qt-project.org/doc/qtcreator-2.5/creator-quick-tour.html>. [Accès en Juin 2014].
- [53] Mathieu Nebra et Matthieu Schaller, OpenClassrooms, «Compiler votre première fenêtre Qt», 2014. [En ligne]. Disponible: <http://fr.openclassrooms.com/informatique/cours/programmez-avec-le-langage-c/compiler-votre-premiere-fenetre-qt>. [Accès en Juin 2014].

Annexe A : Installation des outils utilisés pour l'implémentation de l'application sous GNU/Linux Debian Wheezy

1. Le framework Qt

L'installation des outils du framework Qt peut se faire en sélectionnant le paquet "qtcreator" dans le gestionnaire graphique de paquets "Synaptic".

2. MariaDB

Ce SGBD n'étant pas disponible dans les dépôts officiels de Debian Wheezy, l'installation de la version 5.5 utilisé dans l'élaboration de notre application se fait comme suit :

1. Fermer "Synaptic" s'il est ouvert et taper dans un émulateur de terminal ce qui suit:

```
sudo apt-get install python-software-properties
sudo apt-key adv --recv-keys --keyserver keyserver.ubuntu.com
0xcbc082a1bb943db
```

2. Ouvrir Synaptic et aller dans "Configuration" puis "Dépôts" ;
3. Dans la fenêtre qui s'ouvre, ajouter un dépôt en cliquant sur "Nouveau" et entrer les données suivantes :

```
Binaire : Deb
URI : http://mirror.zol.co.zw/mariadb/repo/5.5/debian
Distribution : Wheezy
Catégorie(s) : main
```

4. Cliquer sur "Valider" ;
5. Cliquer sur le bouton "Recharger" de la fenêtre principale ;
6. Sélectionner le paquet "mariadb-server".^[35]

3. MySQL Workbench

Etant disponible dans les dépôts officiels de Debian Wheezy, son installation ne demande que la sélection du paquet "mysql-workbench" dans Synaptic.

Annexe B : Installation des outils nécessaires pour le fonctionnement de l'application sous GNU/Linux Debian Wheezy

L'application réalisée dépend des deux paquets "libqtgui4" et "libqt4-sql-mysql" dont l'installation peut se faire via "Synaptic".

Etant donné que MariaDB n'est pas disponible dans les dépôts officiels de Debian Wheezy, il faut ajouter le dépôt officiel de MariaDB comme expliqué dans l'annexe A sans sélectionner de paquet "mariadb-server" pour installation.

Le paquet "libqt4-sql-mysql" dépendra ainsi du paquet "libmariadbclient-18" dont il proposera l'installation.

Un dossier nommé "icons" se trouvant à la racine de l'exécutable de notre application devra être présent. Il contient les trois icônes de couleur pour les trois états des transactions, l'icône de l'application et l'image présentée dans la fenêtre de "A propos".

Annexe C : Installation des outils utilisés pour l'implémentation de l'application sous Windows

1. Le framework Qt

Voulant être en synchronisation durant le développement entre Debian et Windows, il nous a fallu avoir la même version de Qt. Et la version présente dans les dépôts officiels de Debian Wheezy est la 4.8, alors que la version la plus récente sur le site officiel de Qt au moment d'écrire ces lignes est la version 5.3. Il a fallu donc installer la même version sur Windows. Pour ce faire, il faut :

1. Installer "qt-opensource-windows-x86-mingw482-4.8.6-1" disponible ici -> <http://download.qt-project.org/archive/qt/4.8/4.8.6/>
2. Installer "mingw32" disponible ici -> http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/4.8.2/threads-posix/dwarf/i686-4.8.2-release-posix-dwarf-rt_v3-rev3.7z/download?use_mirror=garr&r=&use_mirror=garr
3. Installer "qt-creator-win-opensource-2.5.2" disponible ici -> <http://download.qt-project.org/archive/qtcreator/2.5/>

2. MariaDB et MySQL Workbench

MariaDB 5.5 et MySQL Workbench étant disponibles en téléchargement pour Windows à partir de leurs sites officiels respectifs.

Annexe D : Les DLL nécessaires pour fonctionnement de l'application sous Windows

L'application réalisée sous forme d'exécutable aura besoin des DLL suivantes dans la racine du fichier :

- libgcc_s_dw2-1.dll ;
- libmysql.dll ;
- libstdc++-6.dll ;
- libwinpthread-1.dll ;
- QtCore4.dll ;
- QtGui4.dll ;
- QtSql4.dll.

Toutes repérables dans le dossier se trouvant dans "[Chemin d'installation de la version de Qt]\bin". Si "libmysql.dll" est introuvable dans ce dossier, l'explication de sa création est donnée dans l'annexe E.

