

ANNEXE

Table des matières

main.cpp	4
CEpaper.hpp.....	8
Ecran.hpp	9
IHM.hpp	11
Item.hpp.....	13
Variable.hpp.....	15
CEpaper.cpp	17
Ecran.cpp.....	18
IHM.cpp.....	21
Item.cpp	24
Variable.cpp	26
CEnumVehicule.hpp.....	29
CVehiculeDetecte.hpp	29
CVehiculeDetecte.cpp.....	32
CDetecteurPassage.hpp	35
CDetecteurPassage.cpp	37
main.cpp	38
PROCEDURE D'INSTALLATION :	41
launch.json	41
task.json	42
settings.json	42
c_cpp_properties.json	43
CModemSigfox.hpp.....	47
CModemSigfox.cpp	49
CTrameStatsSigfox.hpp	53
CTrameStatsSigfox.cpp	55
CStatsManager.hpp.....	59
CStatsManager.cpp.....	62
CStatsVehicule.hpp	66
CStatsVehicule.cpp.....	72
CStatsEssieu.hpp	80
CStatsEssieu.cpp	82

PrintfBufferedSerial.hpp	85
PrintfBufferedSerial.cpp.....	86
main.cpp	87
TESTA Tony.....	88
CVehicule.h	89
CVehicule.cpp.....	90
CPassage.h	93
CPassage.cpp.....	94
cscenario.h	96
CScenario.cpp.....	97
CStats.h	100
CStats.cpp.....	101
CVehiculeManager.h.....	103
cvehiculemanager.cpp	104
cpassagemanager.h	108
cpassagemanager.cpp.....	109
cscenarioManager.h	111
cscenarioManager.cpp.....	112
cdialoguevehicule.h	117
Cdialoguevehicule.cpp	118
cdialogoscenario.h	119
cdialogoscenario.cpp	120
dialogueconfigsvehicule.h.....	122
dialogueconfigsvehicule.cpp.....	123
dialogueconfigscenario.h	125
dialogueconfigscenario.cpp	126
mainwindow.h	131
Mainwindows.cpp.....	133
SigfoxMessageController.php	144
CollecteSigfoxController.php	148
Web.php.....	149
web.php	150
boitier.php	152
campagne.php	153
statistique.php	153
User.php.....	154
alertController.php	155

clientController.php	157
gestionboitierController.php	160
gestioncampagneController.php	162
gestioncompteController.php	165
HomeController.php	170
PagesController.php	173
Home.blade.php	174
master.blade.php	178
navbar.blade.php	179
footer.blade.php	179
login.blade.php	180
accueil.blade.php	181
alertbatterie.blade.php	186
liste.blade.php	187
consultationcampagne.blade.php	191
graphiques_camion.blade.php	194
listecampagneclient.blade.php	199
popup_highscore.blade.php	201
stat_false.blade.php	204
stat_true.blade.php	205
creationBoitier.blade.php	209
ListeBoiter.blade.php	210
ModifierBoitier.blade.php	213
créationCampagneMesure.blade.php	215
listeCampagneMesure.blade.php	218
modifierCampagneMesure.blade.php	221
inscription.blade.php	225
ListeClient.blade.php	228
ModifierClient.blade.php	231
inscriptionAdminTechnicien.blade.php	233
ListeAdminTechnicien.blade.php	235
ModifierAdminTechnicien.blade.php	238
MessageGoogle.blade.php	240
message-google.blade.php	241
Source E7 :	241

HERVIER XAVIER	main.cpp	1/4
<pre>/*** * @file main.cpp * @brief Mise en œuvre module d'affichage ePaper HAT de waveshare * @author Hervier Xavier * @version 1.0 * @date 09/11/2020 */ #define COLORED 1 #define UNCOLORED 0 enum class BOUTON {HOME, MOINS, PLUS, SELECT} ; #define SLEEP_TIME 1000 // (msec) #include "mbed.h" #include <stdio.h> #include <stdlib.h> #include "epd2in7/epd2in7.h" #include "epd2in7/imagedata.h" #include "epd2in7/epdpaint.h" #include "CEpaper.hpp" #include "IHM.hpp" #include "Ecran.hpp" #include "Item.hpp" //affectation des broches pour la communication spi PinName _Mosi = PE_14 ; //SPI4_MOSI; PinName _Miso = PE_13 ; // SPI4_MISO non connecté PinName _Sclk = PE_12 ; // SPI4_SCK PinName _Cs = PE_11 ; // SPI4_SC //affectation des broches de controle de l'afficheur PinName _Rst = PF_13 ; PinName _Dc = PE_9 ; PinName _Busy = PF_12 ; DigitalOut myledV(LED1); DigitalOut myledB(LED2); DigitalOut myledR(LED3); Timer debounce; #define BT_HOME PD_15 #define BT_PLUS PD_14 #define BT_MOINS PB_9 #define BT_SELECT PB_8 //configuration des entrées en pullup pour les boutons DigitalIn diHome(BT_HOME, PullUp) ; DigitalIn diPlus(BT_PLUS, PullUp) ;</pre>		

HERVIER XAVIER	main.cpp	2/4
<pre>DigitalIn diMoins(BT_MOINS, PullUp); DigitalIn diSelect(BT_SELECT, PullUp); // DigitalIn diHome(BT_HOME, PullDown); // DigitalIn diPlus(BT_PLUS, PullDown); // DigitalIn diMoins(BT_MOINS, PullDown); // DigitalIn diSelect(BT_SELECT, PullDown); //Configuration des interruptions pour la détection de l'appui des boutons InterruptIn interruptHome(BT_HOME); InterruptIn interruptPlus(BT_PLUS); InterruptIn interruptMoins(BT_MOINS); InterruptIn interruptSelect(BT_SELECT); IHM *IHMpt; void afficherMenu(BOUTON btn) { if (btn == BOUTON::PLUS){ IHMpt->btnPlus(); } if (btn == BOUTON::MOINS){ IHMpt->btnMoins(); } if (btn == BOUTON::SELECT){ IHMpt->btnValid(); } if (btn == BOUTON::HOME){ IHMpt->btnHome(); } myledB = 0 ; } bool fctDebounce(void){ if (debounce.read_ms()>200){ debounce.reset(); return true; } else return false; } void btHome(void) { if (fctDebounce()) myledB = 1 ; afficherMenu(BOUTON::HOME) ; } void btPlus(void) { if (fctDebounce()) myledB = 1 ; afficherMenu(BOUTON::PLUS) ; } void btMoins(void) { if (fctDebounce()) myledB = 1 ; afficherMenu(BOUTON::MOINS) ; }</pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	main.cpp	3/4
<pre> void btSelect(void) { if (fctDebounce()) myledB = 1 ; afficherMenu(BOUTON::SELECT) ; } int main() { myledB = 0 ; myledV = 0 ; myledR = 0 ; Epd * gEpd = new Epd(_Mosi, _Miso, _Sclk, _Cs, _Dc, _Rst, _Busy); CEpaper *Epaper = new CEpaper(gEpd); IHMpt = new IHM() ; //initialisation epaper if (gEpd->Init() != 0) { myledR = 1; return -1; } else myledV = 1; Epaper->rotate(); Epaper->effacer() ; // Création des différents Ecran Ecran * EcranAc = new Ecran(Epaper,"Accueil",true,true); Ecran * EcranPara = new Ecran(Epaper,"Parametre",false,false); Ecran * EcranSta = new Ecran(Epaper,"Stats",false,false); Ecran * EcranTes = new Ecran(Epaper,"Test",false,false); // Ajout des Ecran à l'IHM IHMpt->addEcran(EcranAc); // Ajout des Items aux différents Ecran EcranAc->addItem(new Item (Epaper,"Parametre",true)); EcranAc->addItem(new Item (Epaper,"Stats",false)); EcranAc->addItem(new Item (Epaper,"Test",false)); IHMpt->addEcran(EcranPara); EcranPara->addItem(new Item (Epaper,"Capteur",true, new Variable(2,1,0,2))); EcranPara->addItem(new Item (Epaper,"Vitesse",false, new Variable(20,10,20,130))); IHMpt->addEcran(EcranSta); EcranSta->addItem(new Item (Epaper,"Stats",true)); IHMpt->addEcran(EcranTes); EcranTes->addItem(new Item (Epaper,"Mode Test",true)); btHome() ; // Epaper->Ecrire(10, 10, "kolantr", COLORED); // Epaper->Ecrire(10, 20, "kolantr", COLORED); // Epaper->Ecrire(10, 30, "kolantr", COLORED); // Epaper->Ecrire(10, 40, "kolantr", COLORED); // Epaper->Ecrire(10, 50, "kolantr", COLORED); // Epaper->majAffichage(); </pre>		

HERVIER XAVIER	main.cpp	4/4
<pre>// crée le thread chargé de traiter la file d'événements pour envoi de relevés Thread eventThread; EventQueue eventQueue(32 * EVENTS_EVENT_SIZE); eventThread.start(callback(&eventQueue, &EventQueue::dispatch_forever)); debounce.start(); // Interruption pour les Boutons interruptHome.rise(eventQueue.event(callback(btHome))) ; interruptMoins.rise(eventQueue.event(callback(btPlus))) ; interruptPlus.rise(eventQueue.event(callback(btMoins))) ; interruptSelect.rise(eventQueue.event(callback(btSelect))) ; //interruptHome.rise(&btHome) ; //interruptPlus.rise(&btPlus) ; //interruptMoins.rise(&btMoins) ; //interruptSelect.rise(&btSelect) ; while (true) { // wait(0.2); ThisThread::yield() ; // btHome() ; // ThisThread::sleep_for(4000) ; // btSelect() ; // ThisThread::sleep_for(4000) ; // ThisThread::sleep_for(4000) ; // btPlus() ; // ThisThread::sleep_for(4000) ; // btSelect() ; // ThisThread::sleep_for(4000) ; // btMoins() ; // ThisThread::sleep_for(4000) ; // btSelect() ; // ThisThread::sleep_for(4000) ; }</pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	CEpaper.hpp	1/2
<pre>/*** * @class CEpaper * @brief Classe CEpaper : Classe qui paramètre l'affichage sur l'écran * @author Xavier Hervier * @version 1.0 * @date 3/02/2021 */ #ifndef CEPAPER_H #define CEPAPER_H //#include "global.h" #include <stdio.h> #include <mbed.h> #include <string> #include "epd2in7/epd2in7.h" #include "epd2in7/imagedata.h" #include "epd2in7/epdpaint.h" #define COLORED 1 #define UNCOLORED 0 using namespace std; class CEpaper { public: /** * @brief Constructeur surcharger d'un objet Epaper * @param epd Objet de la classe Epd */ CEpaper(Epd *epd); /** * @brief Destructeur d'un objet Epaper */ ~CEpaper(); /** * @brief Permet la rotation de l'affichage */ void rotate(); /** * @brief Permet d'effacer ou nettoyer l'affichage */ void effacer(); /** * @brief Permet l'écriture sur l'écran * @param x Position en abscisse de l'écriture * @param y Position en ordonné de l'écriture * @param texte Texte qui sera afficher sur l'écran * @param couleur Couleur qui correspond à noir ou blanc */ void Ecrire (int x, int y, string texte, bool couleur); /** </pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	CEpaper.hpp	2/2
<pre>* @brief Permet la mise à jour l'affichage */ void majAffichage(); private : // Pointeur vers un objet Epd Epd *_epd; // Variable pour le FrameBuffer unsigned char* _frameBuffer ; // Pointeur vers un objet Paint Paint *_paint; }; #endif // CEPAPER_H</pre>		

HERVIER XAVIER	Ecran.hpp	1/3
<pre>/** * @class Ecran * @brief Classe Ecran : Classe des différents écran * @author Xavier Hervier * @version 1.0 * @date 3/02/2021 */ #ifndef ECRAN_H #define ECRAN_H #include <iostream> #include <Item.hpp> #include <string> #include <vector> #include "CEpaper.hpp" #include "Item.hpp" using namespace std; class Ecran { public: /** * @brief Construit un objet Ecran * @param epaper Objet Epaper pour paramétrer l'affichage */ Ecran(CEpaper *epaper); /** * @brief Construit un objet Ecran * @param epaper Objet Epaper pour paramétrer l'affichage * @param titre Titre de l'écran afficher * @param accueil Désigne si cet ecran sera l'accueil ou pas * @param actif Désigne si l'écran est actif */ Ecran(CEpaper *epaper, string titre, bool accueil, bool actif);</pre>		

HERVIER XAVIER	Ecran.hpp	2/3
<pre> /*** * @brief Détruire un objet Ecran */ ~Ecran(); /*** * @brief Ajouter un objet Item * @param item L'item qui sera ajouté */ void addItem(Item * item); /*** * @brief Getteur de l'Accueil récupère l'accueil * @return bool */ bool getAccueil(); /*** * @brief Change vers l'item suivant * @return bool */ bool itemSuivant(); /*** * @brief Change vers l'item précédent * @return bool */ bool itemPrecedent(); /*** * @brief Récupère l'index de l'item actif * @return unsigned int */ unsigned int indexItemActif(); /*** * @brief Getteur de l'Item * @return Item */ Item * getItemActif(); /*** * @brief Afficher l'écran */ void afficher(); /*** * @brief Récupère la valeur de l'actif * @return bool */ bool getActif(); /*** * @brief Choisi si actif ou pas * @param value */ void setActif(bool value); private: // vecteur vers Item vector <Item *> _vectItems; // Variable désignant si Ecran actif ou non bool _actif = false; string _titre; </pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	Ecran.hpp	3/3
bool _accueil; CEpaper *_epaper ; }; #endif // ECRAN_H		

HERVIER XAVIER	IHM.hpp	1/2
/** * @class IHM * @brief Classe IHM : Classe affichage de l'IHM * @author Xavier Hervier * @version 1.0 * @date 3/02/2021 */ #ifndef IHM_H #define IHM_H #include <iostream> #include "Ecran.hpp" using namespace std; class IHM { public: /** * @brief Construit un objet IHM */ IHM(); /** * @brief Destructeur objet IHM */ ~IHM(); /** * @brief Fonction du Bouton Home qui permet le retour à l'accueil */ void btnHome(); /** * @brief Fonction du Bouton Plus qui permet le passage à l'item suivant */ void btnPlus(); /** * @brief Fonction du Bouton Moins qui permet le passage à l'item précédent */ void btnMoins(); /** * @brief Fonction du Bouton Valid qui permet l'entrée dans un ecran soit mettre actif un ecran */ void btnValid(); /**		

HERVIER XAVIER	IHM.hpp	2/2
<pre> * @brief Ajoute un ecran * @param ecran Objet de la classe Ecran */ void addEcran(Ecran * ecran); /** * @brief Affiche * @return booléen */ bool afficher(); private: // Vecteur vers Ecran permet de récupérer les objets Ecran std::vector <Ecran *> _vectEcran; /** * @brief Récupère l'Ecran Actif * @return un Ecran */ Ecran* _getEcranActif(); /** * @brief Met l'Ecran Actif celon l'index de l'Ecran * @param indexEcran l'index de l'Ecran à mettre actif sur le vecteur */ void _setEcranActif(unsigned int indexEcran); /** * @brief Met l'Ecran Accueil en Actif */ void _setEcranActifAccueil(); /** * @brief Met l'Item suivant en Actif * @return true si l'Item suivant est Actif */ bool _setItemSuivantActif(); /** * @brief Met l'Item precedant en Actif * @return true si l'Item precedant est Actif */ bool _setItemPrecedentActif(); }; #endif // IHM_H </pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	Item.hpp	1/2
<pre>/*** * @class Item * @brief Classe Item : Classe qui permet de gérer les différents Items * @author Xavier Hervier * @version 1.0 * @date 3/02/2021 */ #ifndef ITEM_HPP #define ITEM_HPP #include <iostream> #include <Variable.hpp> #include <string> #include "CEpaper.hpp" using namespace std; class Item { public: /*** * @brief Construit un objet Item * @param epaper l'objet Epaper pour affichage sur Ecran */ Item(CEpaper *epaper); /*** * @brief Construit un objet Item * @param epaper l'objet Epaper pour affichage sur Ecran * @param text Titre objet Item */ Item(CEpaper *epaper, string text); /*** * @brief Construit un objet Item * @param epaper l'objet Epaper pour affichage sur Ecran * @param text Titre objet Item * @param actif Si Item actif ou non actif */ Item(CEpaper *epaper, string text, bool actif); /*** * @brief Construit un objet Item * @param epaper l'objet Epaper pour affichage sur Ecran * @param text Titre objet Item * @param actif Si Item actif ou non actif * @param objVariable Variable de l'Item */ Item(CEpaper *epaper, string text, bool actif, Variable *objVariable); /*** * @brief Destructeur objet Item */ ~Item(); /*** * @brief getteur si Item Actif * @return Retourne si actif Vrai ou Faux */ </pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	Item.hpp	2/2
<pre>/* bool getActif(); /** * @brief setteur Item Actif * @param valeur Modifie la valeur de actif */ void setActif(bool valeur); /** * @brief getteur de edition récupère la valeur de _edition * @return Retourne si edition Vrai ou Faux */ bool getEdition(); /** * @brief setteur Item edition * @param edition Modifie la valeur de actif soit vrai ou fausse */ void setEdition(bool edition); /** * @brief Diminuer */ void plus(); /** * @brief Moins */ void moins(); /** * @brief Affiche l'item * @param y Ordonnée du texte afficher */ void affiche(int y); /** * @brief Permet de récupérer le texte * @return La suite de caractère ou plutot le texte */ string getText(); private: // texte contenu et qui sera afficher sur l'écran pour l'Item string _text; // Valeur qui met les Item actif il est de base en false // pour éviter de sélectionner plusieur Item à la fois bool _actif = false; // Valeur qui met les Item en mode edition qui si il y a une Variable // assigner à l'Item de base en false pour eviter plusieur édition en meme temps bool _edition = false; // Pointeur vers l'objet Variable Variable *_objVariable; // Pointeur vers l'objet CEpaper CEpaper *_epaper; }; #endif // ITEM_H</pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	Variable.hpp	1/2
<pre>/*** * @class Variable * @brief Classe Variable : Classe des Variable sur l'IHM * @author Xavier Hervier * @version 1.0 * @date 3/02/2021 */ #ifndef VARIABLE_H #define VARIABLE_H #include <iostream> using namespace std; class Variable { public: /*** * @brief Constructeur objet Variable */ Variable(); /*** * @brief Construit un objet Variable * @param valinit Valeur initiale * @param inc Valeur de l'incrémentation * @param min Valeur Minimale * @param max Valeur Maximale */ Variable(int valinit , int inc, int min, int max); /*** * @brief Destructeur objet Variable */ ~Variable(); /*** * @brief Permet l'augmentation d'une variable * @return l'entier de la valeur */ int plus(); /*** * @brief Permet la diminution de la variable * @return l'entier de la valeur */ int moins(); /*** * @brief getteur de la valeur * @return retourne la valeur */ int getValeur(); /*** * @brief setteur de la Variable * @param value valeur de la variable */ void setValeur(int value); */ </pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	Variable.hpp	2/2
<pre> * @brief getteur de la valeur minimal * @return _min */ int getMin(); /** * @brief Permet de changer la valeur minimum * @param value Valeur qui sera rentrer pour la valeur minimum */ void setMin(int value); /** * @brief Récuperer la valeur Maximale * @return Retourne la valeur _max */ int getMax(); /** * @brief Permet de changer la valeur minimum * @param value Valeur qui sera rentrer pour la valeur minimum */ void setMax(int value); /** * @brief Récuperer la valeur Incrémentation * @return Retourne la valeur _inc */ int getInc(); /** * @brief Permet de changer la valeur de l'incrémentation * @param value Valeur qui sera rentrer pour la valeur incrémentation */ void setInc(int value); private: // Valeur par défaut int _valeur; // Valeur minimale int _min; // Valeur macimale int _max; // Taux d'incrémentation int _inc; }; #endif // VARIABLE_H </pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	CEpaper.cpp	1/2
<pre>/*** * @file CEpaper.cpp * @author Hervier Xavier * @brief Méthode de la classe CEpaper * @version 1.0 * @date 2021-03-31 * * @copyright Copyright (c) 2021 * */ #include "CEpaper.hpp" #include <mbed.h> /*** * @brief Constructeur surcharger d'un objet Epaper * @param epd Objet de la classe Epd */ using namespace std; CEpaper::CEpaper(Epd *epd){ _epd = epd; _frameBuffer = (unsigned char*)malloc(_epd->width / 8 * _epd->height); _paint = new Paint(_frameBuffer, _epd->width, _epd->height); } /*** * @brief Destructeur d'un objet Epaper */ CEpaper::~CEpaper(){ delete (_epd); delete (_frameBuffer); delete (_paint); } /*** * @brief Permet la rotation de l'affichage */ void CEpaper::rotate() { _paint->Clear(UNCOLORED); _paint->SetRotate(ROTATE_270); majAffichage() ; } /*** * @brief Permet d'effacer ou nettoyer l'affichage */ void CEpaper::effacer() { _paint->Clear(UNCOLORED); majAffichage() ; } /*** * @brief Permet l'écriture sur l'écran * @param x Position en abscisse de l'écriture * @param y Position en ordonné de l'écriture * @param texte Texte qui sera afficher sur l'écran * @param couleur Couleur qui correspond à noir ou blanc */ </pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	CEpaper.cpp	2/2
<pre>void CEpaper::Ecrire (int x, int y, string texte, bool couleur) { _paint->DrawStringAt(x, y, texte.c_str(), &Font16, couleur); } /*** * @brief Permet la mise à jour l'affichage */ void CEpaper::majAffichage () { _epd->DisplayFrame(_paint->GetImage()); }</pre>		

HERVIER XAVIER	Ecran.cpp	1/4
<pre>/** * @file Ecran.cpp * @author Hervier Xavier * @brief Méthode de la Classe Ecran * @version 1.0 * @date 2021-03-31 * * @copyright Copyright (c) 2021 * */ #include "Ecran.hpp" #include <iostream> #include "ThisThread.h" #include "mbed.h" #include "Item.hpp" #define SLEEP_TIME 500 // (msec) #define COLORED 1 #define UNCOLORED 0 using namespace std; /** * @brief Construit un objet Ecran * @param epaper Objet Epaper pour paramétrer l'affichage */ Ecran::Ecran(CEpaper *epaper) { _actif = false; _epaper = epaper ; }; /*** * @brief Construit un objet Ecran * @param epaper Objet Epaper pour paramétrer l'affichage * @param titre Titre de l'écran afficher * @param accueil Désigne si cet ecran sera l'accueil ou pas * @param actif Désigne si l'écran est actif */</pre>		

HERVIER XAVIER	Ecran.cpp	2/4
<pre> Ecran::Ecran(CEpaper *epaper, string titre, bool accueil, bool actif) { _titre =(titre); _accueil = (accueil); _actif = (actif); _epaper = epaper ; }; /** * @brief Détruire un objet Ecran */ Ecran::~Ecran() { }; /** * @brief Ajouter un objet Item * @param item L'item qui sera ajouté */ void Ecran::addItem(Item * item) { _vectItems.push_back(item); }; /** * @brief Getteur de l'Accueil récupère l'accueil * @return bool */ bool Ecran::getAccueil() { return this ->_accueil; }; /** * @brief Change vers l'item suivant * @return bool */ bool Ecran::itemSuivant() { for (int i = 0; i < _vectItems.size(); i++) { if (_vectItems[i]->getActif() == true) { _vectItems[i]->setActif(false); if(i+1 < _vectItems.size()){ _vectItems[i+1]->setActif(true); }else{ _vectItems[0]->setActif(true); } return true; } } return false; }; /** * @brief Change vers l'item precedent * @return bool */ bool Ecran::itemPrecedent() </pre>		

HERVIER XAVIER	Ecran.cpp	3/4
<pre>{ for (int i = 0; i < _vectItems.size(); i++) { if (_vectItems[i]->getActif() == true) { _vectItems[i]->setActif(false); if(i-1 >= 0){ _vectItems[i-1]->setActif(true); }else{ _vectItems[_vectItems.size()-1]->setActif(true); } return true; } } return false; }; /*** * @brief Récupère l'index de l'item actif * @return unsigned int */ unsigned int Ecran::indexItemActif() { for(int i=0;i<_vectItems.size();i++) { if(_vectItems.at(i)->getActif()==true) { return i; } } }; /*** * @brief Récupère l'Item actif * @return Item */ Item * Ecran::getItemActif() { for(int i=0;i<_vectItems.size();i++) { if(_vectItems.at(i)->getActif()==true) { return this->_vectItems.at(i); } } }; /*** * @brief Afficher l'écran */ void Ecran::afficher() { string texteItem ; printf("%s \r\n",_titre.c_str()); _epaper->effacer(); _epaper->Ecrire(10, 10, _titre.c_str(), COLORED); for(int i=0;i<_vectItems.size();i++) </pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	Ecran.cpp	4/4
<pre>{ if (_vectItems.at(i)->getActif() == true) { texteItem = "->" + _vectItems.at(i)->getText() ; _epaper->Ecrire(10, 40 + i*25, texteItem.c_str(), COLORED); } else { texteItem = _vectItems.at(i)->getText() ; _epaper->Ecrire(10, 40 + i*20, texteItem.c_str(), COLORED); } printf("%s \r\n",texteItem.c_str()); ThisThread::sleep_for(SLEEP_TIME); } _epaper->majAffichage() ; }; /*** * @brief Récupère la valeur de l'actif * @return bool */ bool Ecran::getActif() { return this -> _actif; }; /*** * @brief Choisi si actif ou pas * @param value */ void Ecran::setActif(bool value) { this -> _actif = value; };</pre>		

HERVIER XAVIER	IHM.cpp	1/4
<pre>/*** * @file IHM.cpp * @author Hervier Xavier * @brief Méthode de la Classe IHM * @version 1.0 * @date 2021-03-31 * * @copyright Copyright (c) 2021 * */ #include "IHM.hpp" #include "Ecran.hpp" #include <iostream> #define SLEEP_TIME 500 // (msec)</pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	IHM.cpp	2/4
<pre> using namespace std; /** * @brief Construit un objet IHM */ IHM::IHM() { }; /** * @brief Destructeur d'un objet IHM */ IHM::~IHM() { }; /** * @brief Fonction du Bouton Home qui permet le retour à l'accueil */ void IHM::btnHome() { _setEcranActifAccueil(); afficher(); }; /** * @brief Fonction du Bouton Plus qui permet le passage à l'item suivant */ void IHM::btnPlus() { Item *itemActif = _getEcranActif()->getItemActif(); if(itemActif != nullptr){ _getEcranActif()->itemSuivant(); afficher(); } }; /** * @brief Fonction du Bouton Moins qui permet le passage à l'item précédent */ void IHM::btnMoins() { Item *itemActif = _getEcranActif()->getItemActif(); if(itemActif != nullptr){ _getEcranActif()->itemPrecedent(); afficher(); } }; /** * @brief Fonction du Bouton Valid qui permet l'entrée dans un ecran soit mettre actif un ecran */ void IHM::btnValid() { if(_getEcranActif()->getAccueil() == true){ unsigned int ItemActif = _getEcranActif()->indexItemActif(); _setEcranActif(ItemActif); afficher(); } }; </pre>		

HERVIER XAVIER	IHM.cpp	3/4
<pre> } }; /** * @brief Ajoute un ecran * @param ecran Objet de la classe Ecran */ void IHM::addEcran(Ecran* ecran) { _vectEcran.push_back(ecran); }; void IHM::_setEcranActif(unsigned int indexEcran) { _getEcranActif()->setActif(false); _vectEcran.at(indexEcran+1)->setActif(true); }; /** * @brief Récupère l'Ecran Actif * @return un Ecran */ Ecran* IHM::_getEcranActif() { for (int unsigned i = 0; i < _vectEcran.size(); i++){ if (_vectEcran[i]->getActif()==true){ return this->_vectEcran.at(i); } } return nullptr; }; /** * @brief Met l'Ecran Actif celon l'index de l'Ecran * @param indexEcran l'index de l'Ecran à mettre actif sur le vecteur */ void IHM::_setEcranActifAccueil() { _getEcranActif()->setActif(false); _vectEcran.at(0)->setActif(true); }; bool IHM::_setItemSuivantActif() { for (int unsigned i = 0; i < _vectEcran.size(); i++){ return _vectEcran[i]->itemSuivant(); } }; /** * @brief Met l'Item precedant en Actif * @return true si l'Item precedant est Actif */ bool IHM::_setItemPrecedentActif() { for (int unsigned i = 0; i < _vectEcran.size(); i++){ return _vectEcran[i]->itemPrecedent(); } }; </pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	IHM.cpp	4/4
}; /** * @brief Affiche * @return booléen */ bool IHM::afficher() { for (int unsigned i = 0; i < _vectEcran.size(); i++) { if (_vectEcran[i]->getActif() == true){ _vectEcran[i]->afficher(); return true; } } return false; };		

HERVIER XAVIER	Item.cpp	1/3
/** * @file Item.cpp * @author Hervier Xavier * @brief Méthode de la classe Item * @version 1.0 * @date 2021-03-31 * * @copyright Copyright (c) 2021 * */ #include "Item.hpp" #include <iostream> #include "ThisThread.h" #include "mbed.h" #define SLEEP_TIMES 100 // (msec) #define COLORED 1 #define UNCOLORED 0 using namespace std; /** * @brief Construit un objet Item * @param epaper l'objet Epaper pour affichage sur Ecran */ Item::Item(CEpaper *epaper) { _epaper = epaper; }; /** * @brief Construit un objet Item * @param epaper l'objet Epaper pour affichage sur Ecran * @param text Titre objet Item */ Item::Item(CEpaper *epaper, string text) {		

HERVIER XAVIER	Item.cpp	2/3
<pre> _epaper = epaper; _text = text; }; /*** * @brief Construit un objet Item * @param epaper l'objet Epaper pour affichage sur Ecran * @param text Titre objet Item * @param actif Si Item actif ou non actif */ Item::Item(CEpaper *epaper, string text,bool actif) { _epaper = epaper; _text = text; _actif = actif; }; /*** * @brief Construit un objet Item * @param epaper l'objet Epaper pour affichage sur Ecran * @param text Titre objet Item * @param actif Si Item actif ou non actif * @param objVariable Variable de l'Item */ Item::Item(CEpaper *epaper, string text,bool actif, Variable * objVariable) { _epaper = epaper; _text = text; _actif = actif; _objVariable = objVariable; }; /*** * @brief Destructeur objet Item */ Item::~Item() { }; /*** * @brief getteur si Item Actif * @return Retourne si actif Vrai ou Faux */ bool Item::getActif() { return this -> _actif; }; /*** * @brief setteur Item Actif * @param valeur Modifie la valeur de actif */ void Item::setActif(bool value) { this -> _actif = value; }; /*** * @brief Affiche l'item * @param y Ordonnée du texte afficher </pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	Item.cpp	3/3
<pre>/* void Item::affiche(int y) { if (_actif == true){ //_text = "->%s", _text ; //_epaper->_paint->DrawStringAt(10, y, _text.c_str(), &Font24, COLORED); printf("-> %s \r\n",_text.c_str()); } else { //_epaper->_paint->DrawStringAt(10, y, _text.c_str(), &Font24, COLORED); printf("%s \r\n",_text.c_str()); } ThisThread::sleep_for(SLEEP_TIMES); }; /** * @brief Permet de récupérer le texte * @return La suite de caractère ou plutot le texte */ string Item::getText() { return _text; }</pre>		

HERVIER XAVIER	Variable.cpp	1/3
<pre>/** * @file Variable.cpp * @author Hervier Xavier * @brief Méthode de la classe Variable * @version 1.0 * @date 2021-03-31 * * @copyright Copyright (c) 2021 * */ #include "Variable.hpp" #include <iostream> using namespace std; /** * @brief Construit un objet Variable * @param valinit Valeur initiale * @param inc Valeur de l'incrémentation * @param min Valeur Minimale * @param max Valeur Maximale */ Variable::Variable(int valinit, int inc, int min, int max) :_valeur(50), _inc(10), _min(20), _max(180) { };</pre>		

E1 : HERVIER Xavier

HERVIER XAVIER	Variable.cpp	2/3
<pre>/** * @brief Destructeur objet Variable */ Variable::~Variable() { }; /** * @brief Permet l'augmentation d'une variable * @return l'entier de la valeur */ int Variable::plus() { _valeur = _valeur + _inc; return _valeur; }; /** * @brief Permet la diminution de la variable * @return l'entier de la valeur */ int Variable::moins() { _valeur = _valeur - _inc; return _valeur; }; /** * @brief getteur de la valeur * @return retourne la valeur */ int Variable::getValeur() { return this -> _valeur; }; /** * @brief setteur de la Variable * @param value valeur de la variable */ void Variable::setValeur(int value) { this -> _valeur = value; }; /** * @brief getteur de la valeur minimal * @return _min */ int Variable::getMin() { return this -> _min; }; /** * @brief Permet de changer la valeur minimum * @param value Valeur qui sera rentrer pour la valeur minimum */ void Variable::setMin(int value) {</pre>		

HERVIER XAVIER	Variable.cpp	3/3
<pre> this -> _min = value; }; /**/ * @brief Récuperer la valeur Maximale * @return Retourne la valeur _max */ int Variable::getMax() { return this -> _max; }; /**/ * @brief Permet de changer la valeur minimum * @param value Valeur qui sera rentrer pour la valeur minimum */ void Variable::setMax(int value) { this -> _max = value; }; /**/ * @brief Récuperer la valeur Incrémentation * @return Retourne la valeur _inc */ int Variable::getInc() { return this -> _inc; }; /**/ * @brief Permet de changer la valeur de l'incrémentation * @param value Valeur qui sera rentrer pour la valeur incrémentation */ void Variable::setInc(int value) { this -> _inc = value; }; </pre>		

[CEnumVehicule.hpp](#)

```
/*
 * @file CEnumVehicule.hpp
 * @author Thibaut Jouandoudet
 * @brief
 * @version 0.1
 * @date 2021-03-31
 *
 * @copyright Copyright (c) 2021
 *
 */
#ifndef CENUMVEHICULE_HPP
#define CENUMVEHICULE_HPP

#include "mbed.h"

enum class CEnumVehicule {VL,PL};

#endif
```

[CVehiculeDetecte.hpp](#)

```
/*
 * @file CVehiculeDetecte.hpp
 * @author Thibaut Jouandoudet
 * @brief
 * @version 0.1
 * @date 2021-03-31
 *
 * @copyright Copyright (c) 2021
 *
 */
#ifndef CVEHICULEDETECTE_HPP
#define CVEHICULEDETECTE_HPP

#include "mbed.h"
#include "CEnumVehicule.hpp"

class CVehiculeDetecte
{
```

```
public:

/** 
 * @brief Construct a new CVehiculeDetecte object
 *
 */
CVehiculeDetecte();

/** 
 * @brief Destroy the CVehiculeDetecte object
 *
 */
~CVehiculeDetecte();

/** 
 * @brief Construct a new CVehiculeDetecte object
 *
 * @param typeVehicule
 * @param vitesse
 * @param nombreEssieu
 */
CVehiculeDetecte(CEnumVehicule typeVehicule, unsigned char vitesse,unsigned int
nombreEssieu);

/** 
 * @brief Get the typeVehicule object
 *
 * @return CEnumVehicule
 */
CEnumVehicule get_typeVehicule();

/** 
 * @brief Set the typeVehicule object
 *
 * @param _typeVehicule
 */
void set_typeVehicule(CEnumVehicule _typeVehicule);

/** 
 * @brief Get the vitesse object
 *
 */
```

```
* @return unsigned char
*/
unsigned char get_vitesse();

/** 
 * @brief Set the vitesse object
 *
 * @param _vitesse
 */
void set_vitesse(unsigned char _vitesse);

/** 
 * @brief Get the nombreEssieu object
 *
 * @return unsigned int
 */
unsigned int get_nombreEssieu();

/** 
 * @brief Set the nombreEssieu object
 *
 * @param _nombreEssieu
 */
void set_nombreEssieu(unsigned int _nombreEssieu);

/** 
 * @brief Affichage des paramètres
 *
 */
void afficherParam();

private:

CEnumVehicule _typeVehicule;
unsigned char _vitesse;
unsigned int _nombreEssieu;
float _pVitesse;
float _pDistInterEssieux;
int _pDecalage = 0;

};
```

#endif

[CVehiculeDetecte.cpp](#)

```

/**
 * @file CVehiculeDetecte.cpp
 * @author Thibaut Jouandoudet
 * @brief
 * @version 0.1
 * @date 2021-04-01
 *
 * @copyright Copyright (c) 2021
 *
 */

#include "CVehiculeDetecte.hpp"
#include "CEnumVehicule.hpp"
#include "mbed.h"
#include <iostream>
using namespace std;

/**
 * @brief Construct a new CVehiculeDetecte::CVehiculeDetecte object
 *
 */
CVehiculeDetecte::CVehiculeDetecte()
    : _typeVehicule(CEnumVehicule::VL)
    , _vitesse(30)
    , _nombreEssieu(2)
{
    // printf("nouveau Vehicule\r\n");
}

/**
 * @brief Construct a new CVehiculeDetecte::CVehiculeDetecte object
 *
 * @param typeVehicule
 * @param vitesse
 * @param nombreEssieu
 */
CVehiculeDetecte::CVehiculeDetecte(CEnumVehicule typeVehicule,unsigned char
vitesse,unsigned int nombreEssieu)
    : _typeVehicule(typeVehicule)

```

```
, _vitesse(vitesse)
, _nombreEssieu(nombreEssieu)
{
// printf("nouveau Vehicule\r\n");
}

/***
* @brief Destroy the CVehiculeDetecte::CVehiculeDetecte object
*
*/
CVehiculeDetecte::~CVehiculeDetecte()
{
// printf("Vehicule detruit\r\n");
}

/***
* @brief
*
* @param typeVehicule
*/
void CVehiculeDetecte::set_typeVehicule(CEnumVehicule typeVehicule)
{
_typeVehicule = typeVehicule ;
}

/***
* @brief
*
* @return CEnumVehicule
*/
CEnumVehicule CVehiculeDetecte::get_typeVehicule()
{
return _typeVehicule ;
}

/***
* @brief
*
* @return unsigned char
*/
unsigned char CVehiculeDetecte::get_vitesse()
{
```

```
    return _vitesse ;  
}  
  
/**  
 * @brief  
 *  
 * @param vitesse  
 */  
void CVehiculeDetecte::set_vitesse(unsigned char vitesse)  
{  
    _vitesse = vitesse ;  
}  
  
/**  
 * @brief  
 *  
 * @return unsigned int  
 */  
unsigned int CVehiculeDetecte::get_nombreEssieu()  
{  
    return _nombreEssieu ;  
}  
  
/**  
 * @brief  
 *  
 * @param nombreEssieu  
 */  
void CVehiculeDetecte::set_nombreEssieu(unsigned int nombreEssieu)  
{  
    _nombreEssieu = nombreEssieu ;  
}  
  
/**  
 * @brief  
 *  
 */  
void CVehiculeDetecte::afficherParam()  
{  
    if (_typeVehicule == CEnumVehicule::VL)  
    {  
        printf("Vehicule Leger\r\n");  
    }
```

```

printf("%d",_vitesse);
printf(" KM/H \r\n");
printf("essieu:%d \r\n",_nombreEssieu);
}
else
{
printf("Vehicule Lourd\r\n");
printf("%d",_vitesse);
printf(" KM/H \r\n");
printf("essieu:%d \r\n",_nombreEssieu);
}
}

```

[CDetecteurPassage.hpp](#)

```

/***
* @file CDetecteurPassage.hpp
* @author Thibaut Jouandoudet
* @brief
* @version 0.1
* @date 2021-03-31
*
* @copyright Copyright (c) 2021
*
*/
#ifndef CDETECTEURPASSAGE_HPP
#define CDETECTEURPASSAGE_HPP

#include "mbed.h"
#include "CEnumVehicule.hpp"
#include "CVehiculeDetecte.hpp"
#include "CStatsManager.hpp"

class CDetecteurPassage
{

public:

/***
* @brief Construct a new CDetecteurPassage object
*
*/

```

```
CDetecteurPassage();

/** 
 * @brief Destroy the CDetecteurPassage object
 *
 */
~CDetecteurPassage();

/** 
 * @brief
 *
 * @param tick
 */
void addTickDetection(unsigned long tick);

private:

CEnumVehicule identifierVehicule();

unsigned long _tickDetection[20];
unsigned char _indexDetection;
CVehiculeDetecte _vehiculeDetecte;
// CStatsManager* _statsManager;

};

#endif
```

[CDetecteurPassage.cpp](#)

```
/**  
 * @file CDetecteurPassage.cpp  
 * @author Thibaut Jouandoudet  
 * @brief  
 * @version 0.1  
 * @date 2021-04-01  
 *  
 * @copyright Copyright (c) 2021  
 *  
 */  
#include "mbed.h"  
#include "CDetecteurPassage.hpp"  
  
#define Pressostat_1  
#define Pressostat_2  
  
DigitalOut ledV(LED1); DigitalOut ledB(LED2); DigitalOut ledR(LED3);  
  
/**  
 * @brief Construct a new CDetecteurPassage::CDetecteurPassage object  
 *  
 */  
CDetecteurPassage::CDetecteurPassage()  
: _tickDetection()  
, _indexDetection()  
, _vehiculeDetecte()  
// , _statsManager()  
{  
    // Construct CDetecteurPassage  
}  
  
CDetecteurPassage::~CDetecteurPassage()  
{  
}  
  
void CDetecteurPassage::addTickDetection(unsigned long tick)  
{
```

```
}
```

```
CEnumVehicule CDetecteurPassage::identifierVehicule()
```

```
{
```

```
}
```

main.cpp

```
/* *
```

```
* @author Jouandoudet Thibaut
```

```
* @version 1.0
```

```
*/
```

```
#include "mbed.h"
```

```
#include "CVehiculeDetecte.h"
```

```
#include "CEnumVehicule.h"
```



```
#define PRESS_1 D0 // sortie
```

```
#define PRESS_2 D1 // sortie
```



```
#define DETECT_1 PE_15 // Entree
```

```
#define DETECT_2 PB_10 // Entree
```



```
unsigned long CptTicker[10][4];
```

```
char indexCptTickerP1 = 0 ;
```

```
char indexCptTickerP2 = 0 ;
```

```
unsigned long CptP1 = 0 ; unsigned long CptP2 = 0 ;
```



```
void detectP1(){
```

```
    CptP1++ ;
```

```
    CptTicker[indexCptTickerP1++][0] = us_ticker_read();
```

```
    if (indexCptTickerP1==3) indexCptTickerP1 = 0 ;
```



```
}
```



```
void detectP2(){
```

```
    CptP2++ ;
```

```
    CptTicker[indexCptTickerP2++][1] = us_ticker_read();
```

```
    if (indexCptTickerP2==3) indexCptTickerP2 = 0 ;
```



```
}
```



```
void affCptTicker()
```

```

{
    unsigned long dif = 0 ;

    double vitCalule ;
    double interessieu ;
    double interessieu1 ;
    double interessieu2 ;
    double vitCalcule;
    double vitCalcul;

    printf ("\n\r\t --- tab CptTicker ---");
    for (int i = 0 ; i <3 ; i++) {
        if ( CptTicker[i][1] >= CptTicker[i][0] )
            dif = CptTicker[i][1]- CptTicker[i][0] ;

        else
            dif = 0 ; //CptTicker[i][0]- CptTicker[i][1] ;

        printf ("\n\r\t %d : %lu ; %lu ; %lu", i, CptTicker[i][0], CptTicker[i][1], dif) ;
    }

// Permet de calculer la vitesse en KM/H
    vitCalule = dif/1000 ;
    vitCalcule = (1/vitCalule);
    vitCalcul = vitCalcule * 3600;

    interessieu1 = (vitCalcul * 1000 ) / 3600000 ; // Pasage en km/h en m/ms
    interessieu2 = (CptTicker[0][1]- CptTicker[0][0] + CptTicker[1][1]- CptTicker[1][0] +
    CptTicker[2][1]- CptTicker[2][0] ) / 1000 ; // différence des capteurs en ms
    interessieu = interessieu2 * interessieu1 ; // différence des capteurs * la vitesse en
    m/ms

    printf ("\n\r\tVitesse Calculer: = %lf KM/h" ,vitCalcul) ;
    printf ("\n\r\tInteressieu: = %lf m\n\r" , interessieu) ;

    if (interessieu >= 3.45 )
    {
        printf("Vehicule Lourd\r\n");
    }
}

```

```
else
{
printf("Vehicule Leger\r\n");
}

}

InterruptIn intP1(DETECT_1) ; // (PB_9) ;
InterruptIn intP2(DETECT_2) ; // (PA_6) ;

int main()

{
us_ticker_free () ;
us_ticker_init() ;

/*CVehiculeDetecte test (CEnumVehicule::PL,25,5);
test.afficherParam();
CVehiculeDetecte test2;
test2.afficherParam();*/

PinName pnPressostat1 = PRESS_1 ; //PB_0 ;// PB_8 ;
PinName pnPressostat2 = PRESS_2 ; //PA_0 ; // PA_5 ;

intP1.fall(&detectP1) ;
intP2.fall(&detectP2) ;

while (true) {

for(int v=100 ; v <= 140 ; v=v+10){
indexCptTickerP1 = indexCptTickerP2 = 0 ;
    ThisThread::sleep_for(12000);
    printf ("\r\t cpt1 : %lu ; cpt2 : %lu", CptP1, CptP2) ;
    affCptTicker() ;
}

}
}
```

PROCEDURE D'INSTALLATION :

- 1 Installer Visual Studio Code et l'extension Cortex-Debug
- 2 Installer le SEGGER
- 3 Décompresser l'archive d'ARM à l'emplacement de son choix
- 4 Installer mbed-cli
(ATTENTION : Redémarrer l'ordinateur et tester dans l'invite de commande la commande : mbed)
- 5 Si commande applet impossible alors voir version de Python et tester un upgrade de pip avec la commande : python -m pip install --upgrade pip // Redémarrer à nouveau)
- 6 Cloner un template fonctionnelle de mbed : http://gogs.lyceekastler.fr/hug.Carricart/Mbed_Template/
- 7 Réadapter le fichier de configuration : launch.json et c_cpp_properties.json
- 8 Coder!

CARRICART Hugo	launch.json	1/1
<pre>{ "version": "0.2.0", "configurations": [{ "type": "cortex-debug", "request": "launch", "name": "Debug J-Link", "cwd": "\${workspaceRoot}", "executable": "\${workspaceRoot}/BUILD/NUCLEO_F746ZG/GCC_ARM- DEBUG/\${workspaceRootFolderName}.elf", // "serverpath": "/opt/SEGGER/JLink/JLinkGDBServerCLExe", // Linux "serverpath": "C:/Program Files (x86)/SEGGER/JLink/JLinkGDBServerCL.exe", // Windows 10 "servertype": "jlink", "device": "STM32F746ZG", "interface": "swd", "runToMain": true, "gdbPath": "C:/Users/a/gcc-arm-none-eabi/bin/arm-none-eabi-gdb.exe", // Windows 10 //"gdbPath": "/opt/gcc-arm-none-eabi/bin/arm-none-eabi-gdb", // Linux Perso //"gdbPath": "/opt/gcc-arm-none-eabi-9-2019-q4-major/bin/arm-none-eabi-gdb", // Linux Lycée "svdFile": "\${workspaceRoot}/STM32F7x6.svd", "preLaunchTask": "mbed" }] }</pre>		

E3 : CARRICART Hugo

CARRICART Hugo	task.json	1/1
<pre>{ // See https://go.microsoft.com/fwlink/?LinkId=733558 // for the documentation about the tasks.json format "version": "2.0.0", "name": "mbed", "isShellCommand": true, "showOutput": "always", "problemMatcher": { "owner": "cpp", "fileLocation": ["relative", "\${workspaceRoot}"], "pattern": { "regexp": "^(.+):(\d+):(\d+):\s+(warning error):\s+(.+)\$", "file": 1, "line": 2, "column": 3, "severity": 4, "message": 5 } }, "args": ["compile", "--profile=debug", "-t", "GCC_ARM", "-m", "NUCLEO_F746ZG"], "linux": { "command": "mbed" }, "osx": { "command": "mbed" }, "windows": { "command": "mbed" } }</pre>		

CARRICART Hugo

settings.json

1/1

// Place your settings in this file to overwrite default and user settings.
{ "C_Cpp.addWorkspaceRootToIncludePath": false, "C_Cpp.intelliSenseEngine": "Tag Parser", // "C_Cpp.intelliSenseEngine": "Default", // "C_Cpp.intelliSenseEngineFallback": "Enabled", // "cortex-debug.JLinkGDBServerPath": "/opt/SEGGER/JLink/JLinkGDBServerCLExe", // linux // "cortex-debug.JLinkGDBServerPath": "C:/Program Files // (x86)/SEGGER/JLink/JLinkGDBServerCL.exe", // windows 10 "cmake.configureOnOpen": true, "files.associations": { "mstd_iterator": "cpp" }, }

CARRICART Hugo	c_cpp_properties.json	
{	<pre> "configurations": [{ "name": "Windows", "includePath": ["\${workspaceFolder}/**", "\${workspaceFolder}/mbed-os/**", "\${workspaceFolder}/Include/**", "\${workspaceFolder}/Source/**"], "defines": ["DEVICE_SPI=1", "TARGET_STM32F746xG", "DEVICE_EMAC=1", "__MBED__=1", "DEVICE_USBDEVICE=1", "__FPU_PRESENT=1", "DEVICE_PORTOUT=1", "USBHOST_OTHER", "DEVICE_PORTINOUT=1", "__MBED_CMSIS_RTOS_CM", "TARGET_STM32F7", "DEVICE_MPU=1", "DEVICE_SERIAL_ASYNCH=1", "__CMSIS_RTOS", "TARGET_NAME=NUCLEO_F746ZG", "DEVICE_ANALOGOUT=1", "TARGET_STM32F746ZG", "TOOLCHAIN_GCC", "DEVICE_CRC=1", "ARM_MATH_CM7", "TARGET_CORTEX_M", "TARGET_LIKE_CORTEX_M7", "DEVICE_RTC=1", "DEVICE_TRNG=1", "TARGET_M7", "COMPONENT_PSA_SRV_IMPL=1", "EXTRA_IDLE_STACK_REQUIRED", "DEVICE_LPTICKER=1", "DEVICE_PWMOUT=1", "DEVICE_SPI_ASYNCH=1", "MBED_TICKLESS", "USE_FULL_LL_DRIVER", "TARGET_CORTEX", "DEVICE_I2C=1", "TRANSACTION_QUEUE_SIZE_SPI=2", "DEVICE_I2CSLAVE=1", "DEVICE_STDIO_MESSAGES=1",] }] }</pre>	

```

    "__CORTEX_M7",
    "TARGET_STM32F746",
    "TARGET_FAMILY_STM32",
    "USE_HAL_DRIVER",
    "TARGET_FF_ARDUINO",
    "DEVICE_PORTIN=1",
    "TARGET_RELEASE",
    "TARGET_STM",
    "DEVICE_SERIAL_FC=1",
    "COMPONENT_PSA_SRV_EMUL=1",
    "DEVICE_USTICKER=1",
    "DEVICE_WATCHDOG=1",
    "TARGET_NUCLEO_F746ZG",
    "TARGET_LIKEMBED",
    "TARGET_RTOS_M4_M7",
    "DEVICE_SLEEP=1",
    "TOOLCHAIN_GCC_ARM",
    "DEVICE_RESET_REASON=1",
    "DEVICE_CAN=1",
    "COMPONENT_NSPE=1",
    "DEVICE_INTERRUPTIN=1",
    "DEVICE_SPISLAVE=1",
    "DEVICE_ANALOGIN=1",
    "DEVICE_SERIAL=1",
    "MBED_BUILD_TIMESTAMP=1573037844.02",
    "DEVICE_FLASH=1",
    "DEVICE_I2C_ASYNC=1",
    "TARGET_STM_EMAC"
],
// "compilerPath": "C:/Users/a/gcc-arm-none-eabi/bin/arm-none-eabi-gdb.exe",
"cStandard": "c11",
"cppStandard": "c++17",
"intelliSenseMode": "gcc-x64"
},
{
  "name": "Linux",

  "forcedInclude": [
    "${workspaceFolder}/mbed_config.h"
  ],
  "includePath": [
    "${workspaceFolder}/**",
    "${workspaceFolder}/mbed-os/**",
    "${workspaceFolder}/Include/**",
    "${workspaceFolder}/Source/**",
    "${workspaceFolder}/epd2in7/*"
  ],
  "defines": [
    "DEVICE_SPI=1",
    "TARGET_STM32F746xG",
    "DEVICE_EMAC=1",
    "__MBED__=1",
  ]
}

```

```
"DEVICE_USBDEVICE=1",
"__FPU_PRESENT=1",
"DEVICE_PORTOUT=1",
"USBHOST_OTHER",
"DEVICE_PORTINOUT=1",
"__MBED_CMSIS_RTOS_CM",
"TARGET_STM32F7",
"DEVICE_MPUI=1",
"DEVICE_SERIAL_ASYNCH=1",
"__CMSIS_RTOS",
"TARGET_NAME=NUCLEO_F746ZG",
"DEVICE_ANALOGOUT=1",
"TARGET_STM32F746ZG",
"TOOLCHAIN_GCC",
"DEVICE_CRC=1",
"ARM_MATH_CM7",
"TARGET_CORTEX_M",
"TARGET_LIKE_CORTEX_M7",
"DEVICE_RTC=1",
"DEVICE_TRNG=1",
"TARGET_M7",
"COMPONENT_PSA_SRV_IMPL=1",
"EXTRA_IDLE_STACK_REQUIRED",
"DEVICE_LPTICKER=1",
"DEVICE_PWMOUT=1",
"DEVICE_SPI_ASYNCH=1",
"MBED_TICKLESS",
"USE_FULL_LL_DRIVER",
"TARGET_CORTEX",
"DEVICE_I2C=1",
"TRANSACTION_QUEUE_SIZE_SPI=2",
"DEVICE_I2CSLAVE=1",
"DEVICE_STDIO_MESSAGES=1",
"__CORTEX_M7",
"TARGET_STM32F746",
"TARGET_FAMILY_STM32",
"USE_HAL_DRIVER",
"TARGET_FF_ARDUINO",
"DEVICE_PORTIN=1",
"TARGET_RELEASE",
"TARGET_STM",
"DEVICE_SERIAL_FC=1",
"COMPONENT_PSA_SRV_EMUL=1",
"DEVICE_USTICKER=1",
"DEVICE_WATCHDOG=1",
"TARGET_NUCLEO_F746ZG",
"TARGET_LIKE_MBED",
"TARGET_RTOS_M4_M7",
"DEVICE_SLEEP=1",
"TOOLCHAIN_GCC_ARM",
"DEVICE_RESET_REASON=1",
"DEVICE_CAN=1",
```

E3 : CARRICART Hugo

```
"COMPONENT_NSPE=1",
"DEVICE_INTERRUPTIN=1",
"DEVICE_SPISLAVE=1",
"DEVICE_ANALOGIN=1",
"DEVICE_SERIAL=1",
"MBED_BUILD_TIMESTAMP=1573037844.02",
"DEVICE_FLASH=1",
"DEVICE_I2C_ASYNCH=1",
"TARGET_STM_EMAC"
],
// "compilerPath": "/opt/gcc-arm-none-eabi/bin/arm-none-eabi-gdb",
"cStandard": "c11",
"cppStandard": "c++17",
"intelliSenseMode": "gcc-x64"
}
],
"version": 4
}
```

CARRICART Hugo	CModemSigfox.hpp	
/** * @file CModemSigfox.hpp * @author Marc Lajus // Hugo Carricart * @brief * @version 0.1 * @date 2021-03-31 * * @copyright Copyright (c) 2021 * */ #pragma once #include "mbed.h" #include "PrintfBufferedSerial.hpp" /* Niveaux de debug 0:aucun 1:bref 2:détaillé */ #define DEBUG_SIGFOX 2 class Sigfox { public: /** * @brief Construit un objet Sigfox * * @param pUartModem liaison série modeme Sigfox * @param pSerialDebug ls pour affichage infos debug */ Sigfox(PrintfBufferedSerial *pUartModem, PrintfBufferedSerial *pSerialDebug); ~Sigfox(); /** * @brief Envoie une commande (AT) et attend une chaîne OK en retour * * @param commande * @param timeout timeout en secondes * @return true si chaîne reçue commence par "OK" */ bool commandeOk(char * commande, float timeout); /** * @brief teste le modem jusqu'à ce qu'il réponde OK * */ void initOK(); /** * @brief positionne le modem sur clé publique ou privée */		

E3 : CARRICART Hugo

```
* @param publique true = clé publique (simulateur)
*           false = clé privée (réseau sigfox)
*/
void setClePublique(bool publique);

/***
* @brief retourne le device ID
*
* @return char*
*/
char * getDeviceID();

/***
* @brief envoie un message sigfox
*
* @param payload : contenu du message en hexa/ASCII. Max 12 octets = 24 chiffres Hexa
* @return true si message envoyé OK
*/
bool sendMessage(char *payload);

private:

// port série modem SIGFOX
PrintfBufferedSerial *_pUartModem;
// port série USB SERIAL
PrintfBufferedSerial *_pSerialDebug;
// tampon réception modem
char tamponRx[30];
Timer timer;
};
```

CARRICART Hugo	CModemSigfox.cpp	
<pre>/** * @file Sigfox.cpp * @author Marc Lajus // Hugo Carricart * @brief * @version 0.1 * @date 2021-03-31 * * @copyright Copyright (c) 2021 * */ #include "CModemSigfox.hpp" /** * @brief Construct a new Sigfox:: Sigfox object * * @param pUartModem * @param pSerialDebug */ Sigfox::Sigfox(PrintfBufferedSerial *pUartModem, PrintfBufferedSerial *pSerialDebug) { // affectation liaisons série _pUartModem = pUartModem; _pSerialDebug = pSerialDebug; } /** * @brief Destroy the Sigfox:: Sigfox object * */ Sigfox::~Sigfox() { } /** * @brief * * @param commande * @param timeout * @return true * @return false */ bool Sigfox::commandeOk(char *commande, float timeout) { // on commence par vider le tampon de réception while (_pUartModem->readable()) _pUartModem->getc(); // écrit la commande _pUartModem->printf("%s\r\n", commande); if (DEBUG_SIGFOX > 1)</pre>		

```

_pSerialDebug->printf("commande modem : %s\r\n", commande);

// lit une ligne en réponse; avec timeout
int i = 0;
char c;
timer.reset();
timer.start();
do
{
    if (_pUartModem->readable())
    {
        c = _pUartModem->getc();
        tamponRx[i++] = c;
    }
    else
    {
        ThisThread::yield();
    }
} while ((c != '\n') && (timer.read() < timeout));
tamponRx[i] = 0;
timer.stop();
if (DEBUG_SIGFOX > 1)
    _pSerialDebug->printf("réponse modem recue : %s\r\n", tamponRx);
// teste si réponse commence par "OK"
if (tamponRx[0] == 'O' && tamponRx[1] == 'K')
{
    return true;
}
else
{
    return false;
}

/***
 * @brief
 *
 */
void Sigfox::initOK()
{
    while (true)
    {
        if (commandeOk("AT", 1))
        {
            if (DEBUG_SIGFOX)
                _pSerialDebug->printf("modem répond OK\r\n");
            break;
        }
        else if (DEBUG_SIGFOX)
            _pSerialDebug->printf("erreur modem\r\n");
        ThisThread::sleep_for(1000);
    }
}

```

```

}

/***
 * @brief
 *
 * @param publique
 */
void Sigfox::setClePublique(bool publique)
{
    while (true)
    {
        char *commande;
        char *info;
        if (publique)
        {
            commande = "ATS410=1";
            info = "reseau simulateur";
        }
        else
        {
            commande = "ATS410=0";
            info = "reseau SIGFOX";
        }

        if (commandeOk(commande, 2))
        {
            if (DEBUG_SIGFOX)
                _pSerialDebug->printf("modem sur %s OK\r\n", info);
            break;
        }
        else if (DEBUG_SIGFOX)
            _pSerialDebug->printf("erreur modem sur %s\r\n", info);
        ThisThread::sleep_for(1000);
    }
}

/***
 * @brief
 *
 * @return char*
 */
char *Sigfox::getDeviceID()
{
    commandeOk("AT$I=10", 1);
    return tamponRx;
}

/***
 * @brief
 *
 * @param payload
 * @return true
 */

```

```
* @return false
*/
bool Sigfox::sendMessage(char *payload)
{
    char chaine[31];
    sprintf(chaine, "AT$SF=%s", payload);
    if (DEBUG_SIGFOX)
        _pSerialDebug->printf("envoi message Sigfox : %s\r\n", payload);
    bool result = commandeOk(chaine, 10);
    if (DEBUG_SIGFOX)
        if (result)
            _pSerialDebug->printf("message Sigfox envoye\r\n");
        else
            _pSerialDebug->printf("erreur envoi message Sigfox\r\n");
    return result;
}
```

CARRICART Hugo	CTrameStatsSigfox.hpp	
	<pre>/** * @file CTrameStatsSigFox.h * @brief * @author Hugo Carricart * @date 04/02/2021 * @version 1.0 * **/ #ifndef CTRAMESTATSSIGFOX_HPP #define CTRAMESTATSSIGFOX_HPP #include "CStatsVehicule.hpp" #include "StrucTrameStatsSigFox.h" #include "mbed.h" class CTrameStatsSigFox { public: /** * @brief Construct a new CTrameStatsSigFox object * */ CTrameStatsSigFox(); /** * @brief Destroy the CTrameStatsSigFox object * */ ~CTrameStatsSigFox(); /** * @brief Get the Trame object * * @return char */ char *getTrame(); /** * @brief * * @param statsV */ void fabriquerTrame(CStatsVehicule *statsV); /** * @brief */ }</pre>	

E3 : CARRICART Hugo

```
* @param nbreEssieu
*/
void fabriquerTrame(unsigned int nbreEssieu);

private:

StrucTrameStatsSigFox _unk;
unsigned char _trame[13];
unsigned char _sTrame[25];
char *_eTrame;

};

#endif //CTRAMESTATSSIGFOX_H
```

CARRICART Hugo	CTrameStatsSigfox.cpp	
/** * @file CTrameStatsSigFox.cpp * @brief * @author Hugo Carricart * @date 04/02/2021 * @version 1.0 * **/	#include "CTrameStatsSigFox.hpp" /** * @brief Construct a new CTrameStatsSigFox::CTrameStatsSigFox object * */ CTrameStatsSigFox::CTrameStatsSigFox() :_unk() , _trame() , _sTrame() , _eTrame() { } /** * @brief Destroy the CTrameStatsSigFox::CTrameStatsSigFox object * */ CTrameStatsSigFox::~CTrameStatsSigFox() { } /** * @brief Get the Trame object * * @return char */ char * CTrameStatsSigFox::getTrame() { return this -> _eTrame; } /** * @brief * * @param statsVL * @param statsPL * @param nbreEssieu */ void CTrameStatsSigFox::fabriquerTrame(CStatsVehicule *statsV) {	

E3 : CARRICART Hugo

```
_unk.typeStat = statsV->getTypeStat();
_unk.nbrVehicule = statsV->getNbrVehicule();
_unk.nbrEssieu = statsV->getNbrEssieu();
_unk.VitesseMoy = statsV->getVitesseMoy();
_unk.VitesseMax = statsV->getVitesseMax();
_unk.nbrVitesseOK = statsV->getNbrVitesseOK();
_unk.nbrLM20 = statsV->getNbrVitesseLM20();
_unk.nbrLP20 = statsV->getNbrVitesseLP20();
_unk.nbrLP30 = statsV->getNbrVitesseLP30();
_unk.nbrLP40 = statsV->getNbrVitesseLP40();
_unk.nbrLP50 = statsV->getNbrVitesseLP50();

unsigned int c;
unsigned int d;
unsigned int u;
unsigned int e;

_trame[0] = (_unk.typeStat) ;
_trame[0] <<=7;
c = _unk.nbrVehicule & 0x700;
c >>=4;
d = _unk.nbrVehicule & 0x0f0;
d >>=4;
_trame[0] = _trame[0] + c + d;

u = _unk.nbrVehicule & 0x00f;
u <<=4;
c = _unk.nbrEssieu & 0x01f00;
c >>=9;
_trame[1] = u + c;

c = _unk.nbrEssieu & 0x01ff;
c >>=1;
_trame[2] = c;

u = _unk.nbrEssieu & 0x01;
u <<=7;
d = _unk.VitesseMoy & 0xfe;
d >>=1;
_trame[3] = u + d;

u = _unk.VitesseMoy & 0x01;
u <<=7;
d = _unk.VitesseMax & 0xff;
d >>=1;
_trame[4] = d + u ;

u = _unk.VitesseMax & 0x01;
u <<=7;
c = _unk.nbrVitesseOK & 0x7ff;
c >>=4;
_trame[5] = c + u ;
```

```

u = _unk.nbrVitesseOK & 0x00f;
u <<=4;
c = _unk.nbrLM20 & 0x780;
c >>=7;
_trame[6] = u + c ;

d = _unk.nbrLM20 & 0x0f0;
d <<=1;
c = _unk.nbrLM20 & 0x00f;
c <<=1;
u = _unk.nbrLP20 & 0x200;
u >>=9;
_trame[7] = d + c + u ;

c = _unk.nbrLP20 & 0x100;
c >>=1;
d = _unk.nbrLP20 & 0x0f0;
d >>=1;
u = _unk.nbrLP20 & 0x00f;
u >>=1;
_trame[8] = d + c + u ;

u = _unk.nbrLP20 & 0x001;
u <<=7;
c = _unk.nbrLP30 & 0x0100;
c >>=2;
d = _unk.nbrLP30 & 0x0f0;
d >>=2;
e = _unk.nbrLP30 & 0x00c;
e >>=2;
_trame[9] = d + c + u + e;

u = _unk.nbrLP30 & 0x003;
u <<=6;
d = _unk.nbrLP40 & 0x07e;
d >>=1;
_trame[10] = d + u ;

u = _unk.nbrLP40 & 0x001;
u <<=7;
d = _unk.nbrLP50 & 0x07f;
_trame[11] = d + u ;

_trame[12] = '\0' ;

```

// Affichage après fabrication

```

printf("Trame apres fabrication : ");
for(int i = 0; i < 13 ; i++)
{

```

```

    printf("%0x ", _trame[i]);
}
printf(" \r\n");

// Partie conversion
// Passage de l'hexa en code ASCII (pour faire une chaîne de caractère)
for(int i = 0; i < 12; i++){
    _sTrame[i*2] = (_trame[i] >> 4) + 0x30;
    if( (_trame[i] >> 4) >= 0x0A){
        _sTrame[i*2] += 7;
    }
    _sTrame[(i*2)+1] = (_trame[i] & 0x0F) + 0x30;
    if( (_trame[i] & 0x0F) >= 0x0A){
        _sTrame[(i*2)+1] += 7;
    }
}
_sTrame[24] = '\0';

printf("Affichage sTrame : %s \r\n", _sTrame);

_eTrame =(char*)_sTrame; // CAST

printf("Affichage eTrame : %s \r\n", _eTrame);
}

/***
 * @brief
 *
 * @param nbreEssieu
 */
void CTrameStatsSigFox::fabriquerTrame(unsigned int nbreEssieu)
{
}

```

CARRICART Hugo	CStatsManager.hpp	
/** * @file CStatsManager.h * @brief * @author Hugo Carricart * @date 04/02/2021 * @version 1.0 * ***/ #ifndef CSTATSMANAGER_HPP #define CSTATSMANAGER_HPP #include "CTrameStatsSigFox.hpp" #include "CModemSigfox.hpp" #include "CStatsEssieu.hpp" #include "CStatsVehicule.hpp" #include "mbed.h" #include "CDetecteurPassage.hpp" class CStatsManager { public: /** * @brief Construct a new CStatsManager object * */ CStatsManager(); /** * @brief Destroy the CStatsManager object * */ ~CStatsManager(); /** * @brief Set the Vitesse Tronc object * * @param value */ void setVitesseTronc(unsigned int value); /** * @brief Get the Vitesse Tronc object * * @return unsigned int */ unsigned int getVitesseTronc();		

```
/***
 * @brief Set the Nombre Capteur object
 *
 * @param value
 */
void setNombreCapteur(unsigned int value);

/***
 * @brief Get the Nombre Capteur object
 *
 * @return unsigned int
 */
unsigned int getNombreCapteur();

/***
 * @brief Set the Mode Test object
 *
 * @param value
 */
void setModeTest(bool value);

/***
 * @brief Get the Mode Test object
 *
 * @return true
 * @return false
 */
bool getModeTest();

/***
 * @brief
 */
void start();

/***
 * @brief
 */
void initModem();

private:

/***
 * @brief
 *
 * @return CModemSigFox
 */
Sigfox _modemSigFox;

/***
 * @brief
 */
```

```
*  
*/  
CTrameStatsSigFox _trameStatsVL;  
  
/**  
 * @brief  
 *  
 */  
CTrameStatsSigFox _trameStatsPL;  
  
/**  
 * @brief  
 *  
 * @return CStatsVehicule  
 */  
CStatsVehicule _statsVL;  
  
/**  
 * @brief  
 *  
 * @return CStatsVehicule  
 */  
CStatsVehicule _statsPL;  
  
/**  
 * @brief  
 *  
 * @return CStatsEssieu  
 */  
CStatsEssieu _statsEssieu;  
  
unsigned int _vitesseTronc; // CItem // CStatVehicule  
unsigned int _nbrCapteur; // CItem  
bool _modeTest; // CItem  
  
};  
  
#endif
```

CARRICART Hugo	CStatsManager.cpp	
/** * @file CStatsManager.cpp * @brief * @author Hugo Carricart * @date 04/02/2021 * @version 1.0 * **/	#include "CStatsManager.hpp" #include "CVehiculeDetecte.hpp" #define SERIAL_DEBUG USBTX,USBRX #define BAUDS_DEBUG 115200 #define SERIAL_SIGFOX D1,D0 #define BAUDS_SIGFOX 9600 #define SIGFOX_SIMUL true /* Liaison série pour infos de débogage lire sur le PC avec screen /dev/ttyACM0 xxxbauds Note : utilisée par 2 threads */ PrintfBufferedSerial pc(SERIAL_DEBUG); // Modem Sigfox PrintfBufferedSerial uartSigfox(SERIAL_SIGFOX); /** * @brief Construct a new CStatsManager::CStatsManager object * */ CStatsManager::CStatsManager() :_trameStatsVL() , _trameStatsPL() , _statsVL() , _statsPL() , _statsEssieu() , _modemSigFox(&uartSigfox, &pc) { } /** * @brief Destroy the CStatsManager::CStatsManager object * */ CStatsManager::~CStatsManager() {	

```
}

/***
 * @brief
 *
 * @param value
 */
void CStatsManager::setVitesseTronc(unsigned int value) // Défini depuis l'IHM
{
    _vitesseTronc = value;
}

/***
 * @brief
 *
 * @return unsigned int
 */
unsigned int CStatsManager::getVitesseTronc()
{
    return this -> _vitesseTronc;
}

/***
 * @brief
 *
 * @param value
 */
void CStatsManager::setNombreCapteur(unsigned int value)
{
    _nbrCapteur = value;
}

/***
 * @brief
 *
 * @return unsigned int
 */
unsigned int CStatsManager::getNombreCapteur()
{
    return this -> _nbrCapteur;
}

/***
 * @brief
 *
 * @param value
 */
void CStatsManager::setModeTest(bool value)
{
    _modeTest = value;
}
```

```

/***
 * @brief
 *
 * @return true
 * @return false
 */
bool CStatsManager::getModeTest()
{
    return this -> _modeTest;
}

/***
 * @brief
 *
 */
void CStatsManager::start()
{
    initModem();

    _statsPL.setVitesseTronc(_vitesseTronc);
    _statsVL.setVitesseTronc();

    if (&vehicule.get_typeVehicule == VL){
        _statsVL.add(&vehicule);
    }
    if (&vehicule.get_typeVehicule == PL){
        _statsPL.add(&vehicule);
    }

    _trameStatsVL.fabriquerTrame(&_statsVL);

    printf("Trame apres get : %s \r\n", _trameStatsVL.getTrame());

    char trame[25];
    for(int i = 0; i < 24; i++)
    {
        trame[i] = (char)_trameStatsVL.getTrame();
    }

    printf("Trame : ");
    for(int i = 0; i < 24; i++){
        printf("%c", trame[i]);
    }
    printf("\r\n");

    _modemSigFox.sendMessage(trame);

}
/***

```

E3 : CARRICART Hugo

```
* @brief
*
*/
void CStatsManager::initModem()
{
    // initialisations
    pc.set_baud(BAUDS_DEBUG);
    uartSigfox.set_baud(BAUDS_SIGFOX);
    pc.printf("Demarrage du programme Kolantr \r\n");
    pc.printf("Initialisation modem \r\n");

    // active le Modem
    _modemSigFox.initOK();
    // sélectionne le réseau
    _modemSigFox.setClePublique(SIGFOX_SIMUL);
}
```

CARRICART Hugo	CStatsVehicule.hpp	
/** * @file CStatsVehicule.hpp * @brief * @author Hugo Carricart * @date 05/05/2021 * @version 1,5 * **/	#ifndef CSTATSVEHICULE_HPP #define CSTATSVEHICULE_HPP #include "StrucTrameStatsSigFox.h" #include "mbed.h" #include "CVehiculeDetecte.hpp" class CStatsVehicule { public: /** * @brief Construct a new CStatsVehicule object * */ CStatsVehicule(); /** * @brief Construct a new CStatsVehicule object * * @param vitesseTronc * @param typeStat * @param nbrVehicule * @param nbrEssieu * @param VitesseMoy * @param VitesseMax * @param nbrVitesseOK * @param nbrVitesseLM20 * @param nbrVitesseLP20 * @param nbrVitesseLP30 * @param nbrVitesseLP40 * @param nbrVitesseLP50 */ CStatsVehicule(unsigned int vitesseTronc, unsigned int typeStat, unsigned int nbrVehicule, unsigned int nbrEssieu, unsigned int VitesseMoy, unsigned int VitesseMax,	

```

        unsigned int nbrVitesseOK,
        unsigned int nbrVitesseLM20,
        unsigned int nbrVitesseLP20,
        unsigned int nbrVitesseLP30,
        unsigned int nbrVitesseLP40,
        unsigned int nbrVitesseLP50);

/*
 * @brief Construct a new CStatsVehicule object
 */
* @param vitesseTronc
* @param typeStat
*/
CStatsVehicule(unsigned int vitesseTronc, unsigned int typeStat);

/*
 * @brief Destroy the CStatsVehicule object
 */
* /
~CStatsVehicule();

/*
 * @brief
 *
 * @param vehicule
 */
void add(CVehiculeDetecte *vehicule);

/*
 * @brief
 *
 * /
void reinit();

/*
 * @return unsigned int
 */
unsigned int getTypeStat();

/*
 * @brief Set the Type Stat object
 *
 * @param value
 */
void setTypeStat(unsigned int value);

/*
 * @brief Get the Nbr Vehicule object
 *
 * @return unsigned int
 */
```

```
*/
    unsigned int getNbrVehicule();

    /**
     * @brief Set the Nbr Vehicule object
     *
     * @param value
     */
    void setNbrVehicule(unsigned int value);

    /**
     * @brief Get the Nbr Essieu object
     *
     * @return unsigned int
     */
    unsigned int getNbrEssieu();

    /**
     * @brief Set the Nbr Essieu object
     *
     * @param value
     */
    void setNbrEssieu(unsigned int value);

    /**
     * @brief Get the Vitesse Moy object
     *
     * @return unsigned int
     */
    unsigned int getVitesseMoy();

    /**
     * @brief Set the Vitesse Moy object
     *
     * @param value
     */
    void setVitesseMoy(unsigned int value);

    /**
     * @brief Get the Vitesse Max object
     *
     * @return unsigned int
     */
    unsigned int getVitesseMax();

    /**
     * @brief Set the Vitesse Max object
     *
     * @param value
     */
    void setVitesseMax(unsigned int value);
```

```
 /**
 * @brief Get the Nbr Vitesse O K object
 *
 * @return unsigned int
 */
 unsigned int getNbrVitesseOK();

 /**
 * @brief Set the Nbr Vitesse O K object
 *
 * @param value
 */
 void setNbrVitesseOK(unsigned int value);

 /**
 * @brief Get the Nbr Vitesse LM20 object
 *
 * @return unsigned int
 */
 unsigned int getNbrVitesseLM20();

 /**
 * @brief Set the Nbr Vitesse L M20 object
 *
 * @param value
 */
 void setNbrVitesseLM20(unsigned int value);

 /**
 * @brief Get the Nbr Vitesse L P20 object
 *
 * @return unsigned int
 */
 unsigned int getNbrVitesseLP20();

 /**
 * @brief Set the Nbr Vitesse L P20 object
 *
 * @param value
 */
 void setNbrVitesseLP20(unsigned int value);

 /**
 * @brief Get the Nbr Vitesse L P30 object
 *
 * @return unsigned int
 */
 unsigned int getNbrVitesseLP30();

 /**
 * @brief Set the Nbr Vitesse L P30 object
 *
 */
```

```

* @param value
*/
void setNbrVitesseLP30(unsigned int value);

/**
* @brief Get the Nbr Vitesse L P40 object
*
* @return unsigned int
*/
unsigned int getNbrVitesseLP40();

/**
* @brief Set the Nbr Vitesse L P40 object
*
* @param value
*/
void setNbrVitesseLP40(unsigned int value);

/**
* @brief Get the Nbr Vitesse L P50 object
*
* @return unsigned int
*/
unsigned int getNbrVitesseLP50();

/**
* @brief Set the Nbr Vitesse L P50 object
*
* @param value
*/
void setNbrVitesseLP50(unsigned int value);

/**
* @brief Get the Vitesse Tronc object
*
* @return unsigned int
*/
unsigned int getVitesseTronc();

/**
* @brief Set the Vitesse Tronc object
*
*/
void setVitesseTronc(unsigned int value);

private:

unsigned int _vitesseTronc;
unsigned int _typeStat;      // 0 = Véhicule Léger / 1 = Véhicule Lourd (PL)
unsigned int _nbrVehicule;   // Total du nombre de véhicule compter
unsigned int _nbrEssieu;    // Total du nombre d'essieu compter

```

```
unsigned int _VitesseMoy; // Vitesse Moyenne
unsigned int _VitesseMax; // HIGH SCORE / Vitesse maximum atteint
unsigned int _nbrVitesseOK; // Nombre de véhicule à la vitesse ou moins de la voie de circulation
unsigned int _nbrLM20; // Nombre de véhicule au dessus (jusqu'à 20 km/h) de la limitation de la
voie de circulation
unsigned int _nbrLP20; // Nombre de véhicule au dessus des 20 km/h (jusqu'à 30 km/h) de la
limitation de la voie de circulation
unsigned int _nbrLP30; // Nombre de véhicule au dessus des 30 km/h (jusqu'à 40 km/h) de la
limitation de la voie de circulation
unsigned int _nbrLP40; // Nombre de véhicule au dessus des 40 km/h (jusqu'à 50 km/h) de la
limitation de la voie de circulation
unsigned int _nbrLP50; // Nombre de véhicule au dessus des 50 km/h et plus de la limitation de la
voie de circulation

};

#endif
```

CARRICART Hugo	CStatsVehicule.cpp	
/** * @file CStatsVehicule.cpp * @brief * @author Hugo Carricart * @date 05/05/2021 * @version 1.5 * **/	#include "CStatsVehicule.hpp" using namespace std; /** * @brief Construct a new CStatsVehicule::CStatsVehicule object * */ CStatsVehicule::CStatsVehicule() :_vitesseTronc(0) , _typeStat(0) , _nbrVehicule(0) , _nbrEssieu(0) , _VitesseMoy(0) , _VitesseMax(0) , _nbrVitesseOK(0) , _nbrLM20(0) , _nbrLP20(0) , _nbrLP30(0) , _nbrLP40(0) , _nbrLP50(0) { } CStatsVehicule::CStatsVehicule(unsigned int vitesseTronc, unsigned int typeStat, unsigned int nbrVehicule, unsigned int nbrEssieu, unsigned int VitesseMoy, unsigned int VitesseMax, unsigned int nbrVitesseOK, unsigned int nbrVitesseLM20, unsigned int nbrVitesseLP20, unsigned int nbrVitesseLP30, unsigned int nbrVitesseLP40, unsigned int nbrVitesseLP50) :_vitesseTronc(vitesseTronc) , _typeStat(0) , _nbrVehicule(nbrVehicule) , _nbrEssieu(nbrEssieu) , _VitesseMoy(VitesseMoy)	

```

, _VitesseMax(VitesseMax)
, _nbrVitesseOK(nbrVitesseOK)
, _nbrLM20(nbrVitesseLM20)
, _nbrLP20(nbrVitesseLP20)
, _nbrLP30(nbrVitesseLP30)
, _nbrLP40(nbrVitesseLP40)
, _nbrLP50(nbrVitesseLP50)
{
}

CStatsVehicule::CStatsVehicule(unsigned int vitesseTronc, unsigned int typeStat)
: _vitesseTronc(vitesseTronc)
, _typeStat(typeStat)
, _nbrVehicule(0)
, _nbrEssieu(0)
, _VitesseMoy(0)
, _VitesseMax(0)
, _nbrVitesseOK(0)
, _nbrLM20(0)
, _nbrLP20(0)
, _nbrLP30(0)
, _nbrLP40(0)
, _nbrLP50(0)
{
}

/***
* @brief Destroy the CStatsVehicule::CStatsVehicule object
*/
CStatsVehicule::~CStatsVehicule()
{
}

/***
* @brief
* @param vehicle
*/
void CStatsVehicule::add(CVehiculeDetecte *vehicule)
{
    // Ensuite ajout +nbrEssieu à _nbrEssieu
    _nbrEssieu = _nbrEssieu + vehicule->get_nombreEssieu();

    // Calcule vitesse moyenne
    // additionner toute les valeurs /diviser/ par le nombre de valeurs
    _VitesseMoy = ( (_VitesseMoy*_nbrVehicule) + vehicule->get_vitesse() ) / (_nbrVehicule + 1) ;
    //nombre d'itération ;

    // Ensuite ajout +1 à _nbrVehicule
}

```

E3 : CARRICART Hugo

```
_nbrVehicule = _nbrVehicule + 1;

// Calcule vitesse max
// valeur la plus élevée
if(_VitesseMax < vehicule->get_vitesse())
{
    _VitesseMax = vehicule->get_vitesse();
}

// Calcule nbrVitesseOK
// valeur <= vitesseTronc
if(vehicule->get_vitesse() <= _vitesseTronc)
{
    _nbrVitesseOK = _nbrVitesseOK + 1;
}

// Calcule nbrVitesseLM20
// vitesseTronc < valeur < vitesseTronc+20
if(_vitesseTronc < vehicule->get_vitesse() && vehicule->get_vitesse() <= _vitesseTronc + 20)
{
    _nbrLM20 = _nbrLM20 + 1;
}

// Calcule nbrVitesseLP20
// vitesseTronc + 20 <= valeur < vitesseTronc + 30
if(_vitesseTronc + 20 < vehicule->get_vitesse() && vehicule->get_vitesse() <= _vitesseTronc + 30)
{
    _nbrLP20 = _nbrLP20 + 1;
}

// Calcule nbrVitesseLP30
// vitesseTronc + 30 <= valeur < vitesseTronc + 40
if(_vitesseTronc + 30 <= vehicule->get_vitesse() && vehicule->get_vitesse() < _vitesseTronc + 40)
{
    _nbrLP30 = _nbrLP30 + 1;
}

// Calcule nbrVitesseLP40
// vitesseTronc + 40 <= valeur < vitesseTronc + 50
if(_vitesseTronc + 40 <= vehicule->get_vitesse() && vehicule->get_vitesse() < _vitesseTronc + 50)
{
    _nbrLP40 = _nbrLP40 + 1;
}

// Calcule nbrVitesseLP50
// vitesseTronc + 50 <= valeur
if(_vitesseTronc + 50 <= vehicule->get_vitesse())
{
    _nbrLP50 = _nbrLP50 + 1;
}
```

```
}

/***
 * @brief Réinitialise
 *
 */
void CStatsVehicule::reinit()
{
    _nbrVehicule = 0;
    _nbrEssieu = 0;
    _VitesseMoy = 0;
    _VitesseMax = 0;
    _nbrVitesseOK = 0;
    _nbrLM20 = 0;
    _nbrLP20 = 0;
    _nbrLP30 = 0;
    _nbrLP40 = 0;
    _nbrLP50 = 0;
}

/***
 * @brief
 *
 * @return unsigned int
 */
unsigned int CStatsVehicule::getVitesseTronc()
{
    return this-> _vitesseTronc;
}

/***
 * @brief
 *
 * @param value
 */
void CStatsVehicule::setVitesseTronc(unsigned int value)
{
    _vitesseTronc = value;
}

/***
 * @brief
 *
 * @return unsigned int
 */
unsigned int CStatsVehicule::getTypeStat()
{
    return this-> _typeStat;
}

/***
 * @brief
```

```
*  
* @param value  
*/  
void CStatsVehicule::setTypeStat(unsigned int value)  
{  
    _typeStat = value;  
}  
  
/**  
 * @brief  
 *  
 * @return unsigned int  
 */  
unsigned int CStatsVehicule::getNbrVehicule()  
{  
    return this->_nbrVehicule;  
}  
  
/**  
 * @brief  
 *  
 * @param value  
 */  
void CStatsVehicule::setNbrVehicule(unsigned int value)  
{  
    _nbrVehicule = value;  
}  
  
/**  
 * @brief  
 *  
 * @return unsigned int  
 */  
unsigned int CStatsVehicule::getNbrEssieu()  
{  
    return this->_nbrEssieu;  
}  
  
/**  
 * @brief  
 *  
 * @param value  
 */  
void CStatsVehicule::setNbrEssieu(unsigned int value)  
{  
    _nbrEssieu = value;  
}  
  
/**  
 * @brief  
 *  
 * @return unsigned int
```

```
/*
unsigned int CStatsVehicule::getVitesseMoy()
{
    return this->_VitesseMoy;
}

/***
* @brief
*
* @param value
*/
void CStatsVehicule::setVitesseMoy(unsigned int value)
{
    _VitesseMoy = value;
}

/***
* @brief
*
* @return unsigned int
*/
unsigned int CStatsVehicule::getVitesseMax()
{
    return this->_VitesseMax;
}

/***
* @brief
*
* @param value
*/
void CStatsVehicule::setVitesseMax(unsigned int value)
{
    _VitesseMax = value;
}

/***
* @brief
*
* @return unsigned int
*/
unsigned int CStatsVehicule::getNbrVitesseOK()
{
    return this->_nbrVitesseOK;
}

/***
* @brief
*
* @param value
*/
void CStatsVehicule::setNbrVitesseOK(unsigned int value)
```

```
{  
    _nbrVitesseOK = value;  
}  
  
/**  
 * @brief  
 *  
 * @return unsigned int  
 */  
unsigned int CStatsVehicule::getNbrVitesseLM20()  
{  
    return this->_nbrLM20;  
}  
  
/**  
 * @brief  
 *  
 * @param value  
 */  
void CStatsVehicule::setNbrVitesseLM20(unsigned int value)  
{  
    _nbrLM20 = value;  
}  
  
/**  
 * @brief  
 *  
 * @return unsigned int  
 */  
unsigned int CStatsVehicule::getNbrVitesseLP20()  
{  
    return this->_nbrLP20;  
}  
  
/**  
 * @brief  
 *  
 * @param value  
 */  
void CStatsVehicule::setNbrVitesseLP20(unsigned int value)  
{  
    _nbrLP20 = value;  
}  
  
/**  
 * @brief  
 *  
 * @return unsigned int  
 */  
unsigned int CStatsVehicule::getNbrVitesseLP30()  
{  
    return this->_nbrLP30;
```

```
}

/***
 * @brief
 *
 * @param value
 */
void CStatsVehicule::setNbrVitesseLP30(unsigned int value)
{
    _nbrLP30 = value;
}

/***
 * @brief
 *
 * @return unsigned int
 */
unsigned int CStatsVehicule::getNbrVitesseLP40()
{
    return this->_nbrLP40;
}

/***
 * @brief
 *
 * @param value
 */
void CStatsVehicule::setNbrVitesseLP40(unsigned int value)
{
    _nbrLP40 = value;
}

/***
 * @brief
 *
 * @return unsigned int
 */
unsigned int CStatsVehicule::getNbrVitesseLP50()
{
    return this->_nbrLP50;
}

/***
 * @brief
 *
 * @param value
 */
void CStatsVehicule::setNbrVitesseLP50(unsigned int value)
{
    _nbrLP50 = value;
}
```

CARRICART Hugo	CStatsEssieu.hpp	
/** * @file CStatsEssieu.hpp * @brief * @author Hugo Carricart * @date 04/02/2021 * @version 1.0 * **/	#ifndef CSTATSESSIEU_HPP #define CSTATSESSIEU_HPP #include "StrucTrameStatsSigFox.h" #include "mbed.h" class CStatsEssieu { public: /** * @brief Construct a new CStatsEssieu object * */ CStatsEssieu(); /** * @brief Construct a new CStatsEssieu object * * @param nbreEssieu */ CStatsEssieu(unsigned int nbreEssieu); /** * @brief Destroy the CStatsEssieu object * */ ~CStatsEssieu(); /** * @brief * * @param nbreEssieu */ void add(unsigned int nbreEssieu); /** * @brief * */ void reinit();	

```
/**  
 * @brief Set the Nombre Essieu object  
 *  
 * @param nbreEssieu  
 */  
void setNombreEssieu(unsigned int nbreEssieu);  
  
/**  
 * @brief Get the Nombre Essieu object  
 *  
 * @return unsigned int  
 */  
unsigned int getNombreEssieu();  
  
private:  
    unsigned int _nbreEssieu;  
  
};  
#endif
```

CARRICART Hugo	CStatsEssieu.cpp	
/** * @file CStatsEssieu.cpp * @brief * @author Hugo Carricart * @date 04/02/2021 * @version 1.0 * **/	#include "CStatsEssieu.hpp" /** * @brief Construct a new CStatsEssieu::CStatsEssieu object * */ CStatsEssieu::CStatsEssieu() : _nbreEssieu() { } /** * @brief Construct a new CStatsEssieu::CStatsEssieu object * * @param nbreEssieu */ CStatsEssieu::CStatsEssieu(unsigned int nbreEssieu) : _nbreEssieu(nbreEssieu) { } /** * @brief Destroy the CStatsEssieu::CStatsEssieu object * */ CStatsEssieu::~CStatsEssieu() { } /** * @brief * * @param nbreEssieu */ void CStatsEssieu::add(unsigned int nbreEssieu) { } /** * @brief Réinitialise	

```
*  
*/  
void CStatsEssieu::reinit()  
{  
    _nbreEssieu = 0 ;  
}  
  
/**  
 * @brief  
 *  
 * @param nombreEssieu  
 */  
void CStatsEssieu::setNombreEssieu(unsigned int nbreEssieu)  
{  
    _nbreEssieu = nbreEssieu;  
}  
  
/**  
 * @brief  
 *  
 */  
unsigned int CStatsEssieu::getNombreEssieu()  
{  
    return this->_nbreEssieu;  
}
```

CARRICART Hugo	StrucTrameStatsSigfox.h
<pre>/*** * @file StrucTrameStatsSigFox.h * @author Carricart Hugo * @brief * @version 0.1 * @date 2021-04-24 * * @copyright Copyright (c) 2021 * */ #ifndef STRUCTTRAMESTATSIGFOX_H #define STRUCTTRAMESTATSIGFOX_H /*** * @brief Trame Statisque ***/ struct StrucTrameStatsSigFox { unsigned int typeStat; // 0 = Véhicule Léger / 1 = Véhicule Lourd (PL) unsigned int nbrVehicule; // Total du nombre de véhicule compter unsigned int nbrEssieu; // Total du nombre d'essieu compter unsigned int VitesseMoy; // Vitesse Moyenne unsigned int VitesseMax; // HIGH SCORE / Vitesse maximum atteint unsigned int nbrVitesseOK; // Nombre de véhicule à la vitesse ou moins de la voie de circulation unsigned int nbrLM20; // Nombre de véhicule au dessus (jusqu'à 20 km/h) de la limitation de la voie de circulation unsigned int nbrLP20; // Nombre de véhicule au dessus des 20 km/h (jusqu'à 30 km/h) de la limitation de la voie de circulation unsigned int nbrLP30; // Nombre de véhicule au dessus des 30 km/h (jusqu'à 40 km/h) de la limitation de la voie de circulation unsigned int nbrLP40; // Nombre de véhicule au dessus des 40 km/h (jusqu'à 50 km/h) de la limitation de la voie de circulation unsigned int nbrLP50; // Nombre de véhicule au dessus des 50 km/h et plus de la limitation de la voie de circulation }; #endif</pre>	

CARRICART Hugo	PrintfBufferedSerial.hpp	
	<pre>/*** * @file PrintfBufferedSerial.hpp * @author Marc Lajus // Hugo Carricart * @brief * @version 0.1 * @date 2021-03-31 * * @copyright Copyright (c) 2021 * */ #pragma once #include "mbed.h" #include "BufferedSerial.h" /*** * @brief BufferedSerial enrichie * */ class PrintfBufferedSerial : public BufferedSerial { using BufferedSerial::BufferedSerial; // hérite des constructeurs public: int printf(const char* format, ...); char getc(); };</pre>	

CARRICART Hugo	PrintfBufferedSerial.cpp	
	<pre>/*** * @file PrintfBufferedSerial.cpp * @author Marc Lajus // Hugo Carricart * @brief * @version 0.1 * @date 2021-03-31 * * @copyright Copyright (c) 2021 */ #include "PrintfBufferedSerial.hpp" int PrintfBufferedSerial::printf(const char* format, ...) { const int _buf_size = 80; // _buf_size de BufferedSerial est private. La valeur 80 est a priori suffisante char buffer[_buf_size]; memset(buffer, 0, _buf_size); int r = 0; // on doit traiter la liste variable d'arguments de printf va_list arg; va_start(arg, format); r = vsprintf(buffer, format, arg); // this may not hit the heap but should alert the user anyways if(r > _buf_size) { error("%s %d buffer overwrite (max_buf_size: %d exceeded: %d)\r\n", __FILE__, __LINE__, _buf_size, r); va_end(arg); return 0; } va_end(arg); r = BufferedSerial::write(buffer, r); return r; } char PrintfBufferedSerial::getc(void) { while (!BufferedSerial::readable()) { ThisThread::yield(); } char c; BufferedSerial::read(&c, 1); return c; }</pre>	

CARRICART Hugo	main.cpp	
	<pre>/* * @file main.cpp * @author Carricart Hugo // Hervier Xavier // Jouandoudet Thibaut * @brief Kolantr * @version 1 * @date 2021-05-05 * * @copyright Copyright (c) 2021 * */ #include "mbed.h" #include "PrintfBufferedSerial.hpp" #include "CModemSigfox.hpp" #include "CStatsManager.hpp" CStatsManager Manager; int main() { Manager.start(); }</pre>	

TESTA Tony

Sitographie :

<https://os.mbed.com/forum/>

<https://os.mbed.com/docs/mbed-os/v6.11/introduction/index.html>

<https://stackoverflow.com>

Codes et programmes de test :

<https://github.com/TONE-I/KolantRProjet>

```

CVehicule.h
/***
 * \file cvehicule.h
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 17/03/2021
 * \details Classe conteneur des véhicules
 */
#ifndef CVEHICULE_H
#define CVEHICULE_H

#include <QObject>

class CVehicule : public QObject
{
    Q_OBJECT

public:
    explicit CVehicule(QObject *parent = nullptr);
    CVehicule(const CVehicule &vehicule, QObject *parent = nullptr);
    CVehicule(QString nom, QString type, int essieux, std::vector<float>
pEmpattement, QObject *parent = nullptr);
    ~CVehicule();

    QString getNom() const;
    void setNom(const QString &nom);

    QString getType() const;
    void setType(const QString &type);

    int getEssieux() const;
    void setEssieux(const int &essieux);

    std::vector<float> getEmpattement() const;
    void setEmpattement(const std::vector<float> &value);

    int getId() const;
    void setId(int value);

    QString verifierTypeVehicule(QString _type);

private :
    int pId;
    QString pNom;
    QString pType;
    int pEssieux;
    std::vector <float> pEmpattement;
};

#endif // CVEHICULE_H

```

```

CVehicule.cpp
/***
 * \file cvehicule.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 17/03/2021
 * \details Classe conteneur des véhicules
 */
#include "cvehicule.h"

/*
 * \brief CVehicule::CVehicule
 * \param parent
 * \details constructeur par défaut
 */
CVehicule::CVehicule(QObject *parent) : QObject(parent)
{
    //par défaut mettre null aux empatements
    std::vector<float> emp;
    for (int i = 0; 9 > i; i++) {
        emp.push_back(2.60);
    }
    pEmpattement = emp;
    pNom = "Vehicule_";
}
/*
 * \brief CVehicule::CVehicule
 * \param vehicule
 * \param parent
 * \details passage des paramètres
 */
CVehicule::CVehicule(const CVehicule &vehicule, QObject *parent) :
QObject(parent)
{
    pNom = vehicule.getNom();
    pType = vehicule.getType();
    pEssieux = vehicule.getEssieux();
    pEmpattement = vehicule.getEmpattement();
}
/*
 * \brief CVehicule::CVehicule
 * \param nom
 * \param type
 * \param essieux
 * \param empatement
 * \param parent
 * \details passage des paramètres
 */
CVehicule::CVehicule(QString nom, QString type, int essieux,
std::vector<float> empatement, QObject *parent) : QObject(parent)
{
    pNom = nom;
    pType = type;
    pEssieux = essieux;
    pEmpattement = empatement;
}
/*
 * \brief CVehicule::~CVehicule
 * \details destructeur
 */
CVehicule::~CVehicule()

```

```
{
}

/*!
 * \brief CVehicule::getNom
 * \return pNom
 * \details accesseur pNom
 */
QString CVehicule::getNom() const
{
    return pNom;
}
/*!
 * \brief CVehicule::setNom
 * \param nom
 * \details mutateur de pNom
 */
void CVehicule::setNom(const QString &nom)
{
    pNom = nom;
}
/*!
 * \brief CVehicule::getType
 * \return pType
 * \details accesseur de pType
 */
QString CVehicule::getType() const
{
    return pType;
}
/*!
 * \brief CVehicule::setType
 * \param type
 * \details mutateur de pType
 */
void CVehicule::setType(const QString &type)
{
    pType = type;
}
/*!
 * \brief CVehicule::getEssieux
 * \return pEssieux
 * \details accesseur de pEssieux
 */
int CVehicule::getEssieux() const
{
    return pEssieux;
}
/*!
 * \brief CVehicule::setEssieux
 * \param essieux
 * \details mutateur de pEssieux
 */
void CVehicule::setEssieux(const int &essieux)
{
    pEssieux = essieux;
}
/*!
 * \brief CVehicule::getEmpatement
 * \return pEmpatement
 * \details accesseur de pEmpatement
 */
std::vector<float> CVehicule::getEmpatement() const
{
```

```

        return pEmpatement;
    }
/*!
 * \brief CVehicule::setEmpatement
 * \param value
 * \details mutateur de pEmpatement
 */
void CVehicule::setEmpatement(const std::vector<float> &value)
{
    pEmpatement = value;
}
/*!
 * \brief CVehicule::getId
 * \return pId
 * \details accesseur de pId
 */
int CVehicule::getId() const
{
    return pId;
}
/*!
 * \brief CVehicule::setId
 * \param value
 * \details mutateur de pId
 */
void CVehicule::setId(int value)
{
    pId = value;
}
/*!
 * \brief CVehicule::verifierTypeVehicule
 * \param _type
 * \return _type
 * \details verifie avec le nom recuperé de la base de donnée si c'est un
poids lourd ou vehicule leger
 */
QString CVehicule::verifierTypeVehicule(QString _type)
{
    if (_type == "vl")
        _type = "Véhicule léger";

    else
        _type = "Poids lourd";

    return _type;
}

```

```
CPassage.h
/**
 * \file cpassage.h
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/04/2021
 * \details Classe conteneur des passages
 */
#ifndef CPASSAGE_H
#define CPASSAGE_H

#include <QObject>
#include "cvehicule.h"

class CPassage : public QObject
{
    Q_OBJECT

public:
    explicit CPassage(QObject *parent = nullptr);
    CPassage(int vitesse, int interVehicule, CVehicule &vehicule, QObject
*parent = nullptr);
    ~CPassage();

    void addVehicule(CVehicule &Vehicule);
    void delVehicule();

    CVehicule *getVehicule() const;
    void setVehicule(CVehicule *value);

    int getVitesse() const;
    void setVitesse(int vitesse);

    int getInterVehicule() const;
    void setInterVehicule(int interVehicule);

    unsigned int getPassageId() const;
    void setPassageId(unsigned int Id);

private:
    int pVitesse;
    int pInterVehicule;
    unsigned int pId;
    CVehicule *pVehicule = nullptr ;
};

#endif // CPASSAGE_H
```

CPassage.cpp

```
/***
 * \file cpassage.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/04/2021
 * \details Classe conteneur des passages
 */
#include "cpassage.h"
*/
/*!
 * \brief CPassage::CPassage
 * \param parent
 * \details constructeur par defaut
 */
CPassage::CPassage(QObject *parent) : QObject(parent) {}
/*!
 * \brief CPassage::CPassage
 * \param vitesse
 * \param interVehicule
 * \param vehicule
 * \param parent
 * \details passage des parametres
 */
CPassage::CPassage(int vitesse, int interVehicule, CVehicule &vehicule,
QObject *parent)
: QObject(parent)
{
    pVitesse = vitesse;
    pInterVehicule = interVehicule;
    pVehicule = new CVehicule(vehicule);
}
/*!
 * \brief CPassage::~CPassage
 * \details destructureur
 */
CPassage::~CPassage()
{
}
/*!
 * \brief CPassage::addVehicule
 * \param Vehicule
 * \details ajoute l'objet vehicule
 */
void CPassage::addVehicule(CVehicule &Vehicule)
{
    if(pVehicule != nullptr)
        delete pVehicule;
    pVehicule = new CVehicule(Vehicule);
}
/*!
 * \brief CPassage::delVehicule
 * \details detruit l'objet vehicule
 */
void CPassage::delVehicule()
{
    if (pVehicule != nullptr)
        delete pVehicule;
}
/*!
```

```

* \brief CPassage::getVitesse
* \return pVitesse
* \details accesseur pVitesse
*/
int CPassage::getVitesse() const
{
    return pVitesse;
}
/*!
* \brief CPassage::setVitesse
* \param vitesse
* \details mutateur pVitesse
*/
void CPassage::setVitesse(int vitesse)
{
    pVitesse = vitesse;
}
/*!
* \brief CPassage::getInterVehicule
* \return pInterVehicule
* \details accesseur pInterVehicule
*/
int CPassage::getInterVehicule() const
{
    return pInterVehicule;
}
/*!
* \brief CPassage::setInterVehicule
* \param interVehicule
* \details mutateur pInterVehicule
*/
void CPassage::setInterVehicule(int interVehicule)
{
    pInterVehicule = interVehicule;
}
/*!
* \brief CPassage::getPassageId
* \return pId
* \details accesseur pId
*/
unsigned int CPassage::getPassageId() const
{
    return pId;
}
/*!
* \brief CPassage::setPassageId
* \param value
* \details mutateur pId
*/
void CPassage::setPassageId(unsigned int value)
{
    pId = value;
}
/*!
* \brief CPassage::getVehicule
* \return pVehicule
* \details accesseur pVehicule
*/
CVehicule *CPassage::getVehicule() const
{
    return pVehicule;
}
/*!
* \brief CPassage::setVehicule

```

```

* \param value
* mutateur pVehicule
*/
void CPassage::setVehicule(CVehicule *value)
{
    pVehicule = value;
}

```

cscenario.h

```

/***
* \file cscenario.h
* \brief Système de simulation Qt
* \author Arthur LABARRE
* \copyright Copyright (c) 2021 Arthur LABARRE
* \version 1.0
* \date 17/03/2021
* \details Classe conteneur des scenarios, et remplit classe cstats
*/
#ifndef CSCENARIO_H
#define CSCENARIO_H

#include <QObject>
#include "cpassage.h"
#include "cstats.h"

class CScenario : public QObject
{
    Q_OBJECT
public:
    explicit CScenario(QObject *parent = nullptr);
    CScenario(QString nomScenario, QObject *parent = nullptr);
    ~CSenario();

    void addVectPassage(CPassage *passage);

    std::vector<CPassage *> getVectPassage() const;

    QString getNom() const;
    void setNom(const QString &value);

    int getId() const;
    void setId(int value);

    CStats *calculerStatsPL();
    CStats *calculerStatsVL();

private:
    std::vector<CPassage *> pVectPassage ;
    QString pNom;
    unsigned int pId;

    CStats *pStatsPL , *pStatsVL;
};

#endif // CSCENARIO_H

```

CScenario.cpp

```
/***
 * \file cscenario.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 17/03/2021
 * \details Classe conteneur des scenarios, et remplit classe cstats
 */
#include "cscenario.h"

/*!
 * \brief CScenario::CSenario
 * \param parent
 * \details constructeur par defaut
 */
CSenario::CSenario(QObject *parent) : QObject(parent) {
    pNom = "Scenario_";
    pStatsPL= new CStats();
    pStatsVL= new CStats();
}
/*!
 * \brief CScenario::CSenario
 * \param nomScenario
 * \param parent
 * \details passage des parametres
 */
CSenario::CSenario(QString nomScenario, QObject *parent ) : QObject(parent) {
    pNom = nomScenario ;
}
/*!
 * \brief CScenario::~CSenario
 * \details destructeur
 */
CSenario::~CSenario()
{
}

/*!
 * \brief CScenario::addVectPassage
 * \param passage
 * \details ajout vecteur de l'objet passage
 */
void CSenario::addVectPassage(CPassage *passage)
{
    pVectPassage.push_back(passage);
}

/*!
 * \brief CScenario::getVectPassage
 * \return pVectPassage
 * \details accesseur du vecteur de l'object passage
 */
std::vector<CPassage *> CSenario::getVectPassage() const
{
    return pVectPassage ;
}

/*!
 * \brief CScenario::getNom
 */
```

```

* \return pNom
* \details accesseur de pNom
*/
QString CScenario::getNom() const
{
    return pNom;
}
/*!
* \brief CScenario::setNom
* \param value
* \details mutateur de pNom
*/
void CScenario::setNom(const QString &value)
{
    pNom = value;
}
/*!
* \brief CScenario::getId
* \return pId
* \details accesseur de pId
*/
int CScenario::getId() const
{
    return pId;
}
/*!
* \brief CScenario::setId
* \param value
* \details mutateur de pId
*/
void CScenario::setId(int value)
{
    pId = value;
}
/*!
* \brief CScenario::calculerStatsVL
* \return pStatsVL
* \details calcule statistique vehicule leger
*/
CStats * CScenario::calculerStatsVL()
{
    pStatsVL->init();
    int nbreVehicule = 0;
    int vitesseMoy = 0;
    int duree = 0;
    int vitesseMax = 0;
    int vitesseTotal = 0;
    for (long unsigned i = 0; i < pVectPassage.size(); i++) {
        if (pVectPassage[i]->getVehicule()->getType() == "vl") {
            nbreVehicule += 1;
            duree = (int unsigned) ((pVectPassage[i]->getInterVehicule() + duree));
            vitesseTotal += (int unsigned) (pVectPassage[i]->getVitesse());
            if (vitesseMax < pVectPassage[i]->getVitesse())
                vitesseMax = (int unsigned) (pVectPassage[i]->getVitesse());
        }
    }
    if (nbreVehicule > 0)
        vitesseMoy = vitesseTotal / nbreVehicule;

    pStatsVL->setNbreVehicule(nbreVehicule);
    pStatsVL->setVitesseMoy(vitesseMoy);
    pStatsVL->setDuree(duree);
    pStatsVL->setVitesseMax(vitesseMax);
    return pStatsVL;
}

```

```
}

/*!
 * \brief CScenario::calculerStatsPL
 * \return pStatsPL
 * \details calcule statistique poids lourd
 */
CStats *CSenario::calculerStatsPL()
{
    pStatsPL->init();
    int nbreVehicule = 0;
    int vitesseTotal = 0;
    int duree = 0;
    int vitesseMax = 0;
    int vitesseMoy = 0;
    for (long unsigned i = 0; i < pVectPassage.size(); i++) {
        if (pVectPassage[i]->getVehicule()->getType() == "pl") {
            nbreVehicule += 1;
            duree = (int) ((pVectPassage[i]->getInterVehicule() + duree));
            vitesseTotal += (int) (pVectPassage[i]->getVitesse());
            if (vitesseMax < pVectPassage[i]->getVitesse())
                vitesseMax = (int) (pVectPassage[i]->getVitesse());
        }
    }
    if (nbreVehicule > 0)
        vitesseMoy = vitesseTotal / nbreVehicule;

    pStatsPL->setNbreVehicule(nbreVehicule);
    pStatsPL->setVitesseMoy(vitesseMoy);
    pStatsPL->setDuree(duree);
    pStatsPL->setVitesseMax(vitesseMax);

    return pStatsPL;
}
```

CStats.h

```
/***
 * \file cstats.h
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 17/03/2021
 * \details Classe conteneur des stats
 */
#ifndef CSTATS_H
#define CSTATS_H

#include <QObject>

class CStats: public QObject
{
    Q_OBJECT

public:
    explicit CStats(QObject *parent = nullptr);
    CStats(int nbreVehicule, int vitesseMoy, int duree, QObject *parent = nullptr);
    ~CStats();

    int getNbreVehicule() const;
    void setNbreVehicule(int value);

    int getVitesseMoy() const;
    void setVitesseMoy(int value);

    int getDuree() const;
    void setDuree(int value);

    int getVitesseMax() const;
    void setVitesseMax(int value);

    void init();

private:
    int pDuree = 0;
    int pNbreVehicule = 0;
    int pVitesseMoy = 0;
    int pVitesseMax = 0;

};

#endif // CSTATS_H
```

CStats.cpp

```
/***
 * \file stats.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 17/03/2021
 * \details Classe conteneur des stats
 */
#include "cstats.h"
*/
/*!
 * \brief CStats::CStats
 * \param parent
 * \details constructeur par defaut
 */
CStats::CStats(QObject *parent) : QObject(parent)
{
}
/*
 * \brief CStats::CStats
 * \param nbreVehicule
 * \param vitesseMoy
 * \param duree
 * \param parent
 * \details passage des parametres
 */
CStats::CStats(int nbreVehicule, int vitesseMoy, int duree, QObject
*parent) : QObject(parent)
{
    pNbreVehicule=nbreVehicule;
    pVitesseMoy=vitesseMoy;
    pDuree=duree;
}
/*!
 * \brief CStats::~CStats
 * \details destructeur
 */
CStats::~CStats()
{
}

/*
 * \brief CStats::getDuree
 * \return pDuree
 * \details accesseur pDuree
 */
int CStats::getDuree() const
{
    return pDuree;
}
/*!
 * \brief CStats::setDuree
 * \param value
 * \details mutateur de pDuree
 */
void CStats::setDuree(int value)
{
    pDuree = value;
}
/*
 * \brief CStats::getVitesseMax
 */
```

```

* \return pVitesseMax
* \details accesseur de pVitesseMax
*/
int CStats::getVitesseMax() const
{
    return pVitesseMax;
}
/*!
* \brief CStats::setVitesseMax
* \param value
* \details mutateur de pVitesseMax
*/
void CStats::setVitesseMax(int value)
{
    pVitesseMax = value;
}
/*!
* \brief CStats::init
* \details initialise les attributs à 0
*/
void CStats::init()
{
    pNbreVehicule = 0;
    pDuree = 0;
    pVitesseMoy = 0;
    pVitesseMax = 0;
}
/*!
* \brief CStats::getNbreVehicule
* \return pNbreVehicule
* \details accesseur de pNbreVehicule
*/
int CStats::getNbreVehicule() const
{
    return pNbreVehicule;
}
/*!
* \brief CStats::setNbreVehicule
* \param value
* \details mutateur de pNbreVehicule
*/
void CStats::setNbreVehicule(int value)
{
    pNbreVehicule = value;
}
/*!
* \brief CStats::getVitesseMoy
* \return pVitesseMoy
* \details accesseur de pVitesseMoy
*/
int CStats::getVitesseMoy() const
{
    return pVitesseMoy;
}
/*!
* \brief CStats::setVitesseMoy
* \param value
* \details mutateur de pVitesseMoy
*/
void CStats::setVitesseMoy(int value)
{
    pVitesseMoy = value;
}

```

CVehiculeManager.h

```
/** 
 * \file cvehiculemanager.h
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date \date 10/02/2021
 * \details Classe permettant d'interroger ou d'envoyer des requêtes dans
la base de données en rapport avec les véhicules
*/
#ifndef CVEHICULEMANAGER_H
#define CVEHICULEMANAGER_H

#include <QObject>
#include <QtSql>
#include "cvehicule.h"
#include "QDebug"
#include "QString"

class CVehiculemanager : public QObject
{
    Q_OBJECT
public:
    CVehiculemanager(QSqlDatabase *dataBase, QObject *parent = nullptr);
    std::vector<CVehicule *> listeVehicule();
    CVehicule *getVehiculeByNom(QString nom);
    bool delVehicule(QString nom);
    bool upDate(QString nom, const CVehicule &vehicule);
    bool addVehicule(const CVehicule &vehicule);
    CVehicule * getVehiculeByPassageId(unsigned vehiculeId);

private:
    QSqlDatabase *pDataBase = nullptr ;
};

#endif // CVEHICULEMANAGER_H
```

cvehiculemanager.cpp

```
/***
 * \file cvehiculemanager.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date \date 10/02/2021
 * \details Classe permettant d'interroger ou d'envoyer des requêtes dans la base de données en rapport avec les véhicules
 */
#include "cvehiculemanager.h"
/*!
 * \brief CVehicleManager::CVehicleManager
 * \param DataBase
 * \param parent
 * \details Constructeur par défaut
 */
CVehicleManager::CVehicleManager(QSqlDatabase *DataBase, QObject *parent) :
QObject(parent)
{
    DataBase = DataBase;
}
/*!
 * \brief CVehicleManager::listeVehicule
 * \return cvehicule
 * \details Listes des objets CVehicles
 */
std::vector<CVehicle*> CVehicleManager::listeVehicule()
{
    std::vector<CVehicle *> vectVehicule;
    if (!DataBase->open()) {
        qDebug() << "CVehicleManager::ListeVehicule - problème ouverture base";
    } else {
        QSqlQuery *sqlQuery = new QSqlQuery(DataBase->database());
        sqlQuery->prepare("SELECT * FROM simulation_vehicule");
        sqlQuery->exec();

        CVehicle *vehicule = nullptr;
        while (sqlQuery->next()) {
            //Création d'un objet véhicule
            vehicule = new CVehicle();
            vehicule->setNom(sqlQuery->value("v_nom").toString());
            //Ajout de l'objet créé dans vectBadges
            vectVehicule.push_back(vehicule);
        }
    }
    return vectVehicule;
}
/*!
 * \brief CVehicleManager::getVehiculeByNom
 * \param nom
 * \return véhicule
 * \details Revoie les infos sur la voiture à partir du nom
 */
CVehicle *CVehicleManager::getVehiculeByNom(QString nom)
{
    if (!DataBase->open()) {
        qDebug() << "CVehicleManager::getVehiculeByNom - problème ouverture base";
        return nullptr;
    } else {
        QSqlQuery *sqlQuery = new QSqlQuery(DataBase->database());
```

```

        sqlQuery->prepare("SELECT * FROM simulation_vehicule WHERE v_nom = :nom
");
        sqlQuery->bindValue(":nom", nom);
        sqlQuery->exec();

        std::vector<float> empatement ;
        CVehicule *vehicule = nullptr;
        while (sqlQuery->next()) {
            vehicule = new CVehicule();
            vehicule->setId(sqlQuery->value("vehicule_id").toInt());
            vehicule->setNom(sqlQuery->value("v_nom").toString());
            vehicule->setType(sqlQuery->value("v_type").toString());
            vehicule->setEssieux(sqlQuery->value("v_essieux").toInt());
            empatement.push_back(sqlQuery->value("v_empatement1").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement2").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement3").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement4").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement5").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement6").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement7").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement8").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement9").toFloat());
            vehicule->setEmpatement(empatement);
        }
        return vehicule;
    }

}

/*!
* \brief CVehiculemanager::delVehicule
* \param ret
* \return delVehicule
* \details Supprime un vehicule dans la table simulation vehicule
*/
bool CVehiculemanager::delVehicule(QString nom)
{
    bool ret = false;
    if (!pDataBase->open())
        qDebug() << "CVehiculemanager::delVehicule - problème ouverture base";
        return false;
    } else {
        QSqlQuery *sqlQuery = new QSqlQuery(pDataBase->database());
        sqlQuery->prepare("DELETE FROM simulation_vehicule WHERE v_nom = :nom ");
        sqlQuery->bindValue(":nom", nom);
        ret = sqlQuery->exec();
        return ret;
    }
}
/*!
* \brief CVehiculemanager::upDate
* \param nom
* \param vehicule
* \return ret
* \details maj vehicule dans la table simulation vehicule
*/
bool CVehiculemanager::upDate(QString nom, const CVehicule &vehicule)
{
    bool ret = false ;
    if ( ! pDataBase->open() )
    {
        qDebug() << "CVehiculemanager::upDate - problème ouverture base" ;
        return false ;
    }
    else

```

```

    {
        QSqlQuery *sqlQuery = new QSqlQuery (pDataBase->database());
        sqlQuery->prepare(
            "UPDATE simulation_vehicule SET v_nom = :nom, v_type = :type,
v_essieux = :essieux, "
            "v_empatement1 = :empatement1, v_empatement2 = :empatement2,
v_empatement3 = "
            ":empatement3, v_empatement4 = :empatement4, v_empatement5 =
:empatement5, "
            "v_empatement6 = :empatement6, v_empatement7 = :empatement7,
v_empatement8 = "
            ":empatement8, v_empatement9 = :empatement9 WHERE v_nom = :nomAmodif
");
        sqlQuery->bindValue(":nomAmodif", nom);
        sqlQuery->bindValue(":nom", vehicule.getNom());
        sqlQuery->bindValue(":type", vehicule.getType());
        sqlQuery->bindValue(":essieux", vehicule.getEssieux());
        sqlQuery->bindValue(":empatement1", vehicule.getEmpatement() [0]);
        sqlQuery->bindValue(":empatement2", vehicule.getEmpatement() [1]);
        sqlQuery->bindValue(":empatement3", vehicule.getEmpatement() [2]);
        sqlQuery->bindValue(":empatement4", vehicule.getEmpatement() [3]);
        sqlQuery->bindValue(":empatement5", vehicule.getEmpatement() [4]);
        sqlQuery->bindValue(":empatement6", vehicule.getEmpatement() [5]);
        sqlQuery->bindValue(":empatement7", vehicule.getEmpatement() [6]);
        sqlQuery->bindValue(":empatement8", vehicule.getEmpatement() [7]);
        sqlQuery->bindValue(":empatement9", vehicule.getEmpatement() [8]);
        ret = sqlQuery->exec() ;
        return ret ;
    }
}
/*!
* \brief CVehiculemanager::addVehicule
* \param vehicule
* \return sqlQuery->isValid()
* \details ajoute une voiture dans la base de donnée
*/
bool CVehiculemanager::addVehicule(const CVehicule &vehicule)
{
    if (! pDataBase->open() )
    {
        qDebug() << "CVehiculemanager::addVehicule - problème ouverture base"
;
        return false ;
    }
    else
    {
        QSqlQuery *sqlQuery = new QSqlQuery (pDataBase->database());

        sqlQuery->prepare(
            "INSERT "
"INTO`simulation_vehicule`(`v_nom`, `v_type`, `v_essieux`, `v_empatement1`, `v_"
"empatement2`, `v_empatement3`, `v_empatement4`, `v_empatement5`, `v_empatement6`, `v_"
"empatement7`, `v_empatement8`, `v_empatement9`)VALUES "
"(:nom,:type,:essieux,:empatement1,:empatement2,:empatement3,:empatement4,:"
":empatement5,:empatement6,:empatement7,:empatement8,:empatement9)";

        sqlQuery->bindValue(":nom", vehicule.getNom());
        sqlQuery->bindValue(":type", vehicule.getType());
        sqlQuery->bindValue(":essieux", vehicule.getEssieux());
        sqlQuery->bindValue(":empatement1", vehicule.getEmpatement() [0]);
        sqlQuery->bindValue(":empatement2", vehicule.getEmpatement() [1]);

```

```

        sqlQuery->bindValue(":empatement3", vehicule.getEmpatement() [2]);
        sqlQuery->bindValue(":empatement4", vehicule.getEmpatement() [3]);
        sqlQuery->bindValue(":empatement5", vehicule.getEmpatement() [4]);
        sqlQuery->bindValue(":empatement6", vehicule.getEmpatement() [5]);
        sqlQuery->bindValue(":empatement7", vehicule.getEmpatement() [6]);
        sqlQuery->bindValue(":empatement8", vehicule.getEmpatement() [7]);
        sqlQuery->bindValue(":empatement9", vehicule.getEmpatement() [8]);

        sqlQuery->exec() ;
        return sqlQuery->isValid();
    }

}

/*!
* \brief CVehiculemanager::getVehiculeByClientParkingId
* \param simulationId
* \details renvoie un vehicule d'un id_simulation
* \return vehicule
*/
CVehicule *CVehiculemanager::getVehiculeByPassageId(unsigned vehiculeId)
{
    if (! DataBase->open() )
    {
        qDebug() << "CVehiculemanager::getVehiculeBySimulation - problème
ouverture base" ;
        return nullptr ;
    }
    else
    {
        QSqlQuery *sqlQuery = new QSqlQuery(DataBase->database());
        sqlQuery->prepare("SELECT * FROM simulation_vehicule WHERE vehicule_id =
:vehiculeId");
        sqlQuery->bindValue(":vehiculeId", vehiculeId);
        sqlQuery->exec();

        std::vector<float> empatement;
        while (sqlQuery->next()) {
            CVehicule *vehicule = nullptr ;
            vehicule = new CVehicule();
            vehicule->setNom(sqlQuery->value("v_nom").toString());
            vehicule->setType(sqlQuery->value("v_type").toString());
            vehicule->setEssieux(sqlQuery->value("v_essieux").toInt());
            empatement.push_back(sqlQuery->value("v_empatement1").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement2").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement3").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement4").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement5").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement6").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement7").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement8").toFloat());
            empatement.push_back(sqlQuery->value("v_empatement9").toFloat());
            vehicule->setEmpatement(empatement);
            return vehicule ;
        }
        return nullptr ;
    }
}

```

cpassagemanager.h

```

/***
 * \file cpassagemanager.h
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 17/03/2021
 * \details Classe permettant d'interroger ou d'envoyer des requêtes dans
la base de données en rapport avec les passages
*/
#ifndef CPASSAGEMANAGER_H
#define CPASSAGEMANAGER_H

#include <QObject>
#include <QtSql>
#include "cpassage.h"
#include "cvehiculemanager.h"
#include "QString"

class CPassageManager : public QObject
{
    Q_OBJECT
public:
    CPassageManager(QSqlDatabase *dataBase, QObject *parent = nullptr);
    ~CPassageManager();
    std::vector<CPassage *> listePassage();
    std::vector<CPassage *> getPassageByScenarioId(unsigned scenarioId);
    bool delPassage(int id);
    bool addPassage(CPassage &passage, unsigned scenarioId);

private:
    QSqlDatabase *pDataBase = nullptr ;
    CVehiculemanager *pVehiculeManager;
};

#endif // CPASSAGEMANAGER_H

```

cpassagemanager.cpp

```

/***
 * \file cpassagemanager.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 17/03/2021
 * \details Classe permettant d'interroger ou d'envoyer des requêtes dans
la base de données en rapport avec les passages
*/
#include "cpassagemanager.h"
#include "qdebug.h"
/*!
 * \brief CPassageManager::CPassageManager
 * \param DataBase
 * \param parent
 * \details constructeur par défaut
 */
CPassageManager::CPassageManager(QSqlDatabase *DataBase, QObject *parent)
:QObject(parent)
{
    DataBase = DataBase;
    pVehiculeManager = new CVehicleManager(DataBase);
}
/*!
 * \brief CPassageManager::~CPassageManager
 * \details destructeur + détruit pVehiculeManager
 */
CPassageManager::~CPassageManager()
{
    delete pVehiculeManager;
}

/*!
 * \brief CSimulationPassageManager::delPassage
 * \param id
 * \return ret
 * \details Supprime un passage dans la table simulation_passage
 */
bool CPassageManager::delPassage(int id)
{
    bool ret = false;
    if (!DataBase->open()) {
        qDebug() << "CSimulationPassageManager::delPassage - problème
ouverture base";
        return false;
    } else {
        QSqlQuery *sqlQuery = new QSqlQuery(DataBase->database());
        sqlQuery->prepare("DELETE FROM simulation_passage WHERE p_id =
:id");
        sqlQuery->bindValue(":id", id);
        ret = sqlQuery->exec();
        return ret;
    }
}
/*!
 * \brief CSimulationPassageManager::addPassage
 * \param passage
 * \return ret
 * \details ajoute un passage dans la table simulation_passage
 */
bool CPassageManager::addPassage(CPassage &passage, unsigned scenarioId)
{

```

```

CVehicule *vehicule = new CVehicule();
if (!pDataBase->open()) {
    qDebug() << "CSimulationPassageManager::addPassage - problème
ouverture base";
    return false;
} else {
    //récuperation de l'id du véhicule
    vehicule = pVehiculeManager-
>getVehiculeByNom(passage.getVehicule()->getNom());
    passage.setVehicule(vehicule);
    //ajout du passage
    QSqlQuery *sqlQuery = new QSqlQuery(pDataBase->database());
    sqlQuery->prepare("INSERT
INTO`simulation_passage`(`p_vitesse`, `p_inter-
vehicule`, `id_scenario`, `id_vehicule`) VALUES(:vitesse, :interVehicule,
:scenario, :vehicule)");
    sqlQuery->bindValue(":vitesse", passage.getVitesse());
    sqlQuery->bindValue(":interVehicule",
passage.getInterVehicule());
    sqlQuery->bindValue(":scenario", scenarioId);
    sqlQuery->bindValue(":vehicule", passage.getVehicule()->getId());
    sqlQuery->exec();
    return sqlQuery->isValid();
}
delete vehicule;
}

/*!*
 * \brief CSimulationPassageManager::getSimulationByScenarioId
 * \param scenarioId
 * \return vectPassage
 * \details renvoie un véhicule d'un id_scenario
 */
std::vector<CPassage *> CPassageManager::getPassageByScenarioId(unsigned
scenarioId)
{
    std::vector<CPassage *> vectPassage;
    if (!pDataBase->open()) {
        qDebug() << "CSimulationPassageManager::getVehiculeBySimulationId
- problème ouverture base";
    } else {
        QSqlQuery *sqlQuery = new QSqlQuery(pDataBase->database());
        sqlQuery->prepare("SELECT * FROM simulation_passage WHERE
id_scenario = :scenarioId");
        sqlQuery->bindValue(":scenarioId", scenarioId);
        sqlQuery->exec();
        CPassage *passage = nullptr;
        CVehicule *Vehicule = nullptr;

        while (sqlQuery->next()) {
            //Création d'un objet passage et véhicule
            passage = new CPassage();
            Vehicule = new CVehicule();

            Vehicule->setId(sqlQuery->value("id_vehicule").toInt());
            passage->setVehicule(Vehicule);
            passage->setPassageId(sqlQuery->value("p_id").toInt());
            passage->setVitesse(sqlQuery->value("p_vitesse").toInt());
            passage->setInterVehicule(sqlQuery->value("p_inter-
vehicule").toInt());
            //Ajout de l'objet créé dans vectPassage
            vectPassage.push_back(passage);
        }
        //recuperation véhicule
    }
}

```

```

        int i = 0;
        for (CPassage *Passage : vectPassage) {
            Vehicule = pVehiculeManager-
>getVehiculeByPassageId(vectPassage[i]->getVehicule()->getId());
            if(Vehicule!=nullptr)
                Passage->addVehicule(*Vehicule);
            else if (Vehicule == nullptr)
                delPassage(Passage->getPassageId());
            i += 1;
        }
    }
    return vectPassage;
}

```

cscenarioManager.h

```

/*
 * \file cscenarioManager.h
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 17/03/2021
 * \details Classe permettant d'interroger ou d'envoyer des requetes dans
la base de données, de remplir le json et l'envoyer en liason serie
*/
#ifndef CSCENARIOMANAGER_H
#define CSCENARIOMANAGER_H

#include <QObject>
#include <QObject>
#include <QtSql>
#include <QtSerialPort/QSerialPort>
#include "QDebug"
#include "QString"
#include "cscenario.h"
#include "cpassagemanager.h"

class CScenarioManager: public QObject
{
    Q_OBJECT
public:
    CScenarioManager(QSqlDatabase *dataBase, QObject *parent = nullptr);
    ~CScenarioManager();
    std::vector<CScenario *> listeScenarios();
    bool addScenario(const CScenario &scenario);
    bool delScenario(QString nom);
    CScenario *getScenarioByNom(QString nom);

    bool creeJson(const QString emplacement, CScenario *scenario);
    int transfertJson(const QString emplacement, const QString nomPort);

private:
    QSqlDatabase *pDataBase = nullptr ;
    CPassageManager *pPassageManager;

    QSerialPort serialPort;

};
#endif // CSCENARIOMANAGER_H

```

cscenarioManager.cpp

```
/** 
 * \file cscenarioManager.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 17/03/2021
 * \details Classe permettant d'interroger ou d'envoyer des requêtes dans la base de données, de remplir le json et l'envoyer en liaison série
 */
#include "cscenariomanager.h"
/*!
 * \brief CScenarioManager::CScenarioManager
 * \param DataBase
 * \param parent
 * \details Constructeur par défaut
 */
CScenarioManager::CScenarioManager(QSqlDatabase *DataBase, QObject *parent)
{
    DataBase = DataBase ;
    pPassageManager = new CPassageManager(DataBase);
}
/*!
 * \brief CScenarioManager::~CScenarioManager
 * \details Destructeur + détruit pPassageManager
 */
CScenarioManager::~CScenarioManager()
{
    delete pPassageManager;
}
/*!
 * \brief CScenarioManager::listeScenarios
 * \return vectScenarios
 * \details Listes des objets CScenario
 */
std::vector<CScenario *> CScenarioManager::listeScenarios()
{
    std::vector<CScenario *> vectScenarios;
    if (!pDataBase->open()) {
        qDebug() << "CScenarioManager::listeScenarios - problème ouverture base";
    } else {
        QSqlQuery *sqlQuery = new QSqlQuery(DataBase->database());
        sqlQuery->prepare("SELECT * FROM simulation_scenario");
        sqlQuery->exec();

        CScenario *scenario = nullptr;
        while (sqlQuery->next()) {
            //Création d'un objet scenario
            scenario = new CScenario();
            scenario->setID(sqlQuery->value("sc_id").toInt());
            scenario->setNom(sqlQuery->value("sc_nom").toString());
            //Ajout de l'objet créé dans vectPassage
            vectScenarios.push_back(scenario);
        }
    }
    return vectScenarios;
}
```

```

/*!
 * \brief CScenarioManager::addScenario
 * \param scenario
 * \return ret
 * \details ajoute un scenario avec les passages ds la bdd
 */
bool CScenarioManager::addScenario(const CScenario &scenario)
{
    bool ret = false;
    unsigned sc_id = 0;
    if (!pDataBase->open()) {
        qDebug() << "CScenarioManager::addScenario - problème ouverture base";
        return false;
    } else {
        QSqlQuery *sqlQuery = new QSqlQuery(pDataBase->database());
        sqlQuery->prepare("INSERT INTO `simulation_scenario`(`sc_nom`) VALUES (:nom)");
        sqlQuery->bindValue(":nom", scenario.getNom());
        sqlQuery->exec();

        //recuperation de l'id de lenregistrement
        sqlQuery->prepare("select MAX(sc_id) from simulation_scenario");
        sqlQuery->exec();
        while(sqlQuery->next())
            sc_id = sqlQuery->value("MAX(sc_id)").toUInt();
        //ajout des passages et affectation au scenario
        int i = 0;
        for(CPassage *passage : scenario.getVectPassage())
        {
            passage->setVitesse(scenario.getVectPassage()[i]-
>getVitesse());
            passage->setInterVehicule(scenario.getVectPassage()[i]-
>getInterVehicule());
            CVEhicule *vehicule = new CVEhicule();
            vehicule->setNom(scenario.getVectPassage()[i]->getVehicule()-
>getNom());
            passage->setVehicule(vehicule);
            pPassageManager->addPassage(*passage, sc_id);
            i += 1;
        }
        return ret;
    }
}
/*!
 * \brief CScenarioManager::del
 * \param id
 * \return ret
 * \details supprime le scenario avec les passages ds la bdd
 */
bool CScenarioManager::delScenario(QString nom)
{
    bool ret = false;
    if (!pDataBase->open()) {
        qDebug() << "CScenarioManager::del - problème ouverture base";
        return false;
    } else {
        //suppression des passages du scenario
        getScenarioByNom(nom);
        for (CPassage *Passage : getScenarioByNom(nom)->getVectPassage())
        {
            pPassageManager->delPassage(Passage->getPassageId());
        }
        //Suppression du scenario
    }
}

```

```

        QSqlQuery *sqlQuery = new QSqlQuery (pDataBase->database());
        sqlQuery->prepare("DELETE FROM simulation_scenario WHERE sc_nom =
:nom");
        sqlQuery->bindValue(":nom", nom);
        ret = sqlQuery->exec() ;
        return ret;
    }
}
*/
/*!
* \brief CScenarioManager::getSimulationByScenario
* \param nom
* \return Scenario
* \details revoie les passages sur la scenario à partir du nom
*/
CScenario *CScenarioManager::getScenarioByNom(QString nom)
{
    if (!pDataBase->open()) {
        qDebug() << "CScenarioManager::getPassageByScenario- problème
ouverture base";
        return nullptr;
    } else {
        QSqlQuery *sqlQuery = new QSqlQuery(pDataBase->database());
        sqlQuery->prepare("SELECT * FROM simulation_scenario WHERE sc_nom
= :nom ");
        sqlQuery->bindValue(":nom", nom);
        sqlQuery->exec();

        CScenario *Scenario = nullptr;
        while (sqlQuery->next()) {
            //Création d'un objet scenario
            Scenario = new CScenario();
            Scenario->setId(sqlQuery->value("sc_id").toInt());
            Scenario->setNom(sqlQuery->value("sc_nom").toString());
        }
        CPassage *passage = nullptr;
        passage = new CPassage();
        //recuperation passage
        for (long unsigned i = 0; i < pPassageManager-
>getPassageByScenarioId(Scenario->getId()).size(); i++) {
            passage = pPassageManager-
>getPassageByScenarioId(Scenario->getId())[i];
            if(passage!=nullptr)
                Scenario->addVectPassage(passage);
        }
        return Scenario;
    }
}
/*!
* \brief CScenarioManager::creeJson
* \param emplacement
* \param scenario
* \return bool
* \details remplissage fichier Json
*/
bool CScenarioManager::creeJson(const QString emplacement, CScenario
*scenario)
{
    QJsonObject objectJSON;
    QJsonObject passage;
    QJsonArray tabPassage;

    // creation object Json : scenario
    objectJSON.insert("nomScenario", scenario->getNom());
    // remplissage des passages

```

```

    for (long unsigned i = 0; i < scenario->getVectPassage().size(); i++)
    {
        //remplissage vehicule
        // passage.insert("nomVehicule", scenario->getVectPassage() [i]-
>getVehicule()->getNom());
        passage.insert("nbreEssieux", scenario->getVectPassage() [i]->getVehicule()->getEssieux());
        passage.insert("typeVehicule", scenario->getVectPassage() [i]->getVehicule()->getType());
        //tableau empattement
        QJsonArray tabEmpattement;
        for (int i2 = 0; i2 < scenario->getVectPassage() [i]->getVehicule()->getEssieux() - 1; i2++)
            tabEmpattement.push_back(round(scenario->getVectPassage() [i]->getVehicule()->getEmpattement() [i2]*100)/100); //ROund : arondi decimal
        passage.insert("empattement",tabEmpattement);
        //remplissage passage du vehicule
        passage.insert("vitesseVehicule", scenario->getVectPassage() [i]->getVitesse());
        passage.insert("tempsInterVehicule", scenario->getVectPassage() [i]->getInterVehicule());
        //tableau du passage
        tabPassage.push_back(passage);
    }
    objectJSON.insert("passages",tabPassage);
    QJsonDocument document;
    document.setObject(objectJSON);

    QByteArray bytes =
document.toJson(/*QJsonDocument::Compact*/QJsonDocument::Indented);
    QFile file(emplacement);
    if( file.open( QIODevice::WriteOnly | QIODevice::Text |
QIODevice::Truncate ) )
{
    QTextStream iStream( &file );
    iStream.setCodec( "utf-8" );
    iStream << bytes;
    file.close();
    return true;
}
else
{
    return false;
}
}
*/
/*!
* \brief CScenarioManager::transfertJson
* \param emplacement
* \param nomPort
* \return reponse
* \details envoie sur la liason serie
*/
int CScenarioManager::transfertJson(const QString emplacement, const
QString nomPort)
{
    serialPort.close();
    int reponse = 0;
    QString portWindows = nomPort;
    portWindows.truncate(3);
    if (portWindows == "COM") {
        serialPort.setPortName(nomPort);
        serialPort.setBaudRate(9600);
    } else {
        serialPort.setPortName("/dev/" + nomPort);
}

```

```
    serialPort.setBaudRate(9600);

}

//probleme erreur port serie
if (!serialPort.open(QIODevice::WriteOnly)) {
    reponse = 1;
}
QFile dataFile(emplacement);
//probleme Erreur emplacement Json
if (!dataFile.open(QIODevice::ReadOnly))
{
    serialPort.close();
    reponse = 2;
}
QByteArray writeData(dataFile.readAll());
dataFile.close();
//probleme Json vide
if (writeData.isEmpty())
{
    reponse = 3;
}

qint64 bytesWritten = serialPort.write(writeData);
char fin = 0x04;
bytesWritten = serialPort.write(&fin,1);

if (bytesWritten == -1) {
} else if (bytesWritten != writeData.size()) {

} else if (!serialPort.waitForBytesWritten(5000)) {
}
return reponse;

}
```

cdialovehicule.h

```
/***
 * \file cdialovehicule.h
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/03/2021
 * \details Classe conteneur pour l'IHM de dialogueConfigVehicule
 */
#ifndef CDIALOVEHICULE_H
#define CDIALOVEHICULE_H
#include "QObject"
#include "QString"
#include "cvehicule.h"

class CDialoVehicule : public QObject
{
    Q_OBJECT
public:
    explicit CDialoVehicule(QObject *parent = 0);
    CDialoVehicule(QString nomBouton, CVehicule *vehicule, QObject
*parent);
    ~CDialoVehicule();

    QString getNomPage() const;
    void setNomBouton(const QString &value);

    CVehicule *getVehicule() const;
    void setVehicule(CVehicule *value);

private:
    QString pNomBouton;
    CVehicule *pVehicule = nullptr;
};

#endif // CDIALOVEHICULE_H
```

Cdialovehicule.cpp

```
/***
 * \file cdialovehicule.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/03/2021
 * \details Classe conteneur pour l'IHM de dialogueConfigVehicule
 */
#include "cdialovehicule.h"
*/
/*!
 * \brief CDialoVehicule::CDialoVehicule
 * \param parent
 * \details constructeur par defaut
 */
CDialoVehicule::CDialoVehicule(QObject *parent) : QObject(parent)
{
}

*/
/*!
 * \brief CDialoVehicule::CDialoVehicule
 * \param nom_bouton
 * \param vehicule
 * \param parent
 * \details passage des parametres
 */
CDialoVehicule::CDialoVehicule(QString nom_bouton, CVehicule *vehicule,
QObject *parent): QObject(parent)
{
    pNomBouton = nom_bouton;
    pVehicule = vehicule;
}
*/
/*!
 * \brief CDialoVehicule::~CDialoVehicule
 * \details destructeur
 */
CDialoVehicule::~CDialoVehicule()
{
}

*/
/*!
 * \brief CDialoVehicule::getNomPage
 * \return pNomBouton
 * \details accesseur pNomBouton
 */
QString CDialoVehicule::getNomPage() const
{
    return pNomBouton;
}
*/
/*!
 * \brief CDialoVehicule::setNomBouton
 * \param value
 * \details mutateur pNomBouton
 */
void CDialoVehicule::setNomBouton(const QString &value)
{
    pNomBouton = value;
}
*/
/*!
 * \brief CDialoVehicule::getVehicule
 * \return pVehicule
 */
```

```

* \details accesseur pVehicule
*/
CVehicule *CDialoVehicule::getVehicule() const
{
    return pVehicule;
}
/*!!
* \brief CDialoVehicule::setVehicule
* \param value
* \details mutateur pVehicule
*/
void CDialoVehicule::setVehicule(CVehicule *value)
{
    pVehicule = value;
}

```

cdialoscenario.h

```

/***
* \file cdialoscenario.h
* \brief Système de simulation Qt
* \author Arthur LABARRE
* \copyright Copyright (c) 2021 Arthur LABARRE
* \version 1.0
* \date 10/04/2021
* \details Classe conteneur pour l'IHM dialogueConfigScenario
*/
#ifndef CDIALOSCENARIO_H
#define CDIALOSCENARIO_H

#include <QObject>
#include "QString"
#include "cscenario.h"

class CDialoScenario : public QObject
{
    Q_OBJECT
public:
    explicit CDialoScenario(QObject *parent = 0);
    CDialoScenario(QString nomBouton, CScenario *scenario,
std::vector<QString> vehiculeDispo, QObject *parent);
    ~CDialoScenario();

    QString getNomBouton() const;
    void setNomBouton(const QString &value);

    CScenario *getScenario() const;
    void setScenario(CScenario *value);

    std::vector<QString> getVehiculeDispo() const;
    void setVehiculeDispo(const std::vector<QString> &value);

private:
    QString pNomBouton;
    CScenario *pScenario = nullptr;
    std::vector<QString> pVehiculeDispo;
};

#endif // CDIALOSCENARIO_H

```

cdialoscenario.cpp

```
/***
 * \file cdialoscenario.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/04/2021
 * \details Classe conteneur pour l'IHM dialogueConfigScenario
 */
#include "cdialoscenario.h"
/*!
 * \brief CDialoScenario::CDialoScenario
 * \param parent
 * \details constructeur par defaut
 */
CDialoScenario::CDialoScenario(QObject *parent): QObject(parent)
{
}
/*!
 * \brief CDialoScenario::CDialoScenario
 * \param nomBouton
 * \param scenario
 * \param vehiculeDispo
 * \param parent
 * \details passage des parametres
 */
CDialoScenario::CDialoScenario(QString nomBouton, CScenario *scenario,
std::vector<QString> vehiculeDispo, QObject *parent)
: QObject(parent)
{
    nomBouton = pNomBouton;
    scenario = pScenario;
    vehiculeDispo = pVehiculeDispo;
}
/*!
 * \brief CDialoScenario::~CDialoScenario
 * \details destructureur
 */
CDialoScenario::~CDialoScenario()
{
}

/*!
 * \brief CDialoScenario::getNomBouton
 * \return pNomBouton
 * \details accesseur pNomBouton
 */
QString CDialoScenario::getNomBouton() const
{
    return pNomBouton;
}
/*!
 * \brief CDialoScenario::setNomBouton
 * \param value
 * \details mutateur pNomBouton
 */
void CDialoScenario::setNomBouton(const QString &value)
{
    pNomBouton = value;
}
```

```
}

/*!
* \brief CDialoScenario::getScenario
* \return pScenario
* \details accesseur pScenario
*/
CScenario *CDialoScenario::getScenario() const
{
    return pScenario;
}
/*!
* \brief CDialoScenario::setScenario
* \param value
* \details mutateur pScenario
*/
void CDialoScenario::setScenario(CScenario *value)
{
    pScenario = value;
}
/*!
* \brief CDialoScenario::getVehiculeDispo
* \return pVehiculeDispo
* \details accesseur pVehiculeDispo
*/
std::vector<QString> CDialoScenario::getVehiculeDispo() const
{
    return pVehiculeDispo;
}
/*!
* \brief CDialoScenario::setVehiculeDispo
* \param value
* \details mutateur pVehiculeDispo
*/
void CDialoScenario::setVehiculeDispo(const std::vector<QString> &value)
{
    pVehiculeDispo = value;
}
```

dialogueconfigsvehicule.h

```
/***
 * \file dialogueconfigsvehicule.h
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/03/2021
 * \details classe permettant créer un véhicule
 */
#ifndef DIALOGUECONFIGVEHICULE_H
#define DIALOGUECONFIGVEHICULE_H
#include "cdialovehicule.h"
#include <QDialog>
#include <QTableWidget>

namespace Ui {
class DialogueConfigVehicule;
}

class DialogueConfigVehicule : public QDialog
{
    Q_OBJECT

public:
    explicit DialogueConfigVehicule(CDialoVehicule *_Vehicule, QWidget
*parent = nullptr);
    ~DialogueConfigVehicule();

private slots:
    void on_pb_valider_released();
    void on_sp_nombreEssieux_valueChanged(int arg1);

private:
    Ui::DialogueConfigVehicule *ui;
    CDialoVehicule *pDialogVehicule;
};

#endif // DIALOGUECONFIGVEHICULE_H
```

dialogueconfigsvehicule.cpp

```
/***
 * \file dialogueconfigsvehicule.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/03/2021
 * \details classe permettant créer un véhicule
 */
#include "dialogueconfigvehicule.h"
#include "ui_dialogueconfigvehicule.h"

/*!
 * \brief DialogueConfigVehicule::DialogueConfigVehicule
 * \param _Vehicule
 * \param parent
 * \details constructeur par défaut
 */
DialogueConfigVehicule::DialogueConfigVehicule(CDialogVehicule *_Vehicule,
QWidget *parent)
    : QDialog(parent), ui(new Ui::DialogueConfigVehicule)
{
    ui->setupUi(this);
    QObject::connect(ui->pb_valider, SIGNAL(clicked()), this,
SLOT(accept()));
    QObject::connect(ui->pb_annuler, SIGNAL(clicked()), this,
SLOT(reject()));
    pDialogVehicule = _Vehicule;
    //maj comboBox
    ui->cb_typeVehicule->addItem("Véhicule léger");
    ui->cb_typeVehicule->addItem("Poids lourd");
    //min et max de nombres essieux
    ui->sp_nombreEssieux->setMinimum(2);
    ui->sp_nombreEssieux->setMaximum(10);
    //maj nom
    ui->le_nomVehicule->setText(pDialogVehicule->getVehicule()-
>getNom());

    if (pDialogVehicule->getNomPage() == "&Créer")
        this->setWindowTitle("Créer un véhicule");
    else {
        //maj voiture selec
        this->setWindowTitle("Modifier véhicule : " + pDialogVehicule-
>getVehicule()->getNom());
        //maj widget
        if (pDialogVehicule->getVehicule()->getType() == "vl")
            ui->cb_typeVehicule->setCurrentText("Véhicule léger");
        else
            ui->cb_typeVehicule->setCurrentText("Poids lourd");
        ui->sp_nombreEssieux->setValue(pDialogVehicule->getVehicule()-
>getEssieux());
    }
}
/*! \brief DialogueConfigVehicule::~DialogueConfigVehicule
 * \details destructeur
 */
DialogueConfigVehicule::~DialogueConfigVehicule()
{
    delete ui;
}
```

```

/*
 * \brief DialogueConfigVehicule::on_sp_nombreEssieux_valueChanged
 * \param arg1
 * \details maj tableWidget à partir du spinBox
 */
void DialogueConfigVehicule::on_sp_nombreEssieux_valueChanged(int arg1)
{
    QStringList titre;
    QString inter_roues;
    ui->tw_empatement->setRowCount(1); //creer une ligne
    ui->tw_empatement->setColumnCount(arg1-1); //creer les colonnes en
fonction des essieux
    for (int i = 0; arg1 - 1 > i; i++) { //boucle qui tourne pendant x
fois le nombre essieux -1
        QTableWidgetItem *empatement = new
QTableWidgetItem(QString::number(pDialogVehicule->getVehicule()-
>getEmpatement()[i])); //creer une table pour empatement
        empatement->setTextAlignment(Qt::AlignHCenter); //alligne au
centre
        titre += inter_roues = QString(" %1 ").arg(i+1); //créer le titre
        ui->tw_empatement->setItem(0, i, empatement); //ajoute les valeurs
dans les tables
    }
    ui->tw_empatement->setHorizontalHeaderLabels(titre); //ajoute les
titres
}

/*
 * \brief DialogueConfigVehicule::on_pb_valider_released
 * \details pb validation et envoie des parametres du vehicule a la
classe CDialogueVehicule
 */
void DialogueConfigVehicule::on_pb_valider_released()
{
    CVehicule *vehicule = new CVehicule(this);
    vehicule->setNom(ui->le_nomVehicule->text());

    //maj diminutif vl et pl
    if (ui->cb_typeVehicule->currentText() == "Véhicule léger")

        vehicule->setType("vl");
    else
        vehicule->setType("pl");

    vehicule->setEssieux(ui->sp_nombreEssieux->value());
    std::vector<float> vectEmpatement;
    //maj vector empatement avec boucle for
    for (int i = 0; i < vehicule->getEssieux() - 1 ; i++) { //boucle qui
tourne pendant x fois le nombre essieux -1
        vectEmpatement.push_back(ui->tw_empatement->item(0, i)-
>text().toFloat());
    }
    for (int i = vehicule->getEssieux() ; i <= 9 ; i++ ) {
        vectEmpatement.push_back(0.0);
    }
    vehicule->setEmpatement(vectEmpatement);
    pDialogVehicule->setVehicule(vehicule);
}

```

dialogueconfigscenario.h

```
/***
 * \file dialogueconfigscenario.h
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/04/2021
 * \details classe permettant créer un scenario
 */
#ifndef DIALOGUECONFIGSCENARIO_H
#define DIALOGUECONFIGSCENARIO_H
#include "cdialogscenario.h"
#include <QDialog>
#include <QTableWidget>
#include <QMMessageBox>
#include "QDebug"

namespace Ui {
class DialogueConfigScenario;
}

class DialogueConfigScenario : public QDialog
{
    Q_OBJECT

public:
    explicit DialogueConfigScenario(CDialoScenario * _dialogScenario,
QWidget *parent = nullptr);
    ~DialogueConfigScenario();

private slots:
    void on_pb_valider_released();
    void on_lw_listeVehiculeDisponible_itemSelectionChanged();
    void on_pb_ajouter_released();
    void on_pb_supprimer_released();
    void on_pb_monter_released();
    void on_pb_descendre_released();

private:
    void affichePassage();
    Ui::DialogueConfigScenario *ui;
    CDialoScenario *pDialogScenario;
};

#endif // DIALOGUECONFIGSCENARIO_H
```

dialogueconfigscenario.cpp

```
/***
 * \file dialogueconfigscenario.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/04/2021
 * \details classe permettant créer un scenario
 */
#include "dialogueconfigscenario.h"
#include "ui_dialogueconfigscenario.h"

/*!
 * \brief DialogueConfigScenario::DialogueConfigScenario
 * \param _dialogScenario
 * \param parent
 * \details constructeur par défaut
 */
DialogueConfigScenario::DialogueConfigScenario(CDialogScenario
*_dialogScenario, QWidget *parent) :
    QDialog(parent),
    ui(new Ui::DialogueConfigScenario)
{
    ui->setupUi(this);
    QObject::connect(ui->pb_valider, SIGNAL(clicked()), this,
SLOT(accept()));
    QObject::connect(ui->pb_annuler, SIGNAL(clicked()), this,
SLOT(reject()));

    pDialogScenario = _dialogScenario;
    if (pDialogScenario->getNomBouton() == "&Créer")
        this->setWindowTitle("Créer un scenario");
    else{
        this->setWindowTitle("Modifier un scénario : "+pDialogScenario-
>getScenario()->getNom());
        affichePassage();
    }
    //affiche les véhicules dispo
    for (long unsigned i = 0;i < pDialogScenario->getVehiculeDispo().size();
i++) {
        ui->lw_listeVehiculeDisponible->addItem(pDialogScenario-
>getVehiculeDispo()[i]);
    }
    //verrouille edit
    ui->le_nomVehicule->setReadOnly(true);
    //maximum
    ui->sp_vitesse->setMaximum(135);
    //par défaut
    ui->sp_vitesse->setValue(50);
    ui->sp_tempsInterVehicule->setValue(2);
    ui->le_nomDuScenario->setText(pDialogScenario->getScenario()->getNom());
}
/*!
 * \brief DialogueConfigScenario::~DialogueConfigScenario
 * \details destructeur
 */
DialogueConfigScenario::~DialogueConfigScenario()
{
    delete ui;
}
```

```

/*!
 * \brief DialogueConfigScenario::affichePassage
 * \details Affiche les passages du scenario à modifier
 */
void DialogueConfigScenario::affichePassage()
{
    //maj tableWidget
    QStringList titre;
    titre << "Nom" << "Vitesse" << "Temps inter-vehicule";
    ui->tw_listePassage->setEditTriggers(QAbstractItemView
::NoEditTriggers); //verouille tab;
    ui->tw_listePassage->setRowCount(pDialogScenario->getScenario()-
>getVectPassage().size()); //creer des lignes en fonctions du nombre de
passages
    ui->tw_listePassage->setColumnCount(3);
    //ajout dans les tableaux
    for (long unsigned i = 0; i < pDialogScenario->getScenario()-
>getVectPassage().size(); i++) {
        QTableWidgetItem *vehicule = new QTableWidgetItem(pDialogScenario-
>getScenario()->getVectPassage()[i]->getVehicule()->getNom());
        QTableWidgetItem *vitesse = new
QTableWidgetItem(QString::number(pDialogScenario->getScenario()-
>getVectPassage()[i]->getVitesse()));
        QTableWidgetItem *interVehicule = new
QTableWidgetItem(QString::number(pDialogScenario->getScenario()-
>getVectPassage()[i]->getInterVehicle())));
        vehicule->setTextAlignment(Qt::AlignHCenter);
        vitesse->setTextAlignment(Qt::AlignHCenter);
        interVehicule->setTextAlignment(Qt::AlignHCenter);
        ui->tw_listePassage->setItem(i,0,vehicule);
        ui->tw_listePassage->setItem(i,1,vitesse);
        ui->tw_listePassage->setItem(i,2,interVehicule);
    }
    ui->tw_listePassage->setHorizontalHeaderLabels(titre);
}

/*!
 * \brief
DialogueConfigScenario::on_lw_listeVehiculeDisponible_itemSelectionChanged
 * \details ajoute le nom vehicule lineEdit
 */
void
DialogueConfigScenario::on_lw_listeVehiculeDisponible_itemSelectionChanged()
{
    if (ui->lw_listeVehiculeDisponible->currentRow() > -1)
        ui->le_nomVehicule->setText(ui->lw_listeVehiculeDisponible-
>currentItem()->text());
    else
        ui->lw_listeVehiculeDisponible->setCurrentRow(0);
}
/*!
 * \brief DialogueConfigScenario::on_pb_ajouter_released
 * \details ajoute le passage au tableau
 */
void DialogueConfigScenario::on_pb_ajouter_released()
{
    if (ui->le_nomVehicule->text() != "") {
        int ligne = ui->tw_listePassage->rowCount(); //nbre ligne
        QStringList titre;
        titre << "Nom" << "Vitesse" << "Temps inter-vehicule";
        ui->tw_listePassage->setEditTriggers(QAbstractItemView
::NoEditTriggers); //verouille tab;
        ui->tw_listePassage->setColumnCount(3);
        //parametre tableWidgetItem

```

```

QTableWidgetItem *nomVehicule = new QTableWidgetItem(ui-
>lw_listeVehiculeDisponible->currentItem()->text());
    QTableWidgetItem *vitesse = new QTableWidgetItem(QString::number(ui-
>sp_vitesse->value()));
    QTableWidgetItem *interVehicule = new
QTableWidgetItem(QString::number(ui->sp_tempsInterVehicule->value()));
    nomVehicule->setTextAlignment(Qt::AlignHCenter);
    vitesse->setTextAlignment(Qt::AlignHCenter);
    interVehicule->setTextAlignment(Qt::AlignHCenter);
//ajout et selectionne ligne
ui->tw_listePassage->setRowCount(ligne+1);
int endLigne = ui->tw_listePassage->rowCount()-1;
//ajout dans les tableaux
ui->tw_listePassage->setItem(endLigne, 0, nomVehicule);
ui->tw_listePassage->setItem(endLigne, 1, vitesse);
ui->tw_listePassage->setItem(endLigne, 2, interVehicule);
ui->tw_listePassage->setHorizontalHeaderLabels(titre);
}
else
    QMessageBox::information(this,"Ajout un passage","Vous devez ajouter
un vehicule");

}
/*!
* \brief DialogueConfigScenario::on_pb_supprimer_released
* \details supprime un passage au tableau
*/
void DialogueConfigScenario::on_pb_supprimer_released()
{
    if (ui->tw_listePassage->currentRow() > -1)
        ui->tw_listePassage->removeRow(ui->tw_listePassage->currentRow());
    else
        QMessageBox::information(this,"Supprimer un passage","Vous devez
selectionner la ligne à effacer");
}

/*
* \brief DialogueConfigScenario::on_pb_monter_released
* \details monte un passage au tableau
*/
void DialogueConfigScenario::on_pb_monter_released()
{
    int selecLigne = ui->tw_listePassage->currentRow();
    if (ui->tw_listePassage->currentRow() > -1)
    {
        if(ui->tw_listePassage->currentRow() > 0)
        {
            //recuperation des items
            QTableWidgetItem *nomVehicule = new QTableWidgetItem( ui-
>tw_listePassage->item(selecLigne,0)->text());
            QTableWidgetItem *vitesse = new QTableWidgetItem(QString::number(ui-
>tw_listePassage->item(selecLigne,1)->text().toInt()));
            QTableWidgetItem *interVehicule = new
QTableWidgetItem(QString::number(ui->tw_listePassage->item(selecLigne,2)-
>text().toInt()));
            //affichage aligné
            nomVehicule->setTextAlignment(Qt::AlignHCenter);
            vitesse->setTextAlignment(Qt::AlignHCenter);
            interVehicule->setTextAlignment(Qt::AlignHCenter);
            //maj new table
            ui->tw_listePassage->removeRow(selecLigne);
            ui->tw_listePassage->insertRow(selecLigne-1);
        }
    }
}

```

```

        ui->tw_listePassage->setItem(selecLigne - 1, 0, nomVehicule);
        ui->tw_listePassage->setItem(selecLigne - 1, 1, vitesse);
        ui->tw_listePassage->setItem(selecLigne - 1, 2, interVehicule);
    }
} else
    QMessageBox::information(this,"Monter un passage","Vous devez
selectionner la ligne à monter");
}

/*
 * \brief DialogueConfigScenario::on_pb_descendre_released
 * \details descend un vehicule au tableau
 */
void DialogueConfigScenario::on_pb_descendre_released()
{
    int selecLigne = ui->tw_listePassage->currentRow();
    if (ui->tw_listePassage->currentRow() > -1)
    {
        if (ui->tw_listePassage->currentRow() < ui->tw_listePassage-
>rowCount()-1)
        {
            //recuperation des items
            QTableWidgetItem *nomVehicule = new QTableWidgetItem(ui-
>tw_listePassage->item(selecLigne,0)->text());
            QTableWidgetItem *vitesse = new QTableWidgetItem(QString::number(ui-
>tw_listePassage->item(selecLigne,1)->text().toInt()));
            QTableWidgetItem *interVehicule = new
QTableWidgetItem(QString::number(ui->tw_listePassage->item(selecLigne,2)-
>text().toInt()));
            //affichage aligné
            nomVehicule->setTextAlignment(Qt::AlignHCenter);
            vitesse->setTextAlignment(Qt::AlignHCenter);
            interVehicule->setTextAlignment(Qt::AlignHCenter);
            //maj new table
            ui->tw_listePassage->removeRow(selecLigne);
            ui->tw_listePassage->insertRow(selecLigne+1);
            ui->tw_listePassage->setItem(selecLigne + 1, 0, nomVehicule);
            ui->tw_listePassage->setItem(selecLigne + 1, 1, vitesse);
            ui->tw_listePassage->setItem(selecLigne + 1, 2, interVehicule);
        }
    } else
        QMessageBox::information(this,"Descendre un passage","Vous devez
selectionner la ligne à descendre");
}

/*
 * \brief DialogueConfigScenario::on_pb_valider_released
 * \details pb validation et envoie des parametres du scenario a la classe
CDialogueScenario
 */
void DialogueConfigScenario::on_pb_valider_released()
{
    std::vector<QString> vectNomVehicule;
    std::vector<int> vectVitesse;
    std::vector<int> vectInterVehicule;

    for (int i = 0; i < ui->tw_listePassage->rowCount(); i++) {
        vectNomVehicule.push_back(ui->tw_listePassage->item(i, 0)->text());
        vectVitesse.push_back(ui->tw_listePassage->item(i, 1)-
>text().toInt());
        vectInterVehicule.push_back(ui->tw_listePassage->item(i, 2)-
>text().toInt());
    }
    //creation object scenario
}

```

```
CScenario *scenario = new CScenario(this);
scenario->setNom(ui->le_nomDuScenario->text());
for (int unsigned i = 0; i < vectVitesse.size(); i++) {
    //creation object vehicule et passage
    CVehicule *vehicule = new CVehicule(this);
    vehicule->setNom(vectNomVehicule[i]);
    //creation object et passage
    CPassage *passage = new CPassage(this);
    passage->setVehicule(vehicule);
    passage->setVitesse(vectVitesse[i]);
    passage->setInterVehicule(vectInterVehicule[i]);
    scenario->addVectPassage(passage);
}
pDialogScenario->setScenario(scenario);
close();
}
```

mainwindow.h

```
/***
 * \file mainwindow.h
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/02/2021
 * \details classe permettant de simuler un trafic et connaître ces
statistiques pour le projet KOLANTR.
*/
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include "dialogueconfigscenario.h"
#include "cvehicule.h"
#include "cvehiculemanager.h"
#include "dialogueconfigvehicule.h"
#include "cpassage.h"
#include "cpassagemanager.h"
#include "cscenariomanager.h"
#include "cdialosscenario.h"
#include <QMMessageBox>
#include <QSerialPortInfo>
#include "QFileDialog"

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void on_pb_quitter_released();
    void on_pb_accueilVehicule_released();
    void on_pb_acceuilScenario_released();
    void on_pb_accueilStat_released();
    void on_pb_vehicule_released();
    void on_pb_scenario_released();
    void on_pb_stats_released();
    void on_pb_creerVehicule_released();
    void on_pb_CreerSCenario_released();
    void on_pb_transfertScenario_released();
}
```

```

void on_lw_listeVehicule_itemSelectionChanged() // liste affiche les
paramètres du véhicule sélectionné

void on_pb_supprimerVehicule_released();

void on_pb_modifierVehicule_released();

void on_lw_listeScenarios_itemSelectionChanged();

void on_pb_supprimerScenario_released();

void on_pb_modifierScenario_released();

void on_action_Quitter_triggered();

void on_actionJson_triggered();

void on_actionBase_de_donnee_triggered();

void on_actionSite_triggered();

void on_lw_listeDesScenarios_itemSelectionChanged();

private:
Ui::MainWindow *ui;
QSqlDatabase * pDataBase = nullptr;
CVehicleManager *pVehicleManager;
CPassageManager *pPassageManager;
CScenarioManager *pScenarioManager;

void listerLesVehicules();
void afficheSimulation();
void listerScenario();
void affichePortDisponible();
void initBdd();

QString pPathBdd;
QString pPathJson;
QSettings *mQSettings ;

};

#endif // MAINWINDOW_H

```

Mainwindows.cpp

```
/***
 * \file mainwindow.cpp
 * \brief Système de simulation Qt
 * \author Arthur LABARRE
 * \copyright Copyright (c) 2021 Arthur LABARRE
 * \version 1.0
 * \date 10/02/2021
 * \details classe permettant de simuler un trafic et connaître ces statistiques pour le projet KOLANTR.
 */
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QDesktopServices>

/*!
 * \brief MainWindow::MainWindow
 * \param parent
 * \details constructeur par défaut
 */
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    this->setWindowTitle("Kolantr config simulation");

    // Instanciation de l'objet mQSettings
    mQSettings = new QSettings(QSettings::UserScope, "Qt-QSettings",
    "QSettingPrefs");
    //mise en place des icons
    QPixmap icon_vehicule (":Icon/icon_voiture"),
icon_clap(":Icon/icon_clap"), icon_stat(":Icon/icon_stat"),
icon_fleche(":Icon/icon_fleche");
    ui->lb_imageVehicule->setPixmap(icon_vehicule.scaled(ui-
>lb_imageVehicule->width(), ui->lb_imageVehicule->height(),
Qt::KeepAspectRatio));
    ui->lb_imageScenario->setPixmap(icon_clap.scaled(ui-
>lb_imageScenario->width(), ui->lb_imageScenario->height(),
Qt::KeepAspectRatio));
    ui->lb_imageStat->setPixmap(icon_stat.scaled(ui->lb_imageStat-
>width(), ui->lb_imageStat->height(), Qt::KeepAspectRatio));

    //Vérouille l'écriture des lineEdit
    ui->le_nombresVehiculePasss->setReadOnly(true);
    ui->le_dureeScenario->setReadOnly(true);
    ui->le_vitesseMoyVL->setReadOnly(true);
    ui->le_NombresPassesVL->setReadOnly(true);
    ui->le_nomDuVehicule->setReadOnly(true);
    ui->le_nombresEssieux->setReadOnly(true);
    ui->le_NombresPassesPL->setReadOnly(true);
    ui->le_typeDeVehicule->setReadOnly(true);
    ui->le_vitesseMoy->setReadOnly(true);
    ui->le_VitesseMoyPL->setReadOnly(true);

    //remplie les ports series pour liaison série disponible
    affichePortDisponible();

    //etape en attente
    ui->lb_etape1->setStyleSheet("background-color: orange;");
    ui->lb_etape1->setText("En attente\nmaj Json");
    ui->lb_etape2->setText("En attente\nnd'envoie");
}
```

```

ui->lb_etape2->setStyleSheet("background-color: orange ;");

initBdd();
}

/*
 * \brief MainWindow::~MainWindow
 * \details destructeur
 */
MainWindow::~MainWindow()
{
    delete ui;
}
/*
 * \brief MainWindow::on_actionJson_triggered
 * \details selectionne le chemin du fichier json grace à une fenêtre de dialogue
 */
void MainWindow::on_actionJson_triggered()
{
    QString emplacement = QFileDialog::getOpenFileName(this,"Ouverture fichier Json","scenario","Fichier JSON (*.json)");
    //si utilisateur selectionne le fichier
    if (emplacement != nullptr){
        pPathJson = emplacement;
        mQSettings->setValue("cheminJSON", pPathJson);
    }
}
/*
 * \brief MainWindow::on_actionBase_de_donnee_triggered
 * \details selectionne le chemin de la bdd grace à une fenêtre de dialogue et cree object pdatabase
 */
void MainWindow::on_actionBase_de_donnee_triggered()
{
    QString emplacement = QFileDialog::getOpenFileName(this,"Ouverture fichier base de donnée","bdd_simulation","Fichier bdd (*.Fichier)");
    //si utilisateur selectionne le fichier
    if(emplacement != nullptr){
        pPathBdd = emplacement;
        mQSettings->setValue("cheminBDD", pPathBdd);
        initBdd();
    }
}
/*
 * \brief MainWindow::initBddJson
 * \details Initialise chemin bdd et json avec les préférences
 */
void MainWindow::initBdd()
{
    pPathBdd = mQSettings->value("cheminBDD",pPathBdd).toString();
    //Création d'un objet QSQSqlDatabase permettant l'accès à la base de données
    pDataBase = new QSqlDatabase(QSqlDatabase::addDatabase("QSQLITE"));
    // ATTENTION VERIFIER LE CHEMIN
    pDataBase->setDatabaseName(pPathBdd);

    //Création d'un objet CVehiculeManager permettant de gérer les véhicules dans la base de données
    pVehiculeManager = new CVehicleManager(pDataBase);

    //Création d'un objet CSimulationPassageManager permettant de gérer les passages dans la base de données
    pPassageManager = new CPassageManager (pDataBase);
}

```

```

    //Création d'un objet CScenarioManager permettant de gérer les
    scenarios dans la base de données
    pScenarioManager = new CScenarioManager (pDataBase);

    //Remplie lw_listeVehicule avec les noms des véhicules
    listerScenario();
    listerLesVehicules();
}

/*
 * \brief MainWindow::on_actionSite_triggered
 * \details lance le site de consultation
 */
void MainWindow::on_actionSite_triggered()
{
    QDesktopServices::openUrl(QUrl("http://kolantr.snkastler.fr:8000/"));
}

void MainWindow::on_pb_quitter_released()
{
    close();
}

void MainWindow::on_action_Quitter_triggered()
{
    close();
}

void MainWindow::on_pb_accueilVehicule_released()
{
    ui->tabWidget->setcurrentIndex(0);
}

void MainWindow::on_pb_acceuilScenario_released()
{
    ui->tabWidget->setcurrentIndex(0);
}

void MainWindow::on_pb_accueilStat_released()
{
    ui->tabWidget->setcurrentIndex(0);
}

void MainWindow::on_pb_vehicule_released()
{
    ui->tabWidget->setcurrentIndex(1);
}

void MainWindow::on_pb_scenario_released()
{
    ui->tabWidget->setcurrentIndex(2);
}

void MainWindow::on_pb_stats_released()
{
    ui->tabWidget->setcurrentIndex(3);
}

    //Page Vehicule

*/

```

```

/* \brief MainWindow::on_pb_creerVehicule_released
 * \details avec le bouton, ouvre la fenêtre dialogue vehicule pour creer
le vehicule
*/
void MainWindow::on_pb_creerVehicule_released()
{
    if(pDataBase != nullptr ) {
        CDialoVehicule *vehiculeDialog = new CDialoVehicule(this);
        vehiculeDialog->setNomBouton(ui->pb_creerVehicule->text());
        CVehicule *vehicule = new CVehicule(this);
        vehiculeDialog->setVehicule(vehicule);
        DialogueConfigVehicule *dialogueConfVehicule = new
DialogueConfigVehicule(vehiculeDialog,
this);
        int ret = dialogueConfVehicule->exec();

        if (ret == QDialog::Accepted) {
            //récupération tous les parametre du vehicule_dialog a la
methode CVehicule
            CVehicule *newVehicule = vehiculeDialog->getVehicule();
            //ajout dans la bdd
            pVehiculeManager->addVehicule(*newVehicule);
            //maj tabwidget
            listerLesVehicules();
            delete newVehicule;
        }
        delete vehiculeDialog;
        delete dialogueConfVehicule;

    } else
        QMessageBox::information(this, "Ajouter vehicule", "Vous devez
ajouter une base de donnée");
    }

/*!
 * \brief MainWindow::on_pb_modifierVehicule_released
 * \details avec le bouton, ouvre la fenêtre dialogue vehicule pour
modifier le vehicule
*/
void MainWindow::on_pb_modifierVehicule_released()
{
    if (ui->lw_listeVehicule->currentRow() > -1) {
        QString nomSelect = ui->lw_listeVehicule->currentItem()-
>text(); //selec le nom
        CVehicule *ancienVehicule = pVehiculeManager-
>getVehiculeByNom(nomSelect); //prends les parametres du nom selec
        //envoie tous les parametre du vehicule et le nom du bouton selec
a la methode CDialoVehicule
        CDialoVehicule *vehiculeDialog = new CDialoVehicule(this);
        vehiculeDialog->setNomBouton(ui->pb_modifierVehicule->text());
        vehiculeDialog->setVehicule(ancienVehicule);
        //envoie tous les parametre du vehicule_dialog a la methode
CDialogueConfVehicule
        DialogueConfigVehicule *dialogueConfVehicule = new
DialogueConfigVehicule(vehiculeDialog, this);

        int ret = dialogueConfVehicule->exec();

        if (ret == QDialog::Accepted) {
            //récupération tous les parametre du vehicule_dialog a la
methode CVehicule
            CVehicule *newVehicule = vehiculeDialog->getVehicule();
            //modification vehicule ds bdd
        }
    }
}

```

```

        pVehiculeManager->upDate(nomSelect, *newVehicule); //envoi
pour maj la bdd
    //maj tabwidget
    listerLesVehicules();
    delete newVehicule;
}
delete vehiculeDialog;
delete dialogueConfVehicule;

} else
    QMessageBox::information(this,
                            "Modification véhicule",
                            "Vous devez sélectionner un véhicule");
}

/*!
* \brief MainWindow::listerLesVehicules
* \details Listes les véhicules dans listes des véhicule
*/
void MainWindow::listerLesVehicