

**Université Abou Bakr Belkaïd de Tlemcen**  
**Faculté de Technologie**  
**Département de Génie Biomédical**

## **MEMOIRE DE PROJET DE FIN D'ETUDES**

**Pour l'obtention du Diplôme de**

## **MASTER en GENIE BIOMEDICAL**

**Spécialité : Télémedecine**

**Présenté par : MESSAOUR Hanene et BELGUEBLI Radhia**

---

## **Conception d'une plateforme de télédiabète**

---

**Soutenu le 28 Juin 2018 devant le Jury**

Mr	GAOUAR ADIL	<i>MAA</i>	Université de Tlemcen	Président
Mr	DAHO ELHABIB MOSTAFA	<i>MCB</i>	Université de Tlemcen	Examineur
Mme	HAMZACHERIF Souad	<i>MAA</i>	Université de Tlemcen	Encadreur

**Année universitaire 2017-2018**

# **TABLE DE MATIERE**

# INTRODUCTION GENERALE

## Chapitre 1 : CONTEXTE MEDICAL

1. Introduction.....	2
2. Définition de diabète.....	2
2.1.Les types de diabète.....	2
2.2.La surveillance glycémique.....	3
2.3.L'insulinothérapie.....	4
2.4.Les complications.....	7
2.5.L'éducation thérapeutique des patients diabétique .....	7
3. Conclusion.....	9

## Chapitre 2: TRAVAUX CONNEXES

1. Introduction.....	11
2. Problématique.....	11
3. Les travaux réalisées dans le diabète de type 1.....	11
3.1. Consultation téléphonique.....	11
3.2. Télétransmission des données glycémiques.....	12
3.3. Les carnets électroniques actifs sur smartphone.....	12
4. Les expériences réalisées dans le diabète de type 2.....	13
4.1. Consultation téléphonique.....	13
4.2. Systèmes reposant sur un site web interactif.....	13
4.3. Systèmes couplant téléphone cellulaire et serveur à distance.....	13
4.4. Carnets électronique de type smartphone.....	14
5. Synthèse.....	15
6. Conclusion.....	16

## Chapitre 3: CONCEPTION ET IMPLEMENTATION

1. Introduction.....	18
2. Présentation de l'application.....	18
3. Fonctionnalité.....	19
4. Conception.....	19
4.1. Modélisation MERIS.....	19
4.2. Le modèle MVC.....	21
5. Réalisation.....	22
5.1. Environnement de développement.....	22
5.2. Mise en application.....	25

6. Bilan.....	33
6.1. Difficultés rencontrées.....	33
6.2. Perspective.....	33
7. Conclusion.....	34
<b>CONCLUSION GENERALE .....</b>	<b>35</b>
<b>BIBLIOGRAPHIE.....</b>	<b>36</b>
<b>WEBGRAPHIE.....</b>	<b>37</b>

## LISTE DES TABLEAUX

<b>Tableau 1</b> : Les objectifs glycémiques.....	3
<b>Tableau 2</b> : Les familles de l'insuline.....	6
<b>Tableau 3</b> : Les notions nécessaires pour l'équilibre du diabète.....	8

## LISTE DES FIGURES

<b>Figure 1.1</b> : Le pancreas.....	5
<b>Figure 2.1</b> : La transmission par Bluetooth du lecteur de glycémies vers les systèmes WellDoc	
<b>Figure 2.2</b> : Le fonctionnement du système T+.....	15
<b>Figure 2.3</b> : Logo.....	18
<b>Figure 2.4</b> : MCD.....	20
<b>Figure 2.5</b> : Le schéma de la structure générique d'une architecture MVC.....	22
<b>Figure 2.6</b> : Serveur Derby.....	26
<b>Figure 2.7</b> : pilote JDBC Client Derby.....	26
<b>Figure 2.8</b> : Création de la base de données.....	27
<b>Figure 2.9</b> : Création des tables avec SQL Data Definition commands.....	27
<b>Figure 2.10</b> : Exemple du page servlet.....	28
<b>Figure 2.11</b> : Exemple du page JSP.....	28
<b>Figure 2.12</b> : exemple d'un fichier HTML en JavaEE.....	29
<b>Figure 2.13</b> : exemple d'un fichier css en JavaEE.....	29
<b>Figure 2.14</b> : Exemple d'un fichier de configuration.....	30

<b>Figure 2.15</b> : La page d'accueil.....	30
<b>Figure 2.16</b> : Page de connexion.....	31
<b>Figure 2.17</b> : page d'inscription.....	31
<b>Figure 2.18</b> : utilisation de la bibliothèque JSTL.....	31
<b>Figure 2.19</b> : ajout du fichier DWR.jar.....	32

## Liste des abréviations

ADO: Antidiabétique Orale

CFID : Congrès de la Fédération Internationale du Diabète

CSS: Cascading Style Sheets

DT1 : Diabète de type 1

DT2 : Diabète de type 2

FFD : Fédération Françaises des Diabétiques

HDL : High Density Lipoprotein (lipoprotéine de haute densité)

HTTP: Hypertext Transfer Protocol

HTML: Hypertext Markup Language

JDBC : Java Database Connectivity

JSP: Java Server Page

JSTL : JavaServer Pages Standard Tag Library

OMS : Organisation Mondiale de la Santé

PDaphone : Personale Digital Assistant phone

SQL : StructuredQueryLanguage (Langage de requêtes structuré)

SGBD : Système de Gestion de Base de Données

T-IDDM: Telematic Management of Insulin-DependentDiabetesMellitus

XML : eXtensibleMarkupLanguage (langage de balisage extensible)

## **Introduction générale**

La télémédecine est une forme de pratique médicale mettant en rapport des patients avec des médecins ou des professionnels de santé à distance grâce aux technologies de l'information et de la communication qui ont contribué plus que toute autre technologie à une vision du monde "sans frontières", pouvant ainsi faire de la télémédecine une nouvelle « médecine sans frontières », l'un des plus importants domaines d'application de la télémédecine est le suivi des maladies chroniques telles que le diabète. Le diabète touche une tranche très importante de la population mondiale, cette maladie est qualifiée par certains de « le diabète la maladie du XXI siècle » [CFID 2002].

En diabétologie, la télémédecine offre un cadre applicatif pour aider les patients diabétiques à rester en contact permanent avec leurs médecins, elle permet une plus grande réactivité dans la prise en charge de la maladie, dans l'adaptation des doses d'insuline, et donc une plus grande efficacité dans l'obtention d'un bon équilibre glycémique. Les objectifs de la télémédecine sont sans doute différents selon que l'on considère les diabétiques de type 1, ou la télémédecine a été conçue surtout pour les aider à mieux contrôler leurs glycémies, ou les diabétiques de type 2, ou la télémédecine pourrait venir en renforcement d'un système de soin défaillant du fait de la diminution du nombre de médecins face à une augmentation rapide de la prévalence de la maladie. Ainsi de très nombreux travaux ont été réalisés en faveur des diabétiques de type 1 et de type 2.

Dans le cadre de ce mémoire de fin d'étude nous proposons de développer une plateforme Web de télé-diabète qui sert à un outil de télémédecine et cela pour permettre aux personnes diabétiques de communiquer simplement avec leurs soignants qui peuvent de leurs tours suivre en continu et à distance leur données d'auto-surveillance tout en assurant la mobilité, alors qu'une dizaine d'années auparavant, un tel projet aurait nécessité de gros moyens ainsi qu'une grosse infrastructure, aujourd'hui un simple téléphone portable peut contribuer efficacement à la sauvegarde humaine.

Ce mémoire est constitué de chapitres, le premier chapitre présente de manière générale la maladie de diabète ainsi que les différentes méthodes de traitement et de la prise en charge, ces risques et ces complications, et aussi un aperçu sur l'éducation thérapeutique que doit suivre le patient diabétique.

Dans le second chapitre nous présentons les travaux connexes trouvés dans la littérature qui traite de plusieurs façons la problématique qui a rendu la maladie de diabète un axe de recherche très important dans le domaine de l'informatique.



Enfin, dans le dernier chapitre nous présentons notre plateforme web et ses différentes fonctionnalités. Ainsi que les outils utilisés pour son implémentation puisque nous avons opté pour JavaEE qui est une technologie qui utilise un ensemble d'API java et qui permet de développer des applications pourront tourner sous de différents navigateurs, et disposer de la sécurité que procure une application métier java.

**Chapitre 1 : CONTEXTE MEDICAL**

## **1. INTRODUCTION**

Le diabète constitue un problème de santé public majeur, l'organisation mondiale de la santé estime que 422 millions d'adulte dans le monde vivait avec le diabète (Rapport mondial sur le diabète) [OMS 2016]. Cette maladie implique l'adhérence à une conduite thérapeutique par insulinothérapie et une stricte surveillance glycémique afin d'éviter ou de diminuer les risques de complications qui peuvent être provoqué par le diabète tell que la cécité, des accidents vasculaire cérébraux, une insuffisance rénal et même amputation. Nous allons dans ce chapitre définir la maladie du diabète et ses types ainsi que les différentes méthodes de traitement et de la prise en charge, par la suite nous allons citer les risques et les complications. Puis nous allons donner un aperçue sur l'éducation thérapeutique du patient diabétique.

## **2. DEFINITION DU DIABETE**

Le diabète est un trouble métabolique caractérisé par la présence d'une hyperglycémie attribuable à un défaut de la sécrétion d'insuline ou de l'action de l'insuline, ou des deux. L'hyperglycémie chronique liée au diabète est associée à des complications micro vasculaires à long terme assez spécifiques touchant les yeux, les reins et les nerfs, ainsi qu'à un risque accru des maladies cardiovasculaires[Goldenberg et all 2013].

### **2.1 Les types de diabète**

Il existe deux principaux type de diabète, diabète de type1 et diabète de type2, parfois cette maladie peut se développer seulement pendant la période de grossesse, c'est le diabète gestationnel :

#### **2.1.1 Le diabète de type1**

Le diabète de type 1 commence habituellement chez les enfants ou durant l'adolescence, et beaucoup plus rarement chez les adultes. L'organisme des personnes atteintes de diabète type1 ne produit pas assez d'insuline pour contrôler le sucre sanguin, ainsi ils doivent recevoir des injections d'insuline [Parmet et all 2008].

### 2.1.2 Le diabète de type2

Le diabète de type 2, se développe habituellement à l'âge adulte, mais peut également se voir chez les enfants ayant un poids excessif. L'organisme des diabètes de type 2 est résistant aux effets de l'insuline, ce type de diabète est généralement traité par les antidiabétiques oraux ou par l'utilisation de l'insuline injectable qui ne se fait qu'en cas d'échec de ces derniers [Richard M et all 2008].

### 2.1.3 Le diabète gestationnel

Le diabète gestationnel apparaît chez les femmes qui n'ont jamais eu un diabète avant une grossesse, mais dont le glucose dans le sang (**glycémie**) augmente pendant la grossesse. Ce qui exige une évaluation et un traitement durant cette période. Ce type de diabète disparaît habituellement après la grossesse mais peut revenir lors de grossesses ultérieures et peut précéder le développement d'un diabète de type 2 au future[Janet M et all 2008]. Qu'il s'agisse du diabète de type1, diabète de type2 ou diabète gestationnel, une surveillance glycémique est indispensable afin de maintenir l'équilibre de la maladie.

## 2.2 La surveillance glycémique

La surveillance glycémique se fait par la mesure de la glycémie capillaire qui permet de connaître le taux de sucre dans le sang des capillaires, et cela pour adapter les doses d'insuline ou bien détecter les hypoglycémies et les hyperglycémies et parfois pour anticiper sur une activité physique [P. Vague et al 2013]. Dont le but principal est d'atteindre les objectifs glycémiques capillaires recommandés qui sont résumés dans le tableau suivant :

Indications	Objectifs glycémiques
Diabète de type 1	<ul style="list-style-type: none"><li>• Avant les repas : 70 à 120 mg/dl</li><li>• En postprandial : &lt; 160 mg/dl</li></ul>
Diabète de type 2	<ul style="list-style-type: none"><li>• Avant les repas : 70 à 120 mg/dl</li><li>• En postprandial : &lt; 180 mg/dl</li></ul>
Diabète gestationnel	<ul style="list-style-type: none"><li>• A jeun : &lt; 95 mg/dl</li><li>• En postprandial : &lt; 120 mg/dl</li></ul>

**Tableau 1** : Les objectifs glycémiques [L.Makhlouf 2015]

Les diabétiques doivent régulièrement mesurer et contrôler leur glycémie plusieurs fois par jours et cela se fait dans les cas suivants :

- A jeun avant le repas et l'injection d'insuline.
- En cas de malaise ou de signes inhabituels.
- Avant de se coucher.
- Avant, pendant et après une activité physique.

Il est important de bien préparer les patients et de les éduquer à l'auto-surveillance glycémique afin d'optimiser le traitement par insulinothérapie.

### **2.3 L'insulinothérapie**

L'insulinothérapie correspond à l'injection de l'insuline qui est utilisée comme traitement du diabète pour les diabétiques de type 1 et les diabétiques de type 2, elle peut également être prescrite, dans certains cas et de façon temporaire comme le diabète gestationnel[**FFD 2018**], l'injection de l'insuline dépend de l'état physique du patient diabétique, de sa nutrition et de sa glycémie, ce qui définit plusieurs schéma glycémiques.

#### **2.3.1 Définition de l'insuline**

L'insuline est une hormone naturellement sécrétée par le pancréas, plus précisément par des cellules spécialisées situées dans les îlots de Langerhans. Elle permet au glucose d'entrer dans les cellules du corps. Celles-ci utiliseront le glucose comme énergie ou le mettront en réserve pour une utilisation future[**Tremblay et all 2015**]. La **Figure1.1** montre la structure du pancréas et plus précisément les Ilots de Langerhans qui se composent des cellules alpha et beta, ils sont responsables de la sécrétions de l'insuline :

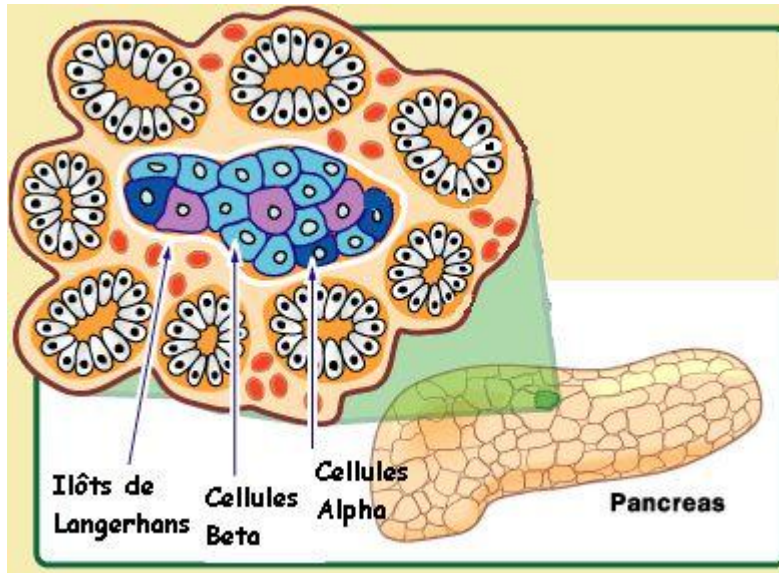


Figure1.1: le pancréas [M.zerouki 2015]

### 2.3.2 Rôle et fonctionnement de l'insuline

L'insuline joue un rôle de régulateur en maintenant la glycémie à des valeurs normales, donc l'organisme produit la quantité nécessaire d'insuline en fonction de ses besoins et des aliments consommés. Mais chez la personne diabétique de type 1, le pancréas ne produit pas d'insuline c'est pourquoi elle doit s'injecter de l'insuline plusieurs fois par jour afin d'imiter le fonctionnement normal du pancréas. Par contre chez la personne diabétique de type 2, le pancréas produit encore de l'insuline, mais celle-ci n'est pas bien utilisée par le corps. Par conséquent, la personne a souvent besoin de médication antidiabétique. Parfois, elle doit également s'injecter de l'insuline[Yolanda Smith 2015].

### 2.3.3 Les différentes familles d'insuline

L'insuline lors de son injection dans le corps humain est caractérisé par son début d'action qui signifie le temps nécessaire dès l'injection pour que ce dernier commence à produire un effet dans l'organisme, un autre paramètre appelé la période d'activité maximale, c'est la période où l'effet de l'insuline est dans un état maximal, et aussi la fin d'action qui signifie le délai de disparition de l'effet de l'insuline dans l'organisme, à partir de ces trois paramètres on distingue sept familles de l'insuline qui sont résumées dans le **Tableau2**.

Les familles d'insuline	Début d'action	Période d'activité maximale	Fin d'action
Ultra-rapides	5-15 min	1-3heures	4heures
Rapides	30min	1h30-4heures	6heures
Semi-Lentes NPH	1heure	2-8heures	12heures
Semi-Lentes zinc	2heures	6-12heures	14-16heures
Lentes Zinc	5heures	6-20heures	24heures

**Tableau2** : Les familles de l'insuline[P. Benhamou 2005]

### 2.3.4 Les schémas insuliniques

Quel que soit le schéma d'insulinothérapie, l'objectif demeure le même : limiter les trop grandes variations de la glycémie. Différents schémas d'injections (de 1 à 5 injections par jour) peuvent être prescrits par le médecin, en fonction du type de diabète, des besoins en insuline et du mode de vie. Dans le cas d'un schéma basal-bolus :

- L'insuline basale, aussi appelée insuline pour vivre, correspond à une ou deux injections d'insuline lente.
- L'insuline pour manger (insuline prandiale) est apportée par un bolus, c'est à dire une injection d'insuline rapide au moment des repas, en fonction du nombre de glucides ingérés.
- L'insuline de correction permet, grâce à une injection d'insuline rapide, de corriger la glycémie si elle est trop élevée[FFD 2018].

## 2.4 Les complications

Les complications liées au diabète peuvent être réparties en deux groupes [Y.J.Drabo 1996]:

- Les complications aiguës : soit métaboliques tel que l'obésité abdominale, hypertriglycéridémie, HDL-cholestérol bas ; soit infectieuses tel que les infections cutanés, pulmonaires, dentaires et urinaires.
- Les complications chronique dégénératives : soit micro-angiopathie tel que les rétinopathies et néphropathies compliquées d'insuffisance rénale ; soit macro-angiopathie dominées par les gangrènes, les atteintes cardiaques et neurologiques.

La grande fréquence de ces complications témoigne des difficultés de prise en charge, de la sous-information et de l'éducation insuffisante des malades.

## 2.5 L'éducation thérapeutique des patients diabétiques

Selon l'organisation mondiale de la santé: L'éducation thérapeutique doit permettre aux patients d'acquérir et de conserver les compétences les aidant à vivre de manière optimale avec leur maladie. Il s'agit d'un processus permanent, intégré dans les soins, et centré sur le patient. Elle vise à aider les patients et leurs familles à comprendre la maladie et le traitement, coopérer avec les soignants, vivre plus sainement et maintenir ou améliorer leur qualité de vie [OMS 1998].

Partant de ce principe, l'importance de l'éducation thérapeutique est reconnue au cours du diabète de type 1, diabète de type2 et du diabète gestationnel, il s'agit d'un outil indispensable dans la prise en charge globale définit par un programme personnalisé avec priorité d'apprentissage qui vise à donner au patient des notions et des généralités pouvant affecter le maintien de sa maladie tel que la nutrition, l'activité physique et le traitement médicamenteux. **Tableau 3** : est un résumé des différentes généralités qui doivent être acquises par le patient pour l'équilibre du diabète.



Généralité sur le diabète et les traitements	<ul style="list-style-type: none"> <li>- Comprendre la maladie pour mieux se prendre en charge.</li> <li>- Présentation des différents traitements et leur mode d'action.</li> </ul>
Alimentation et diabète	<ul style="list-style-type: none"> <li>- Connaître les familles d'aliments et leur influence sur la glycémie afin de se sentir « plus libre dans son assiette ».</li> </ul>
Activité physique	<ul style="list-style-type: none"> <li>- Se rendre compte que l'activité physique et l'exercice régulier sont des changements de style de vie qui doivent être maintenus à long terme pour conserver le même poids.</li> </ul>
Auto surveillance hypoglycémie	<ul style="list-style-type: none"> <li>- Savoir utiliser de manière appropriée le matériel d'autocontrôle.</li> <li>- Reconnaître une hypoglycémie et agir en conséquence.</li> </ul>
Hyperglycémie et complication	<ul style="list-style-type: none"> <li>- Savoir réagir en cas d'hyperglycémie</li> <li>- Connaître les complications et les prévenir.</li> </ul>
Le pied diabétique	<ul style="list-style-type: none"> <li>- Acquérir des connaissances sur les soins des pieds.</li> </ul>

**Tableau 3** : Les notions nécessaires pour l'équilibre du diabète

**[A. Grimaldi et all 2009]**

## **Conclusion**

Dans cette première partie, nous avons essayé dans un premier temps d'aborder la maladie du diabète, ses types et les techniques de traitement ainsi que les complications liées au diabète. Puis nous avons parlé de l'éducation thérapeutique des patients diabétiques. Nous avons également mentionné les généralités nécessaires dans l'accompagnement thérapeutique et éducatif des patients diabétiques.

## **Chapitre 2 : ETAT DE L'ART**

### 1.INTRODUCTION

Le développement actuel des Nouvelles Technologies de l'Information et de la Communication (NTIC) et leurs utilisations dans le domaine de santé nous conduit à nous intéresser à la place occupée par ce media dans l'accompagnement thérapeutique et éducatif des patients diabétiques.

Dans ce chapitre nous allons définir la problématique qui a rendu la maladie de diabète un axe de recherche très important dans le domaine de l'informatique, puis nous allons présenter une classification des différents travaux qui ont été réalisés en faveur des diabétiques de type1 et de type2.

### 2.PROBLEMATIQUE

Le traitement du diabète oblige les patients à surveiller régulièrement leur glycémie et à se rendre chez leur médecin pour réajuster le traitement et comme les consultations n'ont lieu environ que tous les six mois, cela complique considérablement l'analyse à posteriori des prises alimentaires, de l'activité physique et de tous les paramètres qui ont pu jouer sur la glycémie des patients au cours d'une si longue période, par conséquence rares sont les malades qui adaptent correctement leurs traitements.

### 3.LES TRAVAUX REALISEES DANS LE DIABETE DE TYPE1

L'objectif de la télémédecine dans le diabète de type1 est d'aider le patient à mieux contrôler sa glycémie par des conseils précis sur les ajustements de ses doses d'insuline. [Franc1 2011].Trois types d'expériences ont été rapportés :

#### 3.1 Consultation téléphonique

La consultation téléphonique est la méthode de télémédecine la plus simple et la plus répandue, tel que le système d'appelle dure environ 12 à 16 minutes, l'infirmier recueille certains renseignements personnels, il évalue le besoin du patient puis il reprend à ce dernier parfois il oriente le patient vers une ressource appropriée, si nécessaire [Dr.G.Charpentier 2009].

### 3.2 Télétransmission des données glycémiques

Les systèmes de télétransmission des données glycémiques ont été développés au cours de ces dernières années, le plus simple repose sur la transmission des données enregistrées dans la mémoire d'un lecteur glycémique vers le soignant, le feedback du soignant au patient est assuré par sms ou email [Riveline et al 2007]. Des systèmes élaborés permettent au patient de transmettre à partir d'un ordinateur fixe ses données glycémiques mais aussi d'autre type de données tel que les doses d'insulines et des informations sur son alimentation, ces données sont ensuite transmises au soignant et le feedback étant assuré par internet. A titre d'exemple le système T-IDDM [Bellazzi et al 2002]. Un autre système Gluconet qui permet au patient de transmettre leurs données glycémiques toutes les semaines via un PDAphone chargé du logiciel qui sert à un organisateur électronique auquel on a intégré les fonctionnalités d'un téléphone portable, ici le patient reçoit de la part d'un diabétologue expérimenté des indications par sms pour optimiser ses doses d'insuline [Benhamou PY et al 2007].

### 3.3 Les carnets électroniques actifs sur SmartPhone

Ce sont des systèmes plus élaborés de type Smartphone couplés à un site web, le carnet glycémique est appelé actif car il propose une dose d'insuline calculée à partir d'un algorithme prescrit par le médecin, ce type de système permet un feedback plus au moins plus élaboré, un certain nombre de ces systèmes a vu le jour récemment tel que DID (Diabetes Interactive Diary) qui comporte un programme qui adapte les doses d'insuline de chaque patient aux objectifs glycémiques préétablis, et aussi une liste alimentaire électronique dans laquelle le patient peut sélectionner le type et la quantité d'aliments qu'il compte consommer, ainsi DID calcule automatiquement la quantité de glucides correspondante et détermine le nombre d'unités d'insuline à administrer, la connexion avec le médecin se fait par sms [C.E. Rossi et al 2010]. Le système DIABIO est un carnet électronique sur Smartphone qui permet le calcul automatique des doses d'insuline du repas en intégrant trois paramètres glycémie préprandiale, quantité de glucides ingérée et pratique ou non d'une activité physique, il propose une modification des règles d'adaptation des doses d'insuline selon les objectifs glycémiques définis par le médecin, transmise via GPRS sur un site Web sécurisé. Consultable par le personnel soignant autorisé. Le soignant peut ainsi assurer une télésurveillance des données et proposer si besoin des consultations téléphoniques courtes mais répétées

### 4. LES TRAVAUX REALISEES DANS LE DIABETE DE TYPE2

Le diabète de type2 repose classiquement sur le trépied alimentation, activité physique et traitement antidiabétique, donc l'objectif de la télémédecine sera ici non pas tant d'aider le patient à ajuster son traitement que de l'encourager à modifier son comportement (contrôle de l'apport calorique, augmentation de l'activité physique. . .) et de le soutenir dans cette démarche[Daoudi1 et al 2011].

#### 4.1 Consultation téléphonique

Dans le diabète de type 2, le suivi téléphonique par une IDE (Infirmière Diplômée d'Etat) a montré un bénéfice sur le contrôle glycémique chez des patients ayant un accès aux soins limité mais un diabète pas trop déséquilibré [McMahon et al 2005], en France le programme Sophia lancé par la Caisse nationale d'Assurance-maladie propose aux patients jugés à risque intermédiaire ou élevé des appels téléphoniques par une IDE qui avait reçu une formation au préalable [ameli.fr].

#### 4.2 Systèmes reposant sur un siteWeb interactif

Les systèmes qui reposent sur un site web interactif permettent aux patients de saisir leurs résultats glycémiques ainsi que certains paramètres cliniques tel que le poids et des informations relatives à leur traitement tel que les doses d'insuline, posologie d'antidiabétiques oraux, ces données sont consultables sur le siteWeb par le soignant autorisé, en même temps les patients peuvent interroger le soignant via Internet et recevoir en retour des recommandations. Indigo-diabète est un site web interactif et gratuit, dédié à l'éducation thérapeutique pour faciliter la vie des diabétiques de type1 [indigo-diabete.fr].

#### 4.3 Systèmes couplant téléphone cellulaire et serveur à distance

Dans ce type de système les données de glycémie capillaire du lecteur sont transmises par Bluetooth du lecteur de glycémies vers un téléphone portable puis du téléphone portable vers le serveur à distance. Des messages automatiques sont générés en retour selon les recommandations du soignant et qui peut visualiser les données du patient sur son écran d'ordinateur et le contacter si besoin, dans ce cadre le système WellDoc qui permet un feedback immédiat vers le patient sous la forme de messages

automatiques et la possibilité de recevoir les conseils d'une infirmière sur les adaptations du traitement.

La **Figure 2.1** montre la transmission par Bluetooth du lecteur de glycémies vers les systèmes WellDoc qui génère un message automatique informant le patient sur sa glycémie en lui expliquant qu'il s'agit d'une hypoglycémie [**Comstock 2018**].



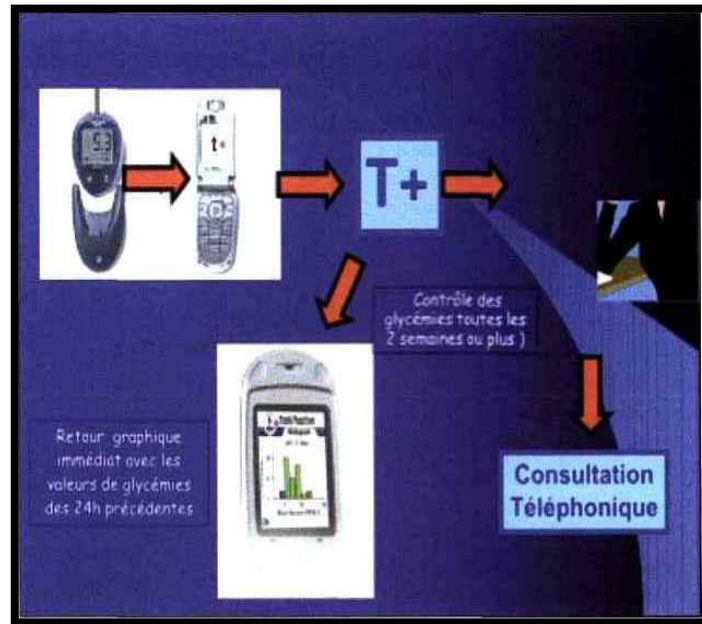
**Figure 2.1:** la transmission par Bluetooth du lecteur de glycémies vers les systèmes WellDoc [**Comstock 2018**].

#### 4.4 Carnets électroniques de type SmartPhone

Il existe des différents types de système, ceux qui ont privilégié l'adaptation du traitement antidiabétique, et ceux qui ont ciblé une modification comportementale en ce qui concerne l'alimentation et l'activité physique. Les expériences qui ont visé l'adaptation du traitement, le système T+, précédemment décrit, dans une version adaptée aux patients DT2, ce système émettait automatiquement une recommandation d'augmentation de la dose d'insuline selon la prescription du médecin, tant que les glycémies restaient au-dessus de l'objectif fixé [**G. Charpentier et al 2010**].

La **Figure 2.2** montre le fonctionnement du système T+, la télétransmission des glycémies à partir de la mémoire d'un lecteur glycémique via Bluetooth et téléphone portable. Conseils en retour en deux temps : immédiat, par un graphique automatique

des glycémies, et retardé, par un soignant au téléphone qui conseille l'adaptation rétrospective des doses d'insuline



**Figure 2.2 :** le fonctionnement du système T+[G. Charpentier et al 2010].

Parmi les systèmes ciblant la modification comportementale, le système DIABIO dans une version adaptée au diabète de type 2, il permet non seulement de faire une proposition automatique de dose selon la prescription médicale mais il peut aussi assurer un coaching au quotidien du patient et lui faire des recommandations sur l'alimentation ou l'activité physique [CNEDIMTS 2016].

## 5. SYNTHÈSE

Ce qu'on vient de voir n'est qu'un survol de quelques-unes des applications utilisées dans le monde de télé-diabète qui reste très captivant, passionnant et prometteur. Toutefois, malgré que la liste ne soit pas exhaustive, ces expériences dont nous avons parlé nous ont ouvert des perspectives pour la conception de notre application qui sert à un outil d'aide au suivi du diabète de type 1 et de type 2 permettant aux patients diabétiques de bénéficier d'une application qui va regrouper la plupart des méthodes précédemment définies, tel que le carnet électronique actif qui permet le calcul des doses d'insuline du repas associé à des recommandations sur l'alimentation ou l'activité physique ainsi que la consultation téléphonique, et cela dans le but de conserver un lien thérapeutique dans l'espace entre les consultations au cabinet.



### Conclusion

Dans ce deuxième chapitre nous avons essayé de lever le voile sur quelques expériences qui ont ouvert de larges perspectives dans la prise en charge des patients diabétiques, nous les avons regroupé dans deux catégories les expériences réalisées dans le diabète de type1 qui visent à optimiser le traitement insulinique ainsi que les expériences réalisées dans le diabète de type2 et qui ont pour objectif d'induire une modification comportementale et à la maintenir dans le temps. Ces modalités de suivi rendues possibles par la télémédecine devraient permettre une économie de temps pour le médecin mais aussi pour le patient qui n'aura plus à se déplacer systématiquement pour ses consultations. Par la suite nous avons présenté brièvement notre travail qui sert à une application de télédiabète et qui a été développée sous la plateforme JavaEE qui a pour objectif de faciliter le développement d'applications web robustes. Dans le chapitre suivant nous allons présenter notre travail et les composants de la plateforme technique, ainsi que les différentes étapes de la réalisation de l'application partant de la phase de conception jusqu'à l'implémentation.

**Chapitre 3 : CONCEPTION ET IMPLEMENTATION**

### 1. INTRODUCTION

Dans le monde actuel de l'informatique et des technologies Internet, les applications sont de plus en plus conséquentes, rapides et doivent être réalisées en un minimum de temps et de coût, Sun Microsystems, désormais Oracle Corporation, propose pour cela la plate-forme Java Enterprise Edition ou Java EE, véritable référence pour le développement d'applications d'envergure et supportant en standard la programmation, la persistance des données, les transactions, la sécurité ou encore le déploiement de projets sur serveur applicatif. Cette dernière partie est consacrée à la conception et la réalisation d'une plateforme collaborative de télédiabète. Nous donnerons au début la description de l'application ainsi que la démarche que nous avons suivie pour la conception. Puis nous passerons à la présentation de la plateforme technique nécessaire pour l'implémentation et qui regroupe la plateforme JavaEE, le serveur Tomcat et la base des données Derby. En dernier lieu, nous allons décrire les étapes effectuées durant la programmation [Lafosse 2012].

### 2. PRESENTATION DE L'APPLICATION

L'objectif principal de ce mémoire est la réalisation d'une application web en faveur des diabétiques qui se présente sous le nom de DailyDiab et qui est développée en JavaEE une plateforme logiciel robuste, complète et axée sur le Web. En effet, l'application peut être utilisée comme logiciel de gestion de patients pour suivre en continu et à distance leurs données d'auto-surveillance ou comme outil de télémedecine, et cela pour pouvoir communiquer simplement avec l'équipe médicale qui les suit. DailyDiab se divise alors en 2 modules principaux:

- Module de l'auto-surveillance et de suivi à distance qui inclut : le carnet glycémique, le journal alimentaire, la surveillance et le suivi du traitement (insuline et ADO) du poids et de l'activité physique.
- Module de messagerie entre le patient et son soignant.



Figure 3.1: logo

### 3. FONCTIONNALITES

L'application DailyDiab doit permettre au patient de bénéficier de:

- Carnet Glycémique
- Journal Alimentaire
- Courbe de Poids
- Messagerie texte avec son médecin
- Carnet de suivi de l'activité physique
- Carnet de suivi de l'insulinothérapie et du traitement par ADO

L'application DailyDiab doit permettre au médecin de :

- Accéder au dossier de chaque patient et de contrôler aussi souvent ses relevés d'auto-surveillance glycémique, son carnet alimentaire, sa courbe de poids et les messages qu'il peut lui adresser.

### 4. CONCEPTION

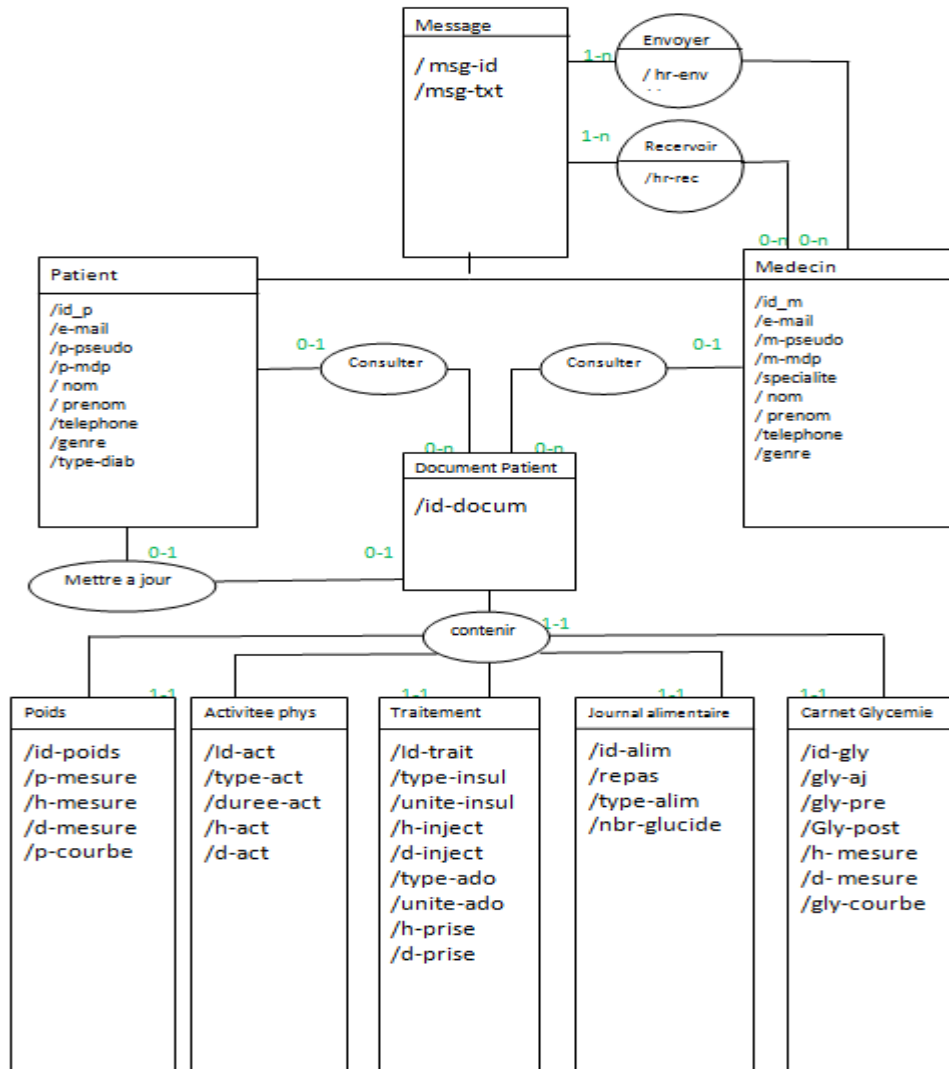
Cette étape essentielle dans la réalisation de l'application va conditionner la qualité, la rapidité et la simplicité du développement, et aussi faciliter la reprise du projet par une autre personne. De toute évidence, nous nous sommes tournées vers une conception objet à l'aide de la modélisation MERISE.

#### 4.1. Modélisation MERISE

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment. (CCM)

- Nous avons réalisé avec MERISE le modèle conceptuel des données (MCD) de l'application qui a pour but d'écrire de façon formelle les données qui seront utilisées. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités:

## Chapitre 3 : CONCEPTION ET IMPLEMENTATION



**Figure 3.2:** MCD

- Ensuite, nous avons fait la transformation du MCD en MLD (Modèle Logique des Données) qui est nécessaire car les structures de données physiques (Fichiers ou Bases de données) impliquent des contraintes logiques.

Voici les tables qui vont construire notre base des données :

- Message ( msg-id, msg-txt )
- Envoyer ( #msg-id, #id\_p, hr-env, d-env )
- Recevoir ( #msg-id, #id\_p, hr-env, d-env )
- Medecin ( id\_m, nom, prenom, date-naissance, e-mail, telephone, genre, specialite, #id-docum )
- Patient ( id\_p, nom, prenom, date-naissance, e-mail, telephone, genre, type-diab, #id-docum )
- Document ( id-docum, dateins )
- Mettre\_a\_jour( #id-docum, # id\_p )

## Chapitre 3 : CONCEPTION ET IMPLEMENTATION

- Carnet\_Glycemie ( id-gly, gly-aj, gly-pre, Gly-post, h- mesure, d- mesure, gly-courbe, #id-docum )
- Journal\_alimentaire ( id-alim, repas, type-alim, nbr-glucide, #id-docum )
- Traitement ( Id-trait, type-insul, unite-insul, h-inject, d-inject, type-ado, unite-ado, h-prise, d-prise, #id-docum )
- Activitee\_phys ( Id-act, type-act, duree-act, h-act, d-act, #id-docum )
- Poids ( id-poids, p-mesure, h-mesure, d-mesure, p-courbe, #id-docum )

### 4.2. Le modèle MVC

Le modèle MVC (Model/View/Controller) est un schéma de programmation qui prend en compte toute l'architecture d'un programme et classe les différents types d'objets qui composent l'application dans 3 catégories:

#### 4.2.1. La vue

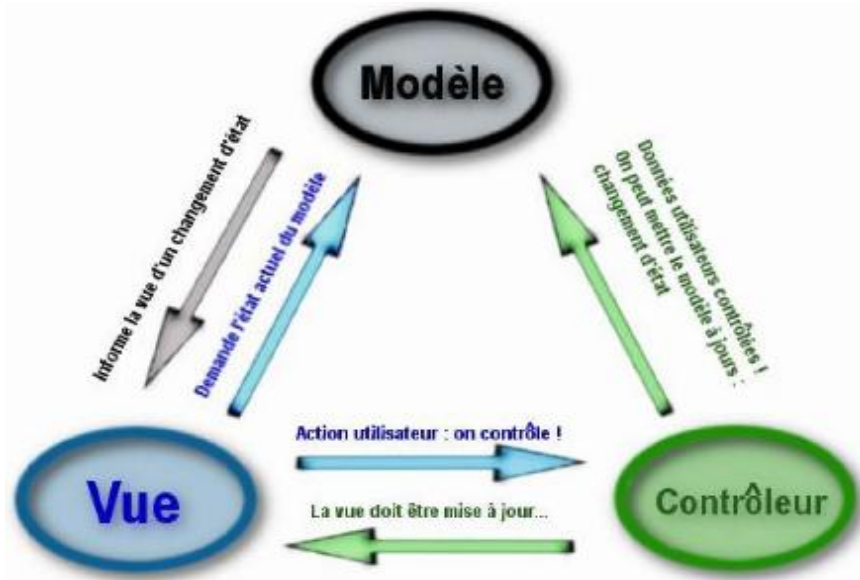
Les vues affichent les résultats provenant des objets "model" et s'assurent que ces données sont correctement affichées.

#### 4.2.2. Le modèle

Les modèles représentent les données de l'application (les bases de données en faisant partie) et définissent la logique de manipulation de ces données. C'est dans cette partie que vont s'effectuer les traitements, on ne s'occupe absolument pas de la mise en forme mais bien des données seules. Dans une application respectant les règles du modèle MVC, les données les plus importantes seront encapsulées dans les objets modèles.

#### 4.2.3. Le contrôleur

Le contrôleur gère l'interaction avec l'utilisateur. Ainsi c'est dans cette partie que va se réaliser l'interaction entre la vue et le modèle. En effet, les objets contrôleur reçoivent les requêtes utilisateur puis détermine quelles parties des objets vues et modèles sont requises. Ils constituent donc l'intermédiaire entre les deux autres types d'objets.



**Figure 3.4:** Le schéma de la structure générique d'une architecture MVC [ARFAOUI 2013].

### 4.2.4. Avantages du modèle MVC

Le modèle MVC impose donc une séparation totale entre le traitement, l'interface et la communication entre ces deux parties. Cela permet d'avoir non seulement des objets réutilisables pour d'autres applications, mais aussi de pouvoir faire évoluer aisément son programme. Ainsi, si l'on souhaite modifier sa base de données, il suffit de revoir son "model" et cela est valable pour le cas où l'on souhaite changer d'interface. Les 3 parties du modèle MVC sont réellement autonomes. Aucune d'elles ne s'occupe du fonctionnement de l'autre [Douiri 2010].

## 5. REALISATION

Dans cette partie nous présentons la plateforme de travail et les étapes du développement du modèle MVC que nous avons choisi comme exemple.

### 5.1. Environnement de développement

Cette partie présente les différents outils et logiciels de développement utilisés avec l'architecture logicielle.

#### 5.1.1. Présentation de la plateforme JavaEE

La plate-forme J2EE (Java 2 Entreprise Edition), est la proposition de SUN pour le développement et la mise en œuvre d'applications d'entreprise multi-niveaux, le terme « Java EE » signifie Java Enterprise Edition, et était anciennement raccourci en « J2EE ». La plate-forme Java EE est construite sur le langage Java et la plate-forme Java SE (Java Standard Edition), et elle y ajoute un grand nombre de bibliothèques remplissant tout un tas de fonctionnalités que la plate-forme standard ne remplit pas d'origine. L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications [GBITHICKI 2013].

### 5.1.1.1. Implémentation du modèle MVC dans JavaEE

Avec la plateforme JEE, chaque élément du modèle MVC porte en quelque sorte un nom. Le Contrôleur porte le nom de Servlet. Le modèle est en général géré par des objets Java ou des JavaBeans. Il peut être amené aussi à communiquer avec les bases de données pour stocker les informations, pour les persister et les garder en mémoire le plus longtemps possible. La Vue quant à elle est gérée par les pages JSP (Java Server Pages) qui sont en effet des pages qui vont utilisées du code HTML et du code spécifique en général en JAVA. Cette Vue est donc retournée au visiteur par le Contrôleur [TEDAJIO SUKI 2018].

#### ○ La servlet

En java EE une servlet est une classe Java ayant la capacité de permettre le traitement des requêtes et de personnaliser les réponses, c'est-à-dire qu'elle est capable de recevoir les requêtes HTTP envoyées depuis le navigateur (Client web) de l'utilisateur et de lui renvoyer une réponse http. Une servlet http doit toujours hériter de la classe HttpServlet qui est en effet une classe abstraite qui fournit des méthodes qui doivent être redéfini dans la classe héritière. Parmi ces méthodes on y retrouve entre autres :

- doGet(), méthode qui permet de récupérer une ressource du serveur coté client (utilisé par le client). Une fois que le serveur reçoit une telle demande, il retourne non seulement la ressource mais également des informations diverses à son sujet. On y retrouve aussi des entêtes (Head http) des informations telles que la longueur des données renvoyées ou la date d'envoi.
- doPost(), c'est une méthode qui permet de soumettre au serveur des données de tailles variables, ou que l'on sait volumineuses.
- doHead(), c'est une méthode identique à la méthode doGet à ceci près que le serveur que le serveur n'y répondra pas en renvoyant les ressources accompagnées des informations la concernant, mais seulement ces informations.

#### ○ Les JSP

Une page JSP est destinée à la vue (présentation des données). Elle est exécutée côté serveur et permet l'écriture des pages. C'est un document qui à première vue, ressemble beaucoup à une page HTML, mais qui en réalité diffère par plusieurs aspects :

- L'extension d'une telle page est .jsp et non .html
- Une page JSP contient des balises HTML mais aussi des balises JSP qui appellent de manière transparente du code JAVA.
- Contrairement à une page HTML, une page JSP est exécutée côté serveur et génère alors une page renvoyée au client.



## Chapitre 3 : CONCEPTION ET IMPLEMENTATION

L'intérêt est de rendre possible la création des pages dynamiques (comme en PHP). Avec une page JSP, on peut générer n'importe quel type de format, HTML, CSS, XML, du texte brut etc. Une page utilisant les Java Server Pages est exécutée au moment de la requête par un moteur de JSP fonctionnant généralement avec un serveur web ou un serveur d'application. Lorsqu'un utilisateur appelle une page JSP pour la première fois, le serveur web appelle le moteur JSP qui crée un programme Java à partir du script JSP, compile la classe afin de fournir un fichier compilé (extension .class). Au prochain appel, le moteur de JSP vérifie si la date du fichier .jsp correspond à celle du fichier .class, si elle est identique, le moteur de JSP n'effectuera aucune compilation et dans le cas contraire la compilation sera de nouveau faite. En d'autres termes le moteur JSP ne transforme et compile la classe que dans le cas où le script a été mis à jour. Ainsi le fait que la compilation ne se fasse que lors de mise à jour ou de la modification du script JSP fait de cette technologie une des plus rapides pour créer des pages dynamiques. En effet la plupart des technologies de pages actives (PHP par exemple), repose sur l'interprétation du code ce qui requiert beaucoup plus de ressource pour produire une réponse http. Etant donné que les JSP sont compilées puisqu'il s'agit en fait du byte code, elles sont beaucoup plus rapides à l'exécution.

### ○ Les JavaBeans

Les Javabeans sont un modèle de composants du langage Java. Ce sont des 'pièces logicielles' qui peuvent être réutilisées pour créer des programmes dans un environnement visuel de développement d'applications. Elles représentent généralement les données du monde réel. Les principaux concepts mis en jeu pour un Bean sont les suivants :

- **Propriétés** : Un JavaBean doit pouvoir être paramétrable. Les propriétés c'est-à-dire les champs non publics présent dans un bean. Qu'elles soient de type primitif ou objets, les propriétés permettent de paramétrer le bean, en y stockant des données.
- **La sérialisation** : Un bean est construit pour pouvoir être persistant. La sérialisation est un processus qui permet de sauvegarder l'état d'un bean et donne ainsi la possibilité de le restaurer par la suite. Ce mécanisme permet une persistance de données voir de l'application elle-même.
- **La réutilisation** : Un bean est un composant conçu pour être réutilisable. Ne contenant que des données du code métier, un tel composant n'a en effet pas de lien direct avec la couche de présentation, et peut également être distant de la couche d'accès aux données. C'est cette indépendance qui lui donne ce caractère réutilisable.
- **L'introspection** : un bean est conçu pour être paramétrable de manière dynamique. L'introspection est un processus qui permet de connaître le contenu d'un composant (attributs, méthodes et événements) de manière dynamique, sans disposer de son code source. C'est ce processus, couplé à

## Chapitre 3 : CONCEPTION ET IMPLEMENTATION

certaines règles de normalisation, qui rend possible une découverte et un paramétrage dynamique du bean.

Ainsi tout objet conforme à ces quelques règles peut être appelé Bean. Il est également important de noter qu'un JavaBean n'est pas un EJB (Entreprise Java Bean). Un bean doit également avoir la structure suivante :

- Doit être une classe publique
- Doit avoir au moins un constructeur par défaut, public et sans paramètres. Java l'ajoutera de lui-même si aucun constructeur n'est explicité ;
- Peut implémenter l'interface serialisable, il devient ainsi persistant et son état peut être sauvegardé ;
- Ne doit pas avoir de champs publics ;
- Peut définir des propriétés (des champs non publics), qui doivent être accessibles via des méthodes publiques getter et setter, suivant des règles de nommage.

### 5.1.1.2. Les bibliothèques JSTL

JSTL est l'acronyme de Java server page Standard Tag Library. C'est un ensemble de tags personnalisés développé sous la JSR qui propose des fonctionnalités souvent rencontrées dans les JSP.

JSTL possède quatre bibliothèques de tags :

- La bibliothèque c.tld : Tag de structure (itération, conditionnement ...)
- La bibliothèque x.tld : Utilisation de documents XML
- La bibliothèque fmt.tld : Internationalisation
- La bibliothèque sql.tld : Exécution de requêtes SQL

JSTL propose un langage particulier constitué d'expressions qui permettent d'utiliser et de faire référence à des objets Java accessibles dans les différents contextes de la page JSP. Le but est de fournir un moyen simple d'accéder aux données nécessaires à une JSP [Jean-Michel 2016].

### 5.1.2. Apache Tomcat

Apache TOMCAT est le conteneur de servlet fourni par la communauté open source Apache Foundation Software. Conteneur "Open Source" de Servlet J2EE. Il fait parti du projet Jakarta au sein de la fondation Apache. Tomcat implémente les spécifications des Servlets et des JSP de Sun Microsystems. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP. La 1er version produite est la version 3, en 1999, TOMCAT est le produit de la fusion du projet Java

## Chapitre 3 : CONCEPTION ET IMPLEMENTATION

Web Server de Sun et du projet JServ de la communauté Apache, en 2001, la version 4 est sorti sous le nom de code Catalina et qui représentait une refonte total du produit

- Jasper : Le compilateur qui traduit la JSP en Servlet
- Coyote : Le connecteur HTTP qui permet à TOMCAT de servir comme serveur http
- Catalina : Le noyau de TOMCAT, Servlet API, Administration, Sécurité, ...

### 5.1.3. Apache Derby

Apache Derby est une base de données relationnelle open source entièrement développée en Java par la fondation Apache, elle peut être embarqué dans des programmes écrits en Java. Derby a la particularité de pouvoir être utilisé comme gestionnaire de base de données embarqué dans une application Java. Ce qui rend inutile l'installation et la maintenance d'un serveur de base de données autonome. A l'inverse Derby supporte aussi un mode de fonctionnement client- serveur [Leroux 2018].

## 5.2. Mise en application

Dans cette partie nous allons décrire les étapes de l'implémentation de l'application web DailyDiab.

### 5.2.1. Installation des logiciels nécessaires

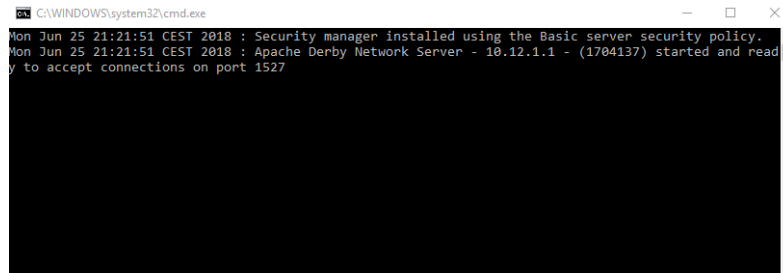
- Eclipse IDE pour Développeurs Java EE, version: Oxygen.2, édition (4.7.2): Il rassemble tous les composants et les packages de la plateforme Web Tools et leurs dépendances dans une plateforme pratique.

- Derby, version :10.12.1.1
- Tomcat, version : 6.0.50

### 5.2.2. Création de la base des données avec Derby

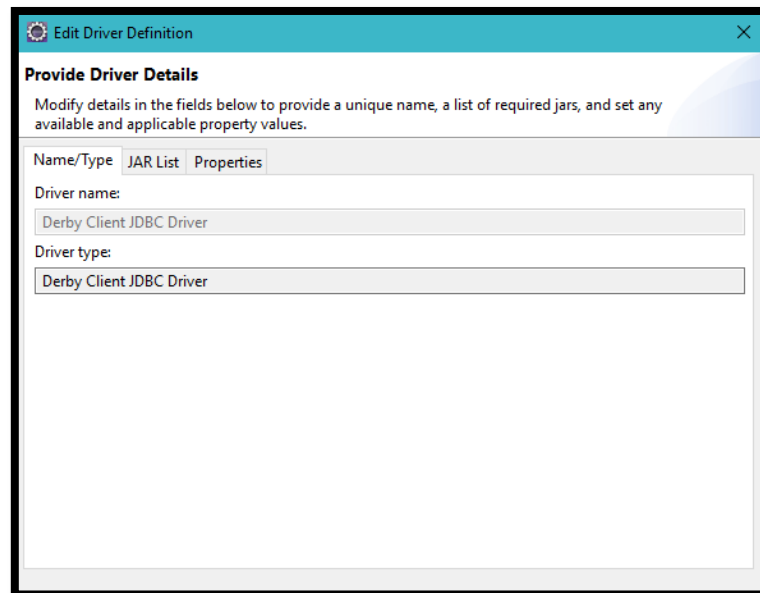
- Comme la plupart des SGBD populaires, Derby a une architecture client / serveur. Le moteur Derby fonctionne comme un processus de serveur, acceptant connexions à partir d'applications client.
- Pour utiliser Derby, nous démarrons le serveur Derby, puis nous utilisons les outils Java de gestion de base de données (Java database management tools) pour se connecter au serveur Derby, créer et remplir la base des données, exécuter des requêtes, etc.

## Chapitre 3 : CONCEPTION ET IMPLEMENTATION



**Figure 3.4 :** serveur Derby

- Nous commençons par la création d'un projet de base des données Derby dans Eclipse qui va contenir notre base des données.
- Ajouter le Pilote JDBC Client Derby et le fichier .jar nécessaire (derbyclient.jar) pour faire la connexion dans Eclipse.



**Figure 3.5 :** pilote JDBC Client Derby

- Pour créer la base des données nous avons besoin d'un SGBD Java pour faire la connexion avec le serveur Derby, pour cela nous avons choisit « SQL Model JDBC Connection ». Puis nous choisissons le pilote JDBC du client Derby, définissons le nom de la base des données "), utilisons un ID utilisateur non vide et un mot de passe (le mot de passe est ignoré par Derby).

## Chapitre 3 : CONCEPTION ET IMPLEMENTATION

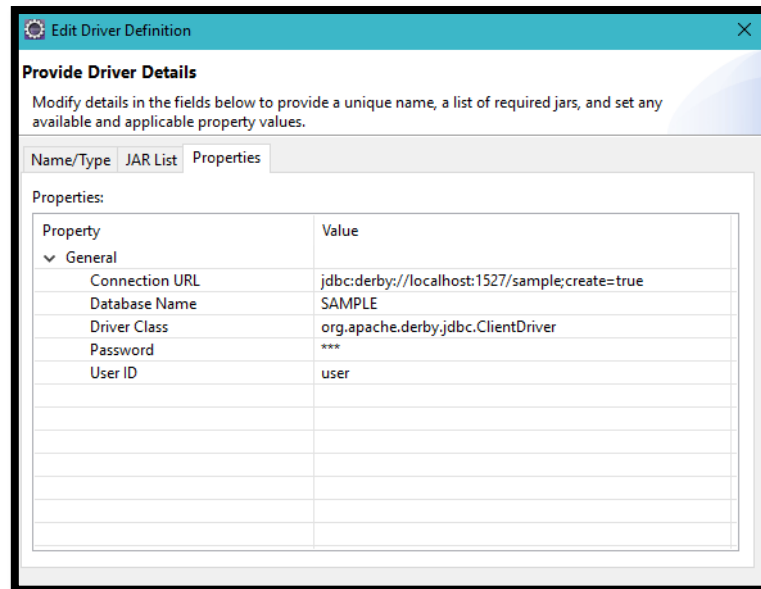


Figure 3.6 : Création de la base de données

- Maintenant que la base des données est créée, il est temps de créer des tables pour notre application, nous avons donc créé les tables à partir de notre MLD et à l'aide de l'outil « SQL Data Definition commands ».

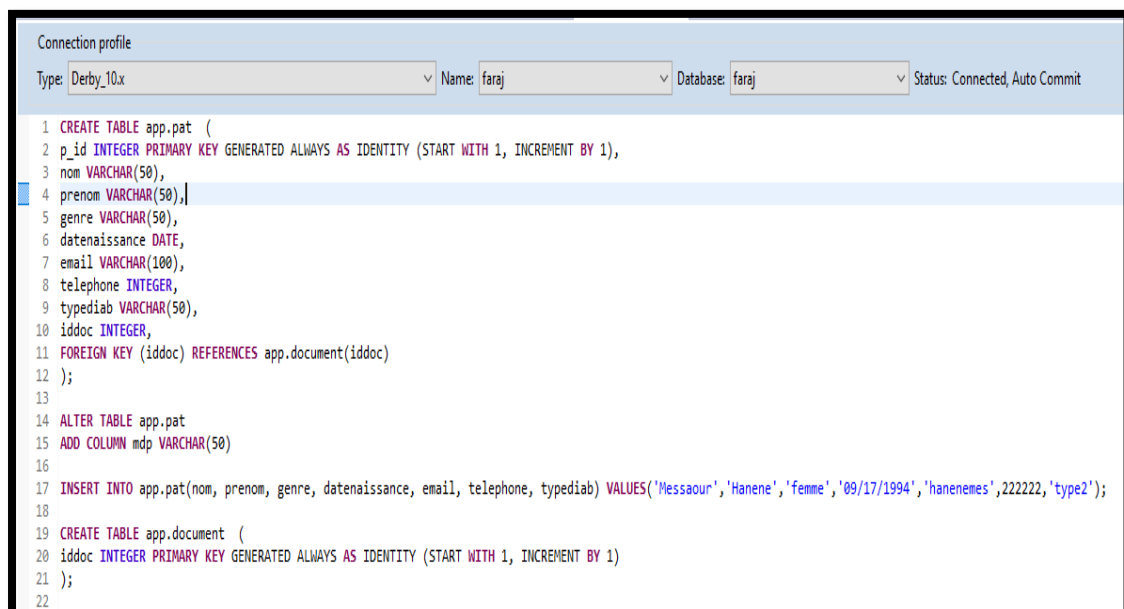


Figure 3.7 : Création des tables avec SQL Data Definition commands

### 5.2.3. Création du projet Web et connexion avec la base des données

Eclipse nous aide à organiser notre applications utilisant un type de projet appelé un projet Web dynamique

- Nous avons d'abord sélectionné Tomcat 6.0 comme un serveur d'applications Web, qui va fournir les bibliothèques nécessaires au projet.
- Notre application Web consiste en :

## Chapitre 3 : CONCEPTION ET IMPLEMENTATION

- une collection de ressources dynamiques (telles que Servlets, JavaServer Pages et d'autres classes Java)

```
package com.as.samples;

import java.io.IOException;

/**
 * Servlet implementation class pageOneProcess
 */
public class pageOneProcess extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        String name=request.getParameter("name");
        String add=request.getParameter("add");
        String phone=request.getParameter("phone");

    }
}
```

Figure 3.8 : Exemple du page servlet

```
!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>DailyDiab</title>
<link rel="stylesheet" type="text/css"
href="${pageContext.request.contextPath}/jsp/style1.css" />
</head>
<body>
<div class="header">

<h1><i>Dailydiab</i></h1>
</div>
<ul>
<li><a href="http://localhost:8085/webapp-01/jsp/printout.jsp"><i></i></a>
</li>
</ul>
<div class="row">
<div class="column">

</div>
</div>
```

Figure 3.9 : Exemple du page JSP

- Des ressources statiques (les pages HTML et CSS et les images)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>

</head>
<body>
<h1>Details</h1>
<form method="post" action="pageOneProcess">
Name:<input type="text" name="name">
</br>
Adresse: <input type="text" name="add">
</br>
Phone: <input type="text" name="phone">
</br>
<input type="submit" value="submit">
</form>
</body>
</html>
```

Figure 3.10 : exemple d'un fichier HTML en JavaEE

```
body
{
background-color:#ddeeff;
}

.item1 { grid-area: header;}
.item2 { grid-area: menu; }
.item3 { grid-area: main; }
.item4 { grid-area: right; }
.item5 { grid-area: footer; }

.grid-container {
display: grid;
grid-template-areas: 'header header header header header header'
'menu main main main main right right'
'menu footer footer footer footer footer';
grid-gap: 10px;
background-color: #ddeeff;
padding: 10px;
}

.grid-container > div {
background-color:rgba(255,255,255,0.8);
text-align:center;
padding:20px 0;
```

Figure 3.11 : exemple d'un fichier css en JavaEE

- Les fichiers de configuration (les fichiers XML)

## Chapitre 3 : CONCEPTION ET IMPLEMENTATION

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.co
<display-name>Daily</display-name>
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>index.css</welcome-file>
  <welcome-file>index.xml</welcome-file>
  <welcome-file>default.html</welcome-file>
  <welcome-file>default.htm</welcome-file>
  <welcome-file>default.jsp</welcome-file>
  <welcome-file>default.css</welcome-file>
  <welcome-file>default.xml</welcome-file>
</welcome-file-list>
<servlet>

  <display-name> DWR Servlet </display-name>
  <servlet-name> dwr-invoker </servlet-name>
  <servlet-class> org.directwebremoting.servlet.DwrServlet </servlet-class>
  <init-param>

    <param-name> debug </param-name>
```

Figure 3.11: Exemple d'un fichier de configuration

- Pour permettre aux utilisateurs d'interagir avec la base de données, voici nos interfaces :

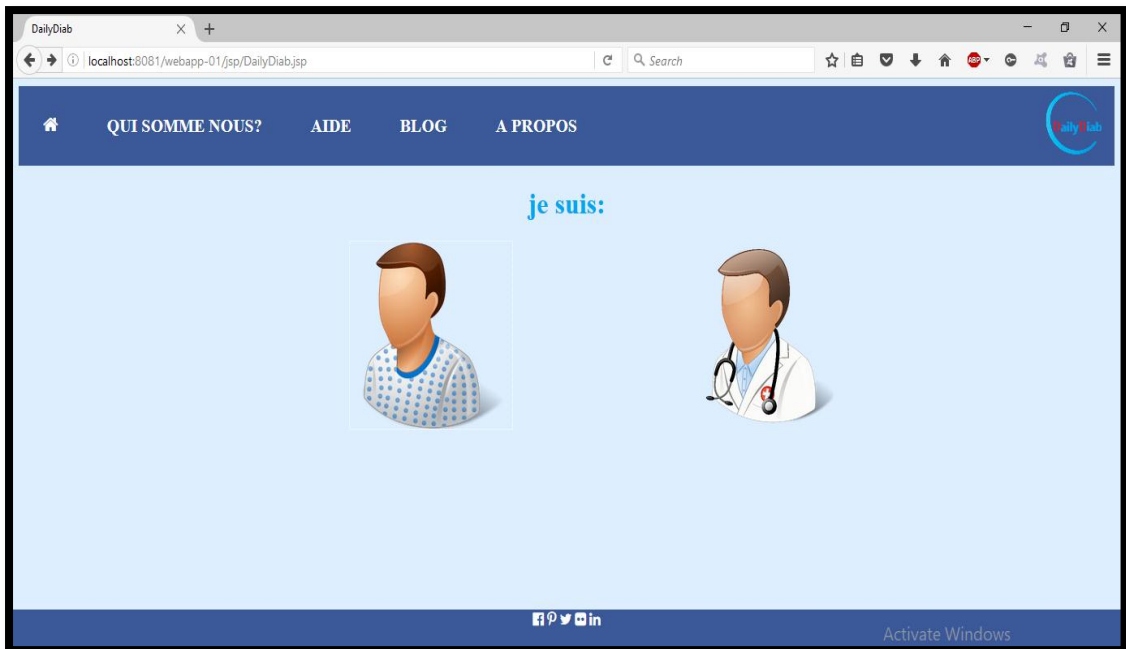


Figure 3.12: La page d'accueil.



Figure 3.13 shows the login page. The header contains navigation links: 'QUI SOMME NOUS?', 'AIDE', 'BLOG', 'A PROPOS', and 'INSCRIPTION'. The main content area is titled 'identification' and contains two input fields: 'Adresse email' and 'Mot de passe'. A blue 'Login' button is positioned below the fields. A link 'mot de passe oublier!' is located at the bottom of the form area. On the left side, there are social media icons for Facebook, Twitter, and Google+.

**Figure 3.13:** Page de connexion.

Figure 3.14 shows the registration page. The header contains navigation links: 'QUI SOMME NOUS?', 'AIDE', 'BLOG', 'A PROPOS', and 'INSCRIPTION'. The main content area is titled 'Inscription:'. Below the title, there are four input fields: 'Nom...', 'Prenom...', 'Genre...', and 'Type de diabete...'. A blue 'Next' button is positioned at the bottom right of the form area. The footer contains the URL 'ocalhost:8081/webapp-01/jsp/DailyDiab.jsp' and social media icons for Facebook, Twitter, and LinkedIn.

**Figure 3.14:** page d'inscription.

- Afin de connecter la base des données avec les pages JSP nous allons utiliser La bibliothèque JSTL qui va fournir toutes les fonctionnalités dont une simple application de base de données a besoin, puis ajouter les fichiers .jar nécessaires (jstl.jar et standard.jar) dans notre projet web.

```
2 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

**Figure 3.15:** utilisation de la bibliothèque JSTL.

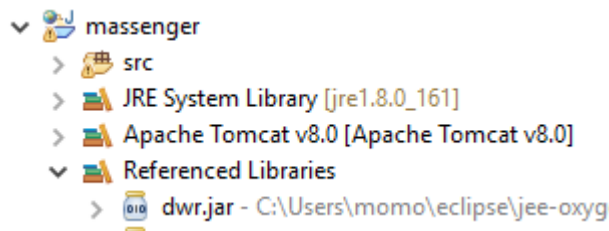
### 5.2.4. Utilisation de la bibliothèque DWR pour créer un service de Chat

DWR (Direct Web Remoting) est une bibliothèque open-source destinée à faciliter l'écriture d'applications AJAX, elle n'est pas vraiment une technologie client ni une technologie serveur mais le pont ("glue") entre les deux.

- DWR masque les couches de "bas niveau" pour les échanges entre le client et le serveur via l'objet XMLHttpRequest
- Prend en charge les communications entre JavaScript (client) et java (serveur)
- Le code JavaScript peut invoquer "directement" des méthodes java s'exécutant depuis le serveur

Les étapes suivantes vont expliquer comment exploiter les fonctionnalités de DWR afin de réaliser le module de Chat dans notre application :

- Nous avons commencé par la création d'un nouveau projet pour notre service de Chat, puis ajouté le fichier .jar « dwr.jar » dans la bibliothèque du projet (lib) et activé le dwr à l'aide d'un fichier de configuration XML.



**Figure 3.16 :** ajout du fichier DWR.jar.

- L'étape suivante est la création des fichiers de présentation, css et html/jsp pour le développement de l'interface utilisateur.
- L'écran de connexion utilise la fonctionnalité DWR pour traiter la connexion de l'utilisateur (les classes Java sont présentées après les pages Web). La fonction loginResults ouvre la page d'échec ou la page principale en fonction du résultat de l'opération de connexion.
- Si la connexion a échoué, une page loginFailed.html très simple est montrée à l'utilisateur.
- La page principale, mainpage.jsp, inclut toute la logique côté client de notre application ChatRoom.
- La première fonction est getPreviousMessages () : Cette fonction est appelée à la fin de mainpage.jsp et récupère les messages de chat précédents pour cette salle de conversation.
- La fonction newMessage () est appelée par le code Java côté serveur lorsqu'un nouveau message est publié. La fonction fait également défiler automatiquement la zone de discussion pour afficher le dernier message.

## Chapitre 3 : CONCEPTION ET IMPLEMENTATION

- Les fonctions `sendMessageIfEnter ()` et `sendMessage ()` sont utilisées pour envoyer les messages des utilisateurs au serveur. Le champ de saisie pour le texte du message est dans le code HTML, la fonction `sendMessageIfEnter ()` écoute les événements `onkeyup` dans le champ de saisie. Si l'utilisateur appuie sur Entrée, la fonction `sendMessage ()` est appelée pour envoyer le message au serveur.
- Passons au développement du code Java, il y a plusieurs classes.
- La classe `Login` gère le login et la déconnexion de l'utilisateur et assure également le suivi des utilisateurs connectés.
- Pour la base des données des utilisateurs il y a la classe `UserDatabase`.
- La classe `Util` est utilisée par la classe `Login` et fournit des méthodes d'assistance pour l'exemple d'application.
- La logique de la fonctionnalité côté serveur se trouve dans la classe `ChatRoomDatabase`. Le contenu du chat est stocké dans un vecteur de chaînes. La méthode `getChatContent ()` renvoie simplement le vecteur `chatContent` au navigateur.

## 6. Bilan

### 6.1. Difficultés rencontrées

La difficulté principale, c'est surtout d'approfondir nos connaissances en Java et d'apprendre à programmer JavaEE en partant de rien et seulement en moins de trois mois. Les différentes difficultés du projet ne sont pas algorithmiques mais touchent beaucoup plus à la gestion des versions des outils utilisés tel que le serveur d'application et les différents fichiers `.jar`.

### Conclusion

Dans ce dernier chapitre nous avons essayé d'éclairer le chemin que nous avons suivi pour la conception et l'implémentation de notre plateforme, nous nous sommes basés surtout sur les grandes lignes comme les étapes du développement du modèle MVC, l'exploitation de la bibliothèque JSTL pour la manipulation de la base des données ainsi que la bibliothèque DWR pour la réalisation du service de Chat.

## Conclusion générale

A travers ce projet de fin d'étude, nous avons pu aborder une grande problématique telle que la prise en charge et l'amélioration de l'état de sante des personnes diabétiques ce qui nous a mené à découvrir le monde de télédiabète ainsi que les différents travaux et recherches dans ce domaine, puis en faisant une étude comparative nous avons finalement pu proposer notre solution, une plateforme collaborative de télédiabète qui prend en charge le suivi des patients diabétiques à distance. Ainsi, elle leur permet d'être à jour avec leurs données d'auto-surveillance glycémique, leur carnet alimentaire, et leur courbe de poids, et de communiquer facilement avec l'équipe médicale qui les suit.

Ce projet, malgré la courte durée dont nous avons disposé, nous a été d'un grand bénéfice. En effet, lorsque nous avons choisi comme outil de développement JavaEE, l'un des principaux objectifs était de proposer un profil extrêmement attractif sur nos CV puisque le marché de l'emploi dans le domaine de l'informatique est aujourd'hui dominé par le développement Java/JavaEE.

Sur un plan plus technique, nous avons découvert le monde du développement web. Néanmoins, la spécificité de JavaEE fait que nous avons du apprendre toutes les bases de cette nouvelle architecture ce qui nous a pris plus de temps, mais finalement, nous avons pu apprendre les bases de cette technologie. Compte tenu de la complexité de ces APIs, et du peu de temps dont nous disposions, nous n'avons pas pu pousser notre approche trop loin mais nous avons quand même eu le temps de juger de leur utilité.

En guise de perspectives, plusieurs travaux pourraient être envisagés pour poursuivre ce projet. Tout d'abord nous pourrions utiliser notre application comme logiciel de gestion de patients et cela en intégrant des algorithmes de calcul des doses d'insuline ainsi que des alertes et des notifications qui vont informer le patient et son soignant sur les ruptures de traitement, les potentiels besoins en insuline...etc. Une application de télédiabète doit répondre concrètement aux attentes et aux besoins des différents acteurs de la prise en charge de cette pathologie, l'idée est de co-construire notre plateforme avec les professionnels de santé, des spécialistes de la santé et des experts du suivi du diabète, et cela pour rendre la plateforme plus efficace. Pour clore, nous dirons que la seule et unique chose qui pourra mettre une limite aux perspectives c'est bien notre imagination.

يتطلب علاج مرض السكري من المرضى مراقبة مستوى جلوكوز الدم بشكل منتظم وزيارة الطبيب لتكييف علاجهم وحيث أن الاستشارات الطبية تتم فقط كل ستة أشهر هذا يعقد إلى حد كبير التحليل اللاحق للتغذية والنشاط البدني وجميع العوامل التي قد تؤثر على مستويات الجلوكوز في الدم لدى المرضى خلال هذه الفترة الطويلة. جعلت هذه الإشكالية من مرض السكري محور اهتمام للعديد من البحوث العالمية. لمعالجة هذه المشكلة، يقدم العمل الحالي منصة ويب تعاونية من شأنها تسهيل المراقبة عن بعد لمرضى السكري من خلال السماح لهم بتحديث بيانات المراقبة الذاتية لجلوكوز الدم، مذكرات تغذيتهم، ومنحنيات الوزن، والتواصل بسهولة مع الفريق الطبي الذي يتبع حالتهم من خلال خدمة مراسلة نصية آمنة. الأداة المستخدمة للتطوير هي منصة JavaEE، وهي تقنية تستخدم مجموعة من واجهات برمجة وتطبيقات جافا والتي تجعل من الممكن تطوير تطبيقات بإمكانها العمل ضمن متصفحات مختلفة و التميز بالتأمين كباقي التطبيقات جافا.

### كلمات مفتاحية

مرض السكري, مراسلة نصية آمنة, منصة واب , المراقبة عن بعد.

### Résumé

Le traitement du diabète oblige les patients à surveiller régulièrement leur glycémie et à se rendre chez leur médecin pour réajuster le traitement et comme les consultations n'ont lieu environ que tous les six mois, cela complique considérablement l'analyse à posteriori des prises alimentaires, de l'activité physique et de tous les paramètres qui ont pu jouer sur la glycémie des patients au cours d'une si longue période. Cette problématique a rendu la maladie de diabète un axe de recherche mondiale. Afin de répondre à cette problématique, le présent travail offre une plateforme web collaborative qui va faciliter le suivi à distance des patients diabétiques en leur permettant d'être à jour avec leurs données d'auto-surveillance glycémique, leur carnet alimentaire, et leur courbe de poids, et de communiquer facilement avec l'équipe médicale qui les suit grâce un service de messagerie sécurisée. L'outil utilisé pour le développement est la plateforme JavaEE, une technologie qui utilise un ensemble d'API java et qui permet de développer des applications pourront tourner sous de différents navigateurs et disposer de la sécurité que procure une application métier java.

### Mot clés

Diabète, Messagerie sécurisée, Plateforme web, JavaEE ; Suivie à distance

### Abstract

Diabetes treatment requires patients to regularly monitor their glycaemia levels and return to their doctor to readjust the treatment, as the consultations take place approximately only every six months, this greatly complicates the post analysis of food intake, physical activity and all the factors that may have played a role in the patients' glycaemia levels over such a long period. This problem has made diabetes as one of the global research focuses. In order to respond to this issue, the present work offers a collaborative web platform that will facilitate remote monitoring of diabetic patients, by allowing them to be up to date with their self-monitoring glycaemia data, their food diary, and their weight curve, and to communicate easily with the medical team that follows them through a secure messaging service. The tool used for the development is the platform JavaEE, a technology that uses a set of java APIs and allows to develop applications that are able to run under different browsers and have the security that a business java application provides.

### Key words

Diabetes , Secure messaging , JavaEE, Remote monitoring , Web platform