Il faut aussi créer un dossier à la racine de l'exécutable nommé "sqldrivers" pour contenir une dernière DLL appelée "qsqlmysql4.dll" dont la génération est aussi expliquée dans l'annexe E.

Il ne restera plus que le dossier "icons" à la racine de l'exécutable aussi à ajouter pour contenir les trois icônes de couleur pour les trois états des transactions, l'icône de l'application et l'image présentée dans la fenêtre de "A propos".

Annexe E: Génération des DLL de MySQL

A cause d'un problème de licence, Qt ne permet pas de se connecter à une base de données de type MySQL sans les étapes suivantes :

1. Installer MariaDB
2. Dans Qt Creator, ouvrir le projet "mysql.pro" se trouvant dans "[Chemin d'installation de la version de Qt]\src\plugins\sqldrivers\mysql"
3. Modifier le fichier "mysql.pro" en insérant entre les deux "include" les lignes suivantes :

INCLUDEPATH	+= "[Chemin du dossier d'installation de MariaDB] /include/mysql"
QMAKE_LIBDIR	+= "[Chemin du dossier d'installation de MariaDB]/lib"
LIBS	+= libmysql

4. Sélectionner le mode de compilation en "Release"
5. Sur la barre d'outils, aller dans "Build" > "Rebuild Project mysql"
6. Après cette compilation, dans le dossier "[Chemin d'installation de la version de Qt]\src\plugins\sqldrivers\mysql\release" se trouveront, entre autres fichiers, "qsqlmysql4.dll" et "qsqlmysql4.a". Ces deux fichiers devront être copiés dans le dossier "[Chemin d'installation de la version de Qt]\plugins\sqldrivers"
7. Faire de même en utilisant le mode de compilation "Debug" pour générer "qsqlmysqld4.dll" et "qsqlmysqld4.a" dans le dossier "[Chemin d'installation de la version de Qt]\src\plugins\sqldrivers\mysql\debug"
8. Copier ensuite le fichier "libmysql.dll", se trouvant dans "[Chemin du dossier d'installation de MariaDB]\lib", dans le dossier "[Chemin d'installation de la version de Qt]\bin"

Glossaire

Mots-clés	Définition
API	Application Programming Interface. C'est une interface de programmation servant de pont entre l'application et une autre application ou le système d'exploitation.
Binding	Etant donné que la plupart des bibliothèques sont écrites en langage C ou C++ pour des besoins de performance. Le binding d'une bibliothèque pour un autre langage est la création d'une API qui permet de faire le pont entre cette bibliothèque et cet autre langage qui est souvent de plus haut niveau.
CLI	Commande Line Interface, ou interface en mode ligne de commande. C'est une interface homme-machine dans laquelle la communication entre l'utilisateur et l'ordinateur s'effectue en mode texte
Cloud Computing	Informatique dans les nuages. Désigne l'ensemble des opérations à grande échelle effectuées sur le Web. Principalement lié aux Big Data, les bases de données géantes.
DLL	Dynamic Link Library. C'est le format des fichiers des bibliothèques logicielles de Windows. Ils utilisent la technique des liens dynamiques pour permettre de les associer avec les programmes qui s'en servent.
Fork	C'est un projet concurrent à un autre projet basé sur une version préexistante du code source de ce dernier.
Framework	Ensemble conséquent de bibliothèques.
GCC	GNU Compiler Collection. C'est un ensemble de compilateurs créés par le projet GNU. GCC est un logiciel libre capable de compiler divers langages de programmation dont le C, C++, Objective-C, Java, Ada et Fortran.
GNU	GNU's Not UNIX. Projet de libération des logiciels. GNU/Linux est le nom parfois donné à un système d'exploitation associant des éléments essentiels du projet GNU et un noyau Linux.

GUI	Graphical User Interface, ou interface graphique. C'est une interface homme-machine dans laquelle la communication entre l'utilisateur et l'ordinateur s'effectue via des fenêtres.
IDE	Integrated Development Environment ou Environnement de Développement Intégré. C'est un ensemble d'outils pour augmenter la productivité des programmeurs qui développent des logiciels. Il comporte un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de construction.
IHM	Interface Homme Machine. Moyen mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une machine
Javadoc	Standard industriel pour la documentation des classes Java. C'est un outil permettant de créer une documentation d'API au format HTML depuis les commentaires présents dans un code source en Java.
Licence BSD	Licence Berkeley Software Distribution. C'est licence libre. Elle permet de réutiliser tout ou une partie du logiciel sans restriction, qu'il soit intégré dans un logiciel libre ou propriétaire.
Licence GNU GPL	GNU General Public Licence ou licence publique générale GNU. C'est une licence libre du projet GNU. L'objectif de la licence GNU GPL, selon ses créateurs est de garantir à l'utilisateur les droits suivants (appelés libertés) sur un programme informatique : <ol style="list-style-type: none"> 1. La liberté d'exécuter le logiciel, pour n'importe quel usage ; 2. La liberté d'étudier le fonctionnement d'un programme et de l'adapter à ses besoins, ce qui passe par l'accès aux codes sources ; 3. La liberté de redistribuer des copies ; 4. L'obligation de faire bénéficier à la communauté des versions modifiées.

Licence MIT	Licence Massachusetts Institute of Technology, ou licence X11. C'est une licence libre. Elle donne à toute personne recevant le logiciel le droit illimité de l'utiliser, le copier, le modifier, le fusionner, le publier, le distribuer, le vendre et de changer sa licence. La seule obligation est de mettre le nom des auteurs avec la notice de copyright.
Makefile	make est un logiciel qui construit automatiquement des fichiers, souvent exécutables, ou des bibliothèques à partir d'éléments de base tels que du code source. Il utilise des fichiers appelés makefile qui spécifient comment construire les fichiers cibles.
MinGW	Minimalist GNU for Windows. C'est une adaptation des logiciels de développement et de compilation du GNU (GCC), à la plateforme Win32.
MSVC	Microsoft Visual C++. C'est le compilateur C++ de Microsoft adapté à la plateforme Windows.
SGBD	Système de gestion de bases de données, c'est un logiciel permettant de gérer et de stocker des collections de données de façon bien organisée et optimisée tout en cachant la complexité des fichiers aux utilisateurs.
SGBDR	Système de gestion de bases de données relationnelles, c'est un logiciel permettant de gérer les bases de données organisées en formes de tables, chacune ayant une relation avec l'autre.
SGBDRO	Système de gestion de bases de données relationnelles-objet, c'est un logiciel permettant de gérer les bases de données mélangeant l'organisation en forme de tables avec le paradigme objet de la Programmation orientée objet, c'est-à-dire les attributs, les classes, l'héritage, etc.
SQL	Structured Query Language, le langage de requête de bases de données le plus utilisé.
Wheezy	Wheezy est le nom de l'actuelle version stable de la distribution GNU/Linux Debian.

Résumé

Dans ce mémoire, nous réalisons une application portable de gestion de caisse pour un centre d'imagerie médicale qui fonctionne aussi bien sous Windows que sous GNU/Linux. Ceci est dû grâce aux outils de développement utilisés : le framework Qt concernant le SDK et le SGBD MariaDB pour l'administration de la base de données.

Dans le présent document, nous commençons par exposer la gestion actuelle des patients dans le centre d'imagerie et donnons une proposition d'amélioration en utilisant les bases de données. Nous expliquons, ensuite, les choix qui nous ont motivés à prendre les deux outils de développement précédemment cités pour l'implémentation du logiciel. Nous terminons par la présentation de l'application réalisée avec les avantages apportés et les perspectives attendues.

MOTS-CLES: Logiciel de gestion, bases de données, Qt, MariaDB.

ملخص

في هذا البحث، نقوم بإنجاز برنامج قابل للحمل لإدارة النّقد لمركز طبيّ للتصوير و الذي يشغل على كلا من ويندوز وجنو/لينكس. هذا يرجع إلى أدوات التّطوير المستخدمة : إطار العمل كيو ت كحزمة أدوات تطوير البرمجيات و ماريا دي بي لإدارة قواعد البيانات.

في هذه الوثيقة، نبدأ بعرض الإدارة الحالية للمرضى في مركز التصوير و نعطي اقتراح تحسين باستخدام قواعد البيانات، نبيّن فيما بعد الخيارات التي دفعتنا لاتخاذ كلا من أدوات التّطوير المذكورة أعلاه لتنفيذ البرنامج. ننتهي بعرض البرنامج المنجز مع الفوائد المترتبة و جهات النّظر المنتظر.

الكلمات المفتاحية : برنامج الإدارة ، قواعد البيانات، كيو ت، ماريا دي بي.

Abstract

In this work, we realize a portable application cash management for a medical imaging center that works on both Windows and GNU/Linux. This is due to the development tools used: Qt framework for the SDK and the DBMS MariaDB for the administration of the database.

In this document, we first expose the current management of the patients in the imaging center and give a suggestion of improvement using the databases. We explain, then, the choices that led us to take both development tools mentioned above for the implementation of the software. We conclude with the presentation of the achieved application with the brought benefits and the awaited perspectives.

KEYWORDS: Management software, databases, Qt, MariaDB.