



République Tunisienne  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université de Tunis El Manar  
École Nationale d'Ingénieurs de Tunis



**Département Electrique**

## **Mini Projet POO-C++**

### **Gestion d'un système intelligent d'éclairage publique**

**Réalisé par :**

**Meriem Brahem Et Sarra Sabri**

**Classe : 2AGE2**

**Encadré par :**

**Mme.Amira Kallel**

**Année universitaire 2023/2024**

## **Remerciements**

Nous tenons à exprimer nos sincères remerciements à notre encadrante, Mme Amira Kallel, pour sa précieuse guidance et son expertise tout au long de ce projet. Sa passion pour son domaine d'expertise et son dévouement à notre réussite ont été une source d'inspiration et ont largement enrichi notre expérience.

## Résumé

Ce rapport vise à développer une application graphique pour gérer un système d'éclairage public intelligent . L'application permettra d'ajouter, de supprimer et de modifier des avenues et des lampadaires, ainsi que de suivre la consommation d'énergie de chaque lampe. Elle offrira également un contrôle d'allumage et d'extinction des lampadaires, ainsi qu'un suivi de la consommation d'énergie. L'objectif est de faciliter la gestion efficace de l'éclairage public, d'optimiser l'utilisation de l'énergie et d'améliorer le cadre de vie des habitants de la ville.

**Mots clés :** éclairage public, application graphique, gestion efficace, système intelligent, ville, lampadaires, consommation d'énergie.

# Table des matières

<b>Table des figures</b>	<b>6</b>
<b>1 Introduction générale</b>	<b>7</b>
<b>2 ETUDE CONCEPTUELLE ET THEORIQUE</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 PROBLEMATIQUE . . . . .	8
2.3 Outils . . . . .	8
2.3.1 Qt Creator . . . . .	8
2.3.2 SQLite . . . . .	9
2.3.3 DB Browser . . . . .	9
2.4 Conception . . . . .	9
2.4.1 Création de tables avec DB browser . . . . .	9
2.4.1.1 Table : "Login" . . . . .	9
2.4.1.2 Table : "AV" . . . . .	10
2.4.1.3 Table : "Lampadaires" . . . . .	10
2.4.1.4 Table : "La-Consommation" . . . . .	10
2.4.2 Schéma relationnel des tables . . . . .	11
2.5 Conclusion . . . . .	12
<b>3 Réalisation du projet</b>	<b>13</b>
3.1 Introduction . . . . .	13
3.2 Interface Login . . . . .	13
3.2.1 La Validation des informations . . . . .	14
3.2.2 Échec de Login . . . . .	15
3.2.3 Connexion à la base de données . . . . .	15
3.2.4 Logo de l'application . . . . .	15
3.3 Interface Admin . . . . .	15
3.4 Interface About . . . . .	16
3.5 Interface Lampadaire . . . . .	17
3.6 Interface Avenue . . . . .	19
3.7 Interface ON/OFF Lampadaire . . . . .	20
3.8 Interface ON/OFF Avenue . . . . .	21
3.9 L'interface de La Consommation Energétique . . . . .	22
3.10 Interface User . . . . .	22
3.11 Interface des Statistiques des Lampadaires . . . . .	24
3.12 Interface Histogramme de la consommation des lampes . . . . .	25

3.13	Interface courbe de la consommation des lampes . . . . .	26
3.14	Interface Statistique des Avenues . . . . .	27
<b>4</b>	<b>Conclusion générale</b>	<b>28</b>
	<b>Bibliographie</b>	<b>28</b>

# Table des figures

2.1	Table Login . . . . .	10
2.2	Table des Avenues . . . . .	10
2.3	Description de table des Lampadaires . . . . .	10
2.4	Description de table de Consommation . . . . .	11
2.5	Description des tables . . . . .	11
3.1	Interface Login . . . . .	14
3.2	La reconnaissance de l'admin (de meme pour le USER) . . . . .	14
3.3	Validation des données (de meme pour le USER) . . . . .	14
3.4	L'echec de Login . . . . .	15
3.5	Logo de l'application . . . . .	15
3.6	Interface de gestion de tous les fonctionnalités par l'Administrateur . . . . .	16
3.7	Interface "About" . . . . .	17
3.8	Interface de gestion des Lampadaires . . . . .	17
3.9	Code de développement de classe "Lampadaire" . . . . .	18
3.10	Interface de gestion des avenues où les lampadaires intelligents sont installés . . . . .	19
3.11	Interface ON/OFF Lampadaire . . . . .	20
3.12	Interface de gestion d'allumage de tout un avenue . . . . .	21
3.13	Interface de visualisation de la consommation de chaque lampe . . . . .	22
3.14	Interface de l'Utilisateur . . . . .	23
3.15	Interface des Statistiques des Lampadaires . . . . .	24
3.16	Interface Histogramme de la consommation des lampes . . . . .	26
3.17	Interface graphe de la consommation des lampes . . . . .	26

# Chapitre 1

## Introduction générale

Dans l'ère actuelle de la transformation numérique et de la nécessité croissante de préserver notre planète, le développement d'applications intelligentes de gestion des consommations devient essentiel. Parmi ces innovations, la gestion des systèmes d'éclairage public constitue un domaine d'intervention primordial. Ces systèmes intelligents ne se limitent plus à la simple illumination des rues, mais offrent également des avantages significatifs en termes de réduction de la consommation d'énergie, d'optimisation de la gestion des ressources, et d'amélioration de la qualité de vie des citoyens.

Dans ce contexte, nous avons entrepris le développement d'une application graphique novatrice destinée à gérer efficacement un système d'éclairage public au sein d'une ville. L'objectif principal de ce projet est de créer une plateforme intuitive et puissante, permettant aux administrateurs de superviser l'ensemble du réseau d'éclairage, de manière simple et efficace.

Cette application offrira une gamme de fonctionnalités étendue, allant de la gestion des avenues et des lampadaires à la surveillance de la consommation énergétique de chaque lampe. Grâce à une interface conviviale, les utilisateurs pourront aisément contrôler l'allumage et l'extinction des lampadaires, tout en bénéficiant d'un suivi précis de la consommation énergétique.

En résumé, notre projet vise à fusionner innovation technologique et développement durable pour offrir une solution de gestion d'éclairage public à la fois performante et respectueuse de l'environnement.

## Chapitre 2

# ETUDE CONCEPTUELLE ET THEORIQUE

### 2.1 Introduction

Dans cette section du rapport, nous allons détailler les phases de conception et de développement de notre système d'éclairage public intelligent. Nous aborderons également les décisions technologiques prises afin d'atteindre les objectifs fixés pour notre projet. Enfin, nous examinerons les résultats obtenus et identifierons les pistes d'amélioration envisageables pour le système.

### 2.2 PROBLEMATIQUE

Nous avons pour objectif de concevoir et développer une application graphique sous Qt Creator pour gérer efficacement le système d'éclairage public d'une ville. Cette solution devra permettre la gestion des avenues et des lampadaires, ainsi que la surveillance de la consommation énergétique de chaque lampe. L'application offrira des fonctionnalités permettant d'ajouter, de supprimer et de modifier les avenues et les lampadaires, ainsi que de contrôler l'état d'allumage des lampes via des boutons dédiés.

### 2.3 Outils

#### 2.3.1 Qt Creator

Qt Creator, conçu par l'équipe de Qt chez Nokia, est un environnement de développement multiplateforme dédié aux utilisateurs du framework Qt. Les fonctionnalités intégrées comprennent :

1. Éditeur avancé de code C++
2. Designer pour créer des interfaces graphiques
3. Gestionnaire de projet et de compilation
4. Aide intelligente intégrée
5. Debugger visuel



6. Outil de navigation rapide à travers le code
7. Support multiplateforme (Windows, Mac, Linux)

Offrant ainsi un environnement complet pour le développement d'applications multiplateformes sur Windows, Mac et Linux.

### **2.3.2 SQLite**

Nous avons opté pour SQLite comme base de données, une bibliothèque logicielle de gestion de base de données relationnelle. Contrairement aux bases de données traditionnelles, SQLite stocke les données dans un seul fichier portable, en faisant un choix adapté pour les applications mobiles, les navigateurs web et les systèmes embarqués.

Sa légèreté, sa rapidité et sa faible empreinte mémoire en font une solution efficace pour les applications avec des ressources limitées. SQLite prend en charge le langage SQL standard ainsi que de nombreuses fonctionnalités avancées, telles que les transactions ACID et les index, offrant une gestion de données efficace.

### **2.3.3 DB Browser**

DB Browser est un outil open source de gestion de bases de données relationnelles, spécialement conçu pour les bases de données SQLite. Son interface utilisateur graphique offre diverses fonctionnalités telles que la création et la modification de tables, l'insertion et la modification de données, la création de requêtes, ainsi que la gestion des schémas de base de données.

De plus, il permet la sauvegarde et la récupération de bases de données, garantissant ainsi la gestion et la sécurité des données. Avec son interface intuitive, DB Browser est une option solide pour les utilisateurs de tous niveaux de compétence en gestion de bases de données.

## **2.4 Conception**

### **2.4.1 Création de tables avec DB browser**

Les données stockées dans la base de données doivent respecter certaines règles de gestion définies pour le projet.

#### **2.4.1.1 Table : "Login"**

Cette table présentant les informations des utilisateurs est définie par :

1. idlogin :clé primaire ,auto incrémenté
2. User-name
3. PSW
4. Role

Nom	Type	NN	CP	IA	U
idlogin	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
User_name	varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PSW	varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Role	varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

FIGURE 2.1 – Table Login

#### 2.4.1.2 Table : "AV"

Cette table présentant la liste des avenues à contrôler dans la ville est définie par :

1. ID-AV : clé primaire ,auto incrémenté
2. Nom-AV

Nom	Type	NN	CP	IA	U
ID_Av	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Nom_Av	varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

FIGURE 2.2 – Table des Avenues

#### 2.4.1.3 Table : "Lampadaires"

Cette table contenant la liste des lampadaires dans les différentes avenues est définie par :

1. id-Lampe : clé primaire ,auto incrémenté
2. id-avenue : clé étrangère
3. energie consommation

```
CREATE TABLE "Lampadaires" (
  "id_lampe" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
  "id_avenue" INTEGER,
  "energie_consommation" INTEGER,
  FOREIGN KEY("id_avenue") REFERENCES "AV" ("ID_Av")
);
```

FIGURE 2.3 – Description de table des Lampadaires

#### 2.4.1.4 Table : "La-Consommation"

Cette table contenant les informations liées à la consommation d'énergie pour toutes les lampes est définie par :

1. id : clé primaire ,auto incrémenté
2. id-lampe : clé étrangère
3. consom-lampe

4. date-hr-debut
5. dte-hr-fin
6. consom-lamp

```
CREATE TABLE "La_Conso_mation" (
  "id"      INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
  "id_lampe" INTEGER,
  "date_hr_debut" TEXT,
  "dte_hr_fin" TEXT,
  "consom_lamp" REAL,
  FOREIGN KEY("id_lampe") REFERENCES "Lampadaires"("id_lampe")
);
```

FIGURE 2.4 – Description de table de Consommation

### 2.4.2 Schéma relationnel des tables

Les données sont organisées sous forme de tables, qui représentent la structure logique ou le niveau conceptuel du modèle relationnel.

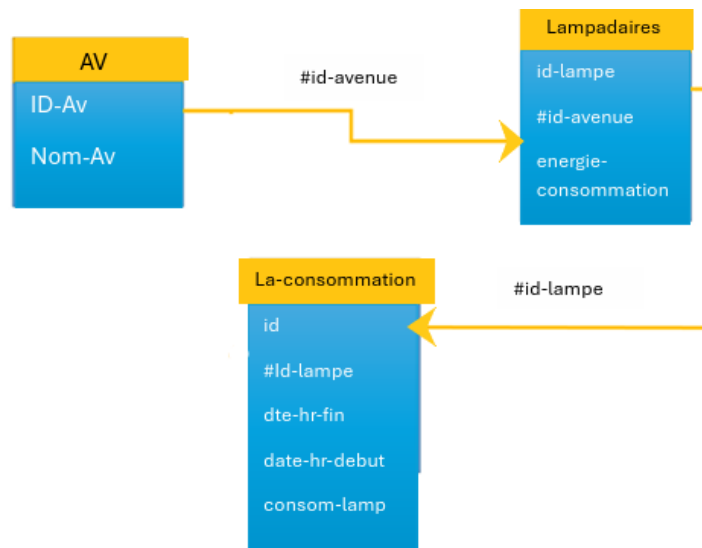


FIGURE 2.5 – Description des tables

L'objectif du modèle relationnel est de fournir des schémas de données conviviaux, d'optimiser l'accès à la base de données, et d'améliorer l'intégrité et la confidentialité des données.

Pour des raisons de sécurité on ajoutera une table Login pour que seuls les administrateurs ayant le droit de se connecter puissent accéder à l'interface et faire les changements .

En se basant sur les règles établies dans l'étape précédente, nous pouvons définir le modèle conceptuel de données.

## **2.5 Conclusion**

Ce chapitre introductif présente l'entité hôte, le contexte du projet, ainsi que les outils logiciels employés pour la mise en œuvre du projet de Gestion des avenues . Le chapitre suivant sera consacré à la conception et à la mise en place des interfaces graphiques.

# Chapitre 3

## Réalisation du projet

### 3.1 Introduction

Ce chapitre offrira une analyse approfondie de chaque interface, en décrivant ses fonctionnalités, sa conception ainsi que les étapes clés de sa réalisation.

### 3.2 Interface Login

L'interface de connexion que nous avons élaborée offre une approche simple et intuitive pour accéder à notre système. Composée de deux champs de texte permettant à l'utilisateur de saisir son nom d'utilisateur et son mot de passe, elle garantit une expérience fluide et conviviale. De plus, deux boutons radio clairement définis offrent la possibilité de choisir entre une connexion en tant qu'administrateur ou en tant qu'utilisateur standard.

Une fois que l'utilisateur a saisi ses informations et a appuyé sur le bouton de connexion, notre programme procède à une vérification auprès de la base de données SQLite, grâce à la puissance de Qt Creator C++. Si les informations d'identification sont valides, l'utilisateur est dirigé vers une nouvelle fenêtre correspondant à son profil, qu'il soit utilisateur ou administrateur. En cas d'informations incorrectes, un message d'erreur s'affiche, invitant l'utilisateur à corriger ses données.

Notre interface de connexion vise à être à la fois simple et sécurisée. Les champs de mot de passe sont masqués pour protéger les informations sensibles de l'utilisateur, tandis que l'utilisation de Qt Creator C++ garantit une base solide pour notre système. Cette approche nous permet de fournir une interface élégante et fonctionnelle, offrant ainsi une expérience utilisateur optimale à tous nos utilisateurs, quels que soient leurs besoins et leurs privilèges au sein de notre système.

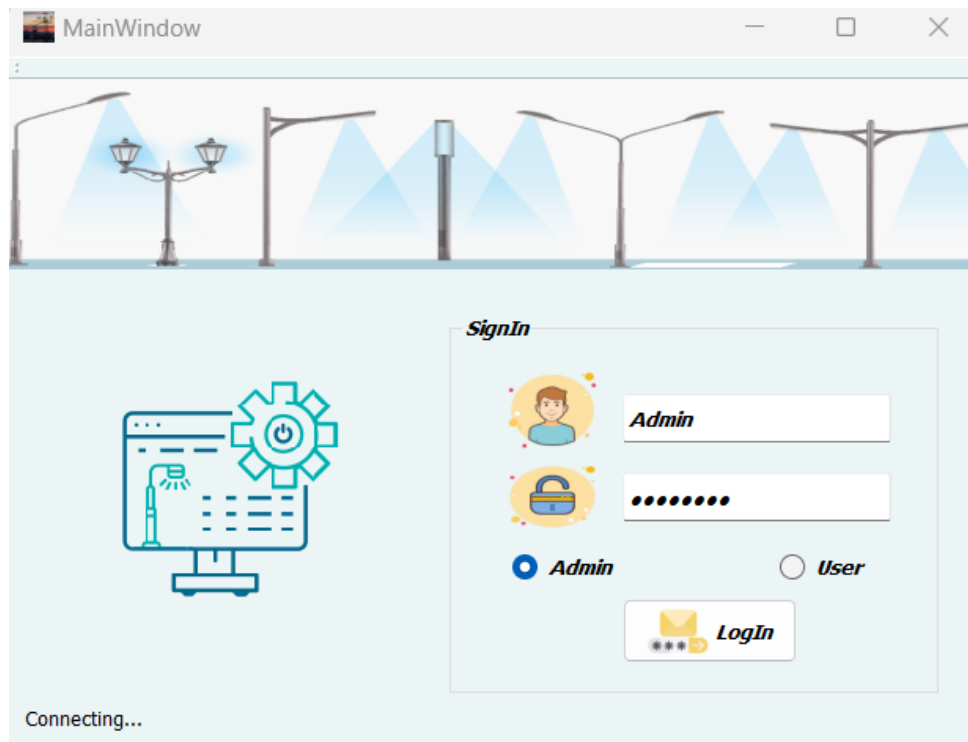


FIGURE 3.1 – Interface Login

### 3.2.1 La Validation des informations

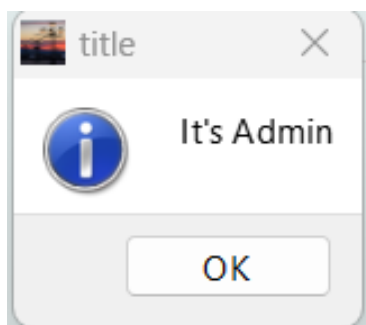


FIGURE 3.2 – La reconnaissance de l'admin (de meme pour le USER)

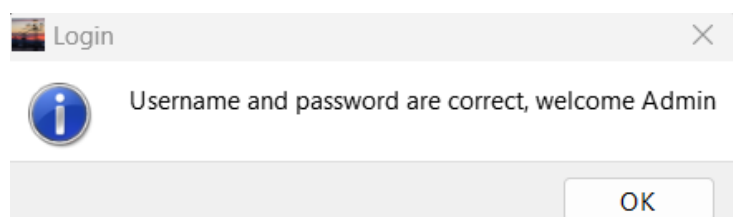


FIGURE 3.3 – Validation des données (de meme pour le USER)

### 3.2.2 Échec de Login

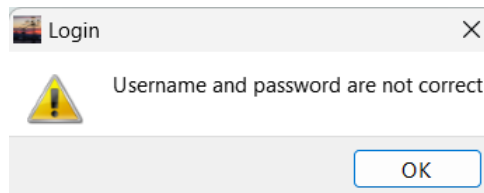


FIGURE 3.4 – L’échec de Login

### 3.2.3 Connexion à la base de données

Le nom de la classe que nous avons créée pour gérer la connexion à la base de données SQLite est "DBConnexion". Nous avons implémenté cette classe pour encapsuler les détails de la connexion à la base de données et fournir une interface simple pour interagir avec la base de données. La classe "DBConnexion" contient les fonctions nécessaires pour se connecter à la base de données SQLite et une fonction pour fermer la connexion. En utilisant cette classe, nous avons pu rendre notre programme de connexion de base de données plus modulaire et facile à maintenir.

En résumé, la classe "DBConnexion" est un composant important de notre programme de connexion de base de données, qui nous permet de gérer efficacement les connexions à la base de données SQLite en utilisant Qt Creator C++.

### 3.2.4 Logo de l’application

Lors de lancement de l’application le logo apparaît.

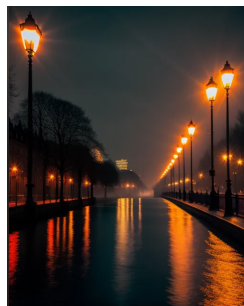


FIGURE 3.5 – Logo de l’application

## 3.3 Interface Admin

L’interface d’administration : "AdminDialog" que nous avons développée se compose de cinq boutons distincts : "Avenue", "Lampadaire", "Energy Consumption", "ON/OFF Lampadaire" et "ON/OFF Avenue".

Chacun de ces boutons offre des fonctionnalités spécifiques telles que l'ajout d'une nouvelle avenue et de nouveaux lampadaires, le suivi de la consommation d'énergie, ainsi que le contrôle de l'allumage et de l'extinction des lampes.

Cette fenêtre permet à l'administrateur de modifier les informations stockées dans la base de données et la gestion des différentes fonctionnalités de l'application.

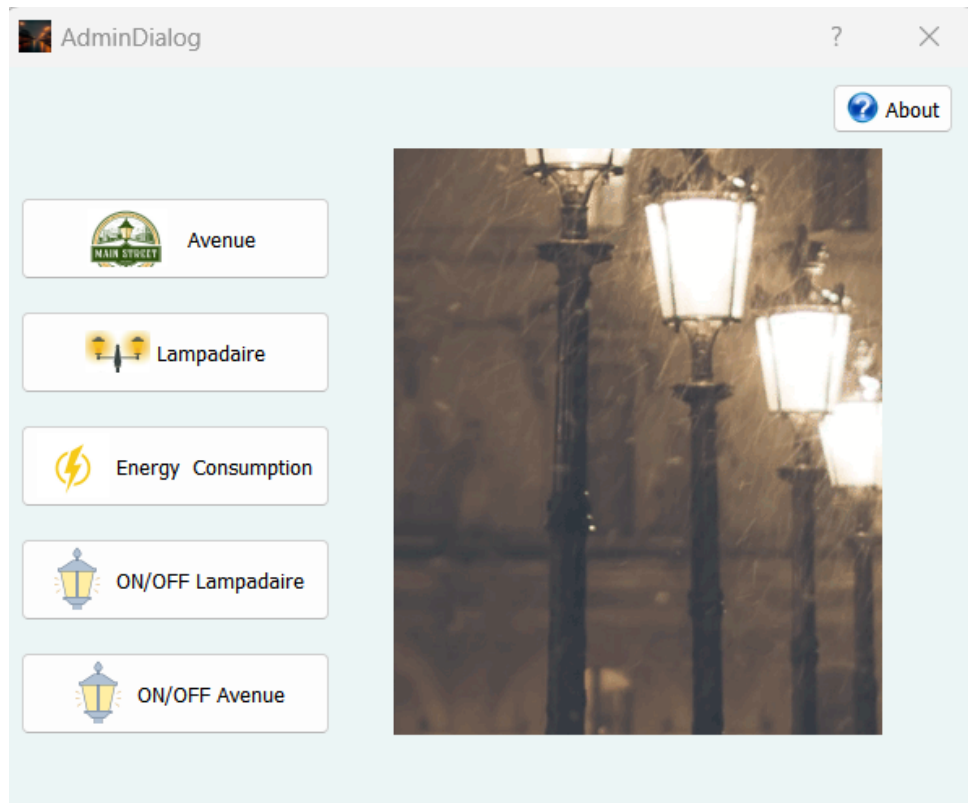


FIGURE 3.6 – Interface de gestion de tous les fonctionnalités par l'Administrateur

La fenêtre "About" contient généralement des informations sur l'application, comme son nom, son numéro de version, sa description, et d'autres informations importantes.

Pour améliorer l'interaction avec l'utilisateur de notre application, nous avons opter pour des icones animés de type GIF. Nous avons utilisé la classe QMovie de Qt Creator pour charger les fichiers GIF et les afficher dans les widgets correspondants.

### 3.4 Interface About

La page "About" offre un contexte supplémentaire sur le développement de l'application et son objectif.



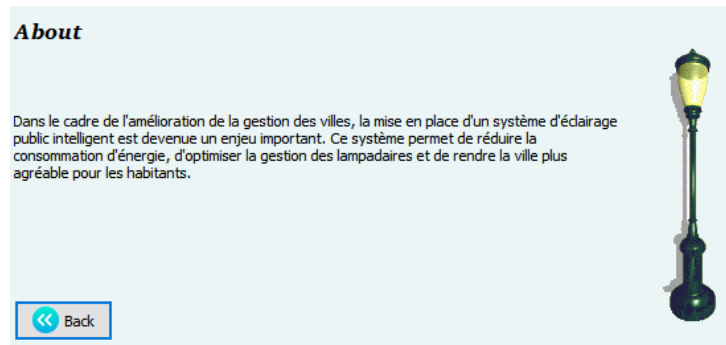


FIGURE 3.7 – Interface "About"

### 3.5 Interface Lampadaire

L'interface "lampadairedialog" regorge de fonctionnalités, telles qu'un bouton "Load" permettant d'importer les informations de la base de données : "Lampadaires" dans un QTableView, ainsi qu'un bouton "Initialiser" pour pré-remplir les champs LineEdit avec les détails de lampadaire sélectionnée dans le QTableView.

Les champs LineEdit offrent à l'utilisateur la possibilité de saisir les détails nécessaires pour l'ajout d'un nouveau lampadaire, tels que l'identifiant de l'avenue, le nom de l'avenue et la puissance de lampadaire. Une fois les informations saisies, l'utilisateur peut les enregistrer dans la base de données à l'aide du bouton "Save" activé suite au clic sur le bouton "Init".

De plus, notre interface propose la possibilité de mettre à jour les informations d'un lampadaire existant dans la base de données grâce au bouton "Update". Le bouton "Delete" permet également à l'utilisateur de retirer un lampadaire en se basant sur son identifiant, automatiquement chargé à partir du QTableView lorsqu'il est sélectionné.

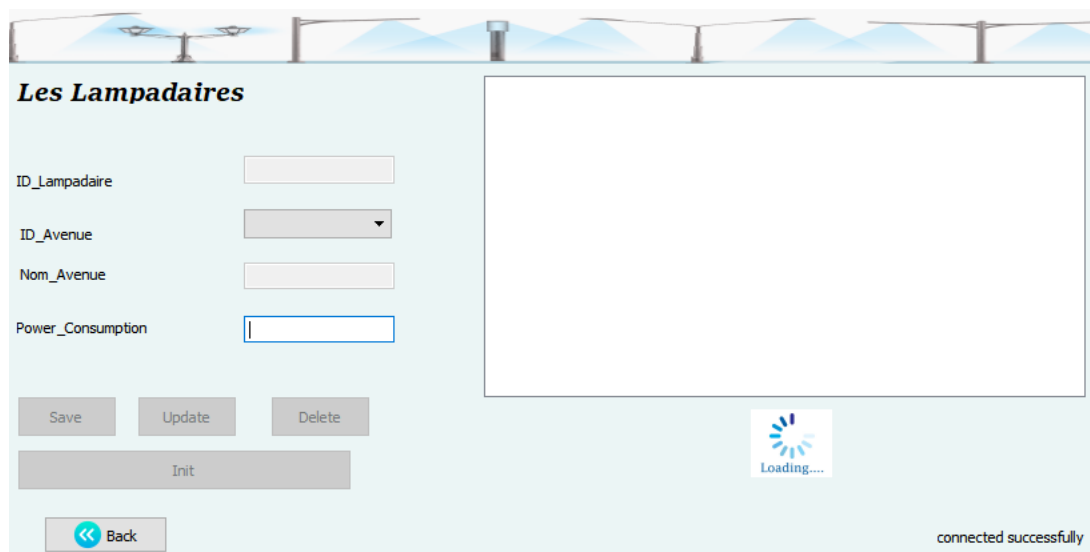


FIGURE 3.8 – Interface de gestion des Lampadaires

Pour faciliter la programmation , nous avons créé des classes comme la classe "Lampadaire" qui contient les getters et setters pour ses attributs , ce qui facilite l'accès et la manipulation de ces informations tout au long de notre programme.

Pour gérer l'interaction de l'utilisateur et la base de données, on crée à chaque fois à une instance de la classe "DBConnexion".

```
class Lampadaire
{private:
    QSqlDatabase mydb ;
    int id_lampe,id_avenue;
    float energie_consommation;
public:
    Lampadaire();
    ~ Lampadaire();
    void set_id_lampe(int l);
    void set_id_avenue(int a);
    void set_consommation(int c);
    int get_id_lampe();
    int get_id_avenue();
    int get_consommation();
    void Ajouter();
    void Supprimer();
    void Modifier();
    QSqlQueryModel * Load();
    QSqlQueryModel * Load2();
};
```

FIGURE 3.9 – Code de développement de classe "Lampadaire"

### 3.6 Interface Avenue

L'interface "Les Avenue" de notre application offre une fonctionnalité étendue permettant à l'administrateur de gérer les informations concernant les rues où les lampadaires intelligents sont installés. Centralisant ces données, elle présente un tableau widget affichant les informations provenant de la table "Avenue" de notre base de données. Les attributs de cette table incluent l'identifiant de l'avenue ainsi que le nom de la rue correspondante.

Pour simplifier l'interaction avec ces données, l'interface intègre un bouton "Load" permettant à l'utilisateur de charger les informations depuis la table "Avenue" directement dans le tableau widget, offrant ainsi une visualisation claire et actualisée des données validées. De plus, un bouton "Clear" est prévu pour réinitialiser le champ du nom lorsqu'un identifiant est sélectionné dans le tableau widget, facilitant ainsi les actions de l'utilisateur.

Concernant la gestion des données, l'ajout d'une nouvelle avenue est simplifié grâce au bouton "Save", qui permet de sauvegarder les informations saisies dans la base de données, tout en attribuant automatiquement un identifiant unique à l'avenue ajoutée. De même, le bouton "Update" offre la possibilité de mettre à jour les informations d'une avenue existante dans la base de données, assurant ainsi la précision et l'actualité des données. Enfin, pour assurer une gestion complète, le bouton "Delete" permet à l'administrateur de supprimer une avenue spécifique à partir de son identifiant, garantissant ainsi la flexibilité et la robustesse de notre système de gestion des rues et des lampadaires intelligents.

The interface, titled "Les Avenues", features a header with an illustration of smart streetlights. Below the header, there is a form with two input fields: "ID Av" and "Nom Av". To the right of the form is a table displaying a list of avenues. The table has two columns: "ID\_Av" and "Nom\_Av". The first row is highlighted in blue. Below the form and table are several buttons: "Save", "Delete", "Update", "Clear", and a "Back" button with a double arrow icon. At the bottom left, it says "Connected successfully", and at the bottom right, there is a "Loading...." indicator with a circular progress icon.

	ID_Av	Nom_Av
1	2	avenue de Carthage
2	4	avenue de rose velte
3	35	avenue plutarque S...
4	37	avenue de hannibal
5	38	avenue de liberté
6	42	avenue de gustave f...
7	43	avenue de la républi...

FIGURE 3.10 – Interface de gestion des avenues où les lampadaires intelligents sont installés

### 3.7 Interface ON/OFF Lampadaire

Dans l'interface "controldialog" de l'application, l'utilisateur peut éteindre ou allumer un nouvel lampadaire en spécifiant leur id dans le Combox, ainsi la puissance de la lampe est affiché. En ce qui concerne l'état de lampadaire, il est par défaut désactivé, activé suite au clic sur le bouton "Action".

Le bouton "Contol" est essentiel pour vérifier la liste des lampadaires déjà allumés, permettant ainsi de cliquer sur l'indice de lampe à éteindre.

Suite au contrôle des lampadaires, les attributs sont chargés instantanément dans la table "La-Consommation" incluent l'identifiant du lampadaire, la date d'allumage, la date d'éteint, et la consommation d'énergie cumulée.

Cette interface offre aux utilisateurs la possibilité de surveiller aisément des lampadaires.

Pour améliorer l'interaction avec l'utilisateur de notre application, nous avons remplacé les boutons statiques par des icônes.

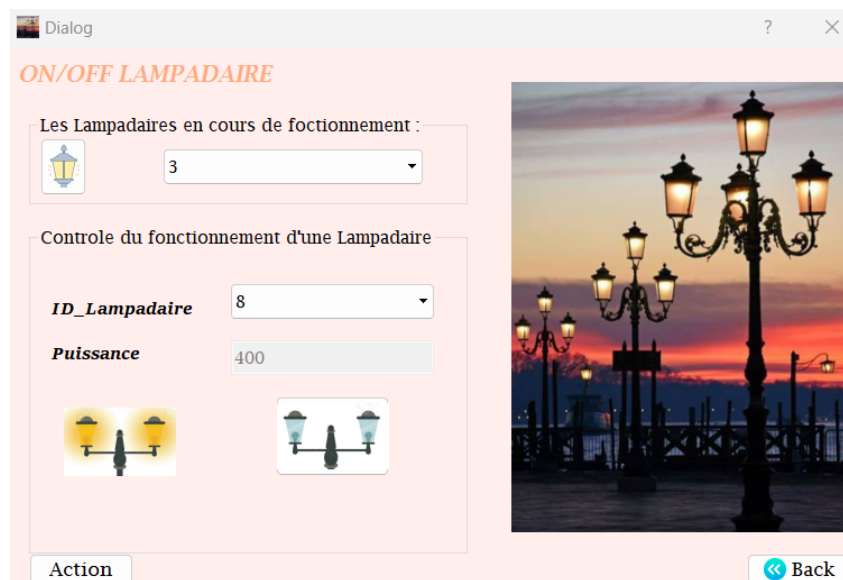


FIGURE 3.11 – Interface ON/OFF Lampadaire

### 3.8 Interface ON/OFF Avenue

Pour garantir l'éclairage optimal d'une avenue, notre application offre une page dédiée au contrôle des lampadaires. Cette interface permet à l'administrateur de gérer l'allumage et l'extinction de toutes les lampes d'une avenue en utilisant un bouton "Activer" et en sélectionnant l'avenue correspondante via un menu déroulant.

Lorsque l'administrateur appuie sur le bouton "Allumer", symbolisé par une icône représentant une avenue éclairée, toutes les données sont enregistrées dans la table de consommation de la base de données. La consommation d'énergie est générée aléatoirement entre 50 et 2000 afin de refléter au mieux une application réelle. L'heure de début est enregistrée dans la table de consommation au moment où le bouton "Allumer" est activé, tandis que l'heure de fin est initialisée par défaut à "00 :00 :00". Cette dernière est mise à jour automatiquement lorsque l'administrateur appuie sur le bouton "Éteindre", reflétant ainsi l'instant où les lampes sont éteintes.

Le calcul de la consommation d'énergie est réalisé en multipliant la durée d'allumage par la puissance de chaque lampe. Cette puissance est récupérée depuis la table "Lampadaires" de la base de données. Les valeurs de consommation ainsi calculées sont ensuite ajoutées à la table "Consommation", assurant ainsi un suivi précis de l'utilisation énergétique des lampadaires de l'avenue.

Cette page de contrôle d'allumage d'avenue offre à l'administrateur un outil puissant et précis pour gérer l'éclairage de manière efficace tout en assurant un suivi minutieux de la consommation d'énergie.

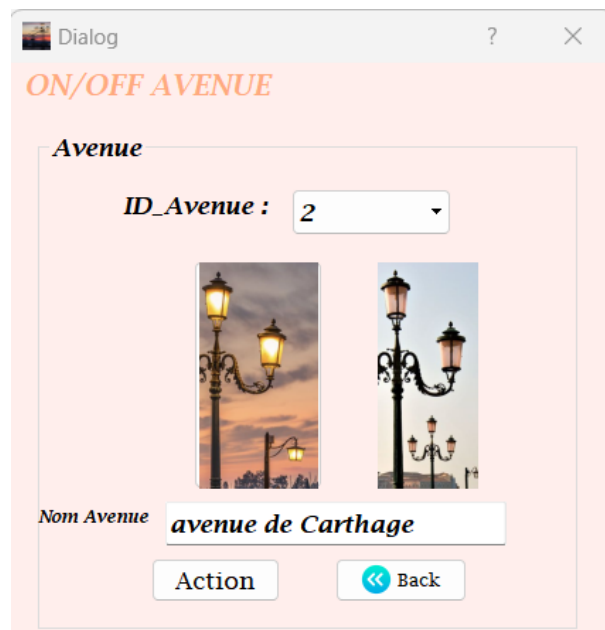


FIGURE 3.12 – Interface de gestion d'allumage de tout un avenue

### 3.9 L'interface de La Consommation Énergétique

En complément des interfaces précédemment présentées, l'interface de "Consommation Énergétique" offre à l'administrateur une vue claire et accessible de la consommation d'énergie des lampadaires. Cette interface est conçue pour simplifier la surveillance de la consommation énergétique, fournissant ainsi une vision globale de l'utilisation des ressources lumineuses.

Au cœur de cette interface se trouve le bouton "Load", permettant de charger les données directement depuis la base de données, plus précisément depuis la table "Consommation". Cette table est dynamiquement mise à jour par les actions effectuées via les interfaces de contrôle d'allumage d'avenue et de lampadaire, qui permettent à l'administrateur de gérer l'allumage et l'extinction des lampadaires, que ce soit individuellement ou en groupe au sein d'une avenue.

En offrant cette visibilité sur la consommation énergétique, notre interface permet aux utilisateurs de surveiller facilement l'impact de l'éclairage sur la consommation d'énergie. Cette transparence favorise la prise de décisions éclairées en matière de gestion de l'énergie, permettant ainsi d'optimiser l'utilisation des ressources disponibles et de contribuer à la création d'un environnement plus durable.

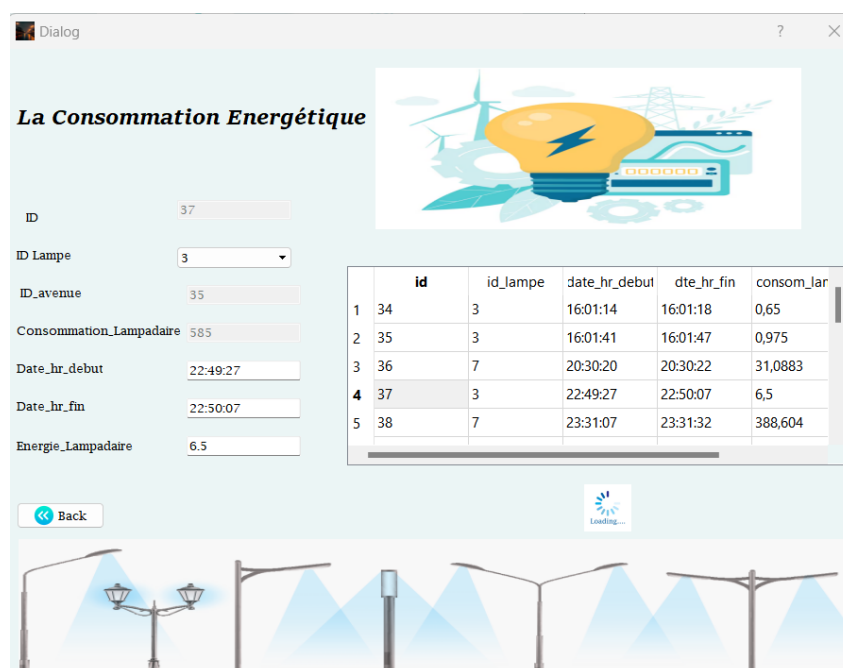


FIGURE 3.13 – Interface de visualisation de la consommation de chaque lampe

### 3.10 Interface User

Cette interface propose une fonctionnalité essentielle : la visualisation des statistiques de consommation. Deux boutons distincts sont disponibles , l'un pour afficher les statis-

tiques de consommation des lampadaires et l'autre pour celles des avenues. Ces boutons permettent d'accéder à des graphiques clairs et précis, offrant une représentation visuelle de l'énergie consommée.

Cette fonctionnalité revêt une importance particulière, car elle permet une analyse approfondie des données et une meilleure compréhension des schémas de consommation. Les utilisateurs peuvent ainsi surveiller facilement la consommation d'énergie des lampadaires, ce qui leur permet de prendre des décisions éclairées en matière de gestion de l'énergie.

En offrant cette capacité d'analyse avancée, notre interface contribue à optimiser l'utilisation des ressources et à favoriser un environnement plus durable. Elle permet aux utilisateurs de mieux comprendre leur consommation énergétique, ce qui les incite à adopter des pratiques plus responsables et à contribuer à la préservation de notre planète.

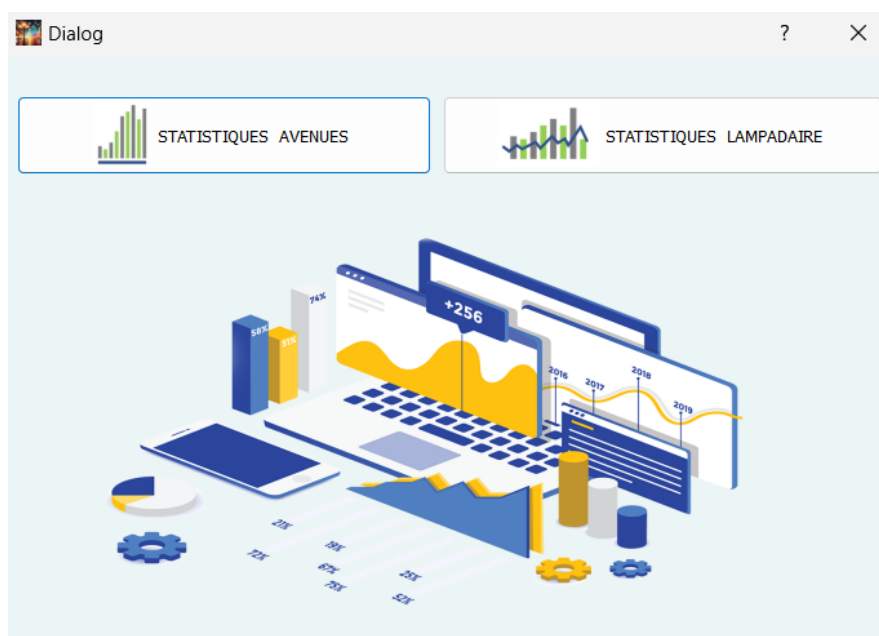


FIGURE 3.14 – Interface de l'Utilisateur

### 3.11 Interface des Statistiques des Lampadaires

Dans cette interface, l'utilisateur standard a la possibilité de choisir la statistique qu'il souhaite visualiser en cliquant sur l'un des deux boutons disponibles : soit la courbe d'évolution de la consommation électrique, soit l'histogramme de la consommation de chaque lampe.



FIGURE 3.15 – Interface des Statistiques des Lampadaires



## 3.12 Interface Histogramme de la consommation des lampes

L'interface permet de visualiser l'histogramme de l'énergie consommée en fonction de l'identifiant de chaque lampadaire .

Cette fonctionnalité offre une analyse détaillée des données, permettant ainsi une meilleure compréhension des schémas de consommation.

Le code fourni est une implémentation utilisant la bibliothèque QCustomPlot pour générer l'histogramme à partir de la table "La-Consommation" dans la base de données . Voici une description de ce travail :

1. Initialisation de l'interface utilisateur (UI) : La classe StatisticDialog est une boîte de dialogue Qt qui affiche un histogramme de la consommation d'énergie des lampadaires.
2. Connexion à la base de données : Le constructeur de StatisticDialog tente d'ouvrir une connexion à la base de données SQLite en utilisant la classe DBConnexion. Si la connexion est établie avec succès, une requête SQL est exécutée pour extraire les données de consommation d'énergie des lampes à partir de la table "La-Consommation".
3. Préparation des données : Les données sont agrégées pour obtenir la consommation. Ces données sont ensuite utilisées pour créer un histogramme.
4. Création de l'histogramme : Un objet QCPBars est utilisé pour représenter les données sous forme d'histogramme dans le widget "customPlot".
5. Configuration de l'histogramme : L'axe x pour les identifiant des lampadaires ainsi l'axe y pour les valeurs d'énergie sont configurés pour afficher les informations pertinentes . Les barres de l'histogramme sont dimensionnées et positionnées en fonction des données extraites de la base de données.

En permettant aux utilisateurs de surveiller facilement la consommation d'énergie des lampadaires, cette interface facilite la prise de décisions en matière de gestion énergétique. De cette manière, elle favorise une utilisation optimale des ressources, contribuant ainsi à un environnement plus durable.

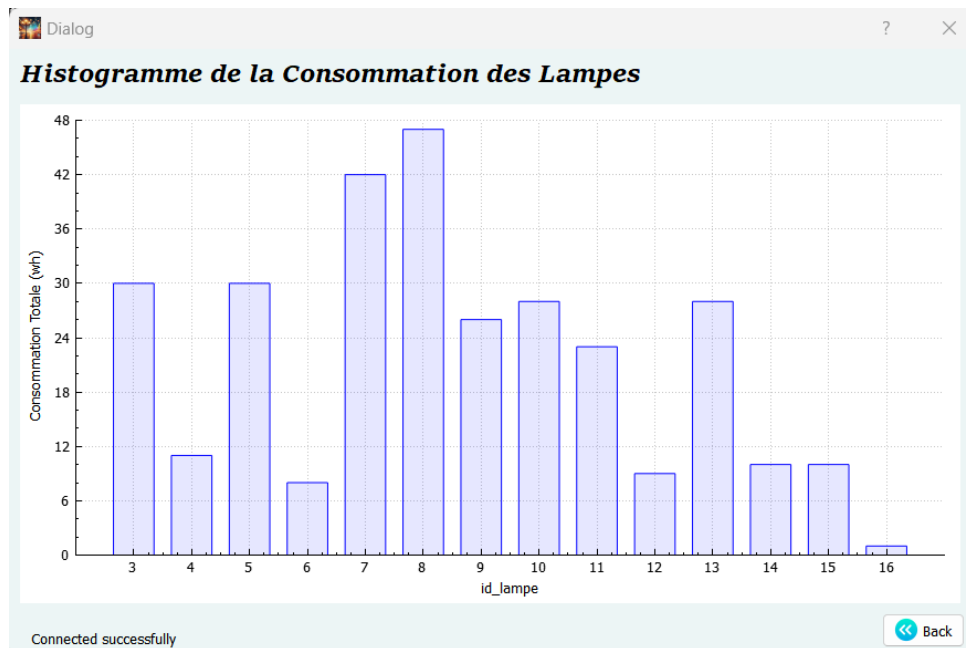


FIGURE 3.16 – Interface Histogramme de la consommation des lampes

### 3.13 Interface courbe de la consommation des lampes

L'interface permet de visualiser le graphe de l'énergie consommée en fonction de l'identifiant de chaque lampadaire .

Le code fourni est une implémentation utilisant la bibliothèque QCustomPlot pour générer le graphe à partir de la table "La-Consommation" dans la base de données. L'axe x pour les identifiant des lampadaires ainsi l'axe y pour les valeurs d'énergie .

Cette caractéristique permet une analyse approfondie des données, facilitant ainsi la compréhension des modèles de consommation.

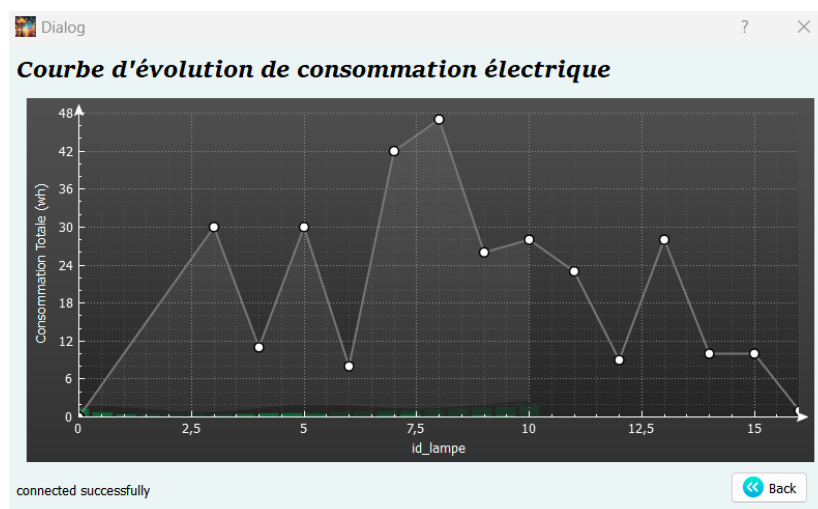


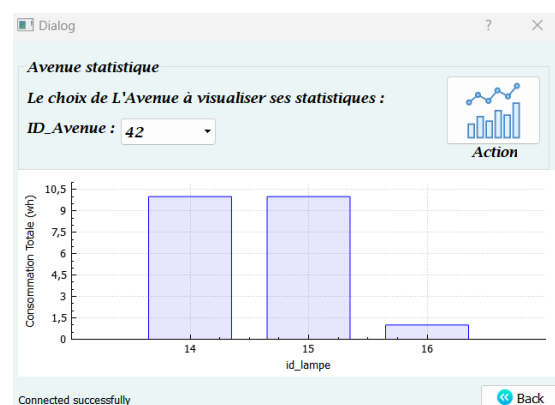
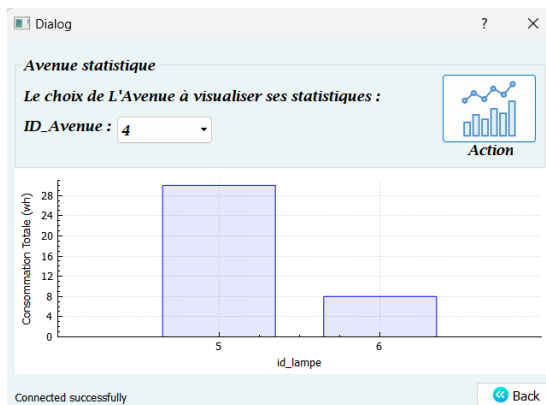
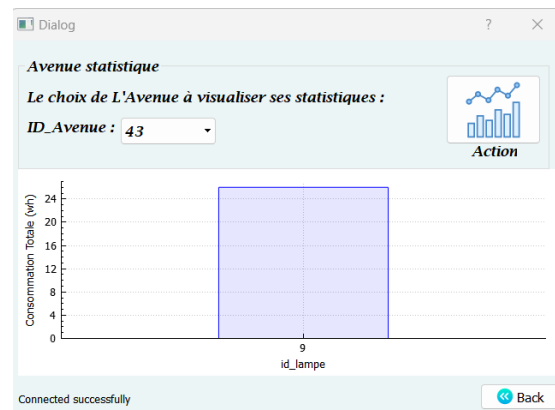
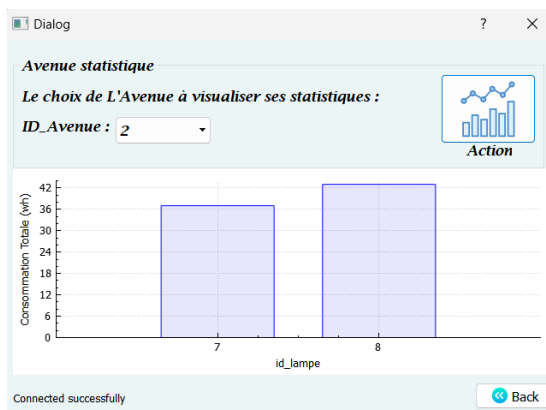
FIGURE 3.17 – Interface graphe de la consommation des lampes

### 3.14 Interface Statistique des Avenues

L'application de statistiques d'avenue permet aux utilisateurs de visualiser la consommation d'énergie des lampes dans d'une avenue spécifique. En sélectionnant une avenue à partir d'une liste déroulante, l'application récupère les données de consommation de chaque lampe associée à cette avenue dans la base de données. Ces données sont ensuite affichées sous forme d'histogramme, où chaque barre représente la consommation totale d'une lampe.

L'interface utilisateur est réactive, permettant une mise à jour instantanée des données en fonction des sélections de l'utilisateur. Ce processus offre une vue claire et concise de la consommation d'énergie des lampes, aidant ainsi à la gestion efficace de l'éclairage public.

Voici Histogramme de la consommation des lampes par avenue :



# Chapitre 4

## Conclusion générale

En synthèse, ce rapport retrace notre parcours de développement de compétences à travers l'utilisation de Qt Creator, C++ et MySQL. Notre projet consistait à concevoir une application de gestion de l'éclairage public dans une ville intelligente .

Cette expérience nous a permis d'approfondir notre compréhension de la programmation orientée objet, de la création d'interfaces graphiques avec des bibliothèques telles que Qt, ainsi que de la gestion de données relationnelles via MySQL. Nous sommes désormais mieux outillés pour développer des applications professionnelles répondant aux besoins spécifiques des utilisateurs.

Cette expérience est enrichissante tant sur le plan personnel que professionnel, renforçant notre confiance dans notre capacité à concevoir des applications efficaces, fiables et sécurisées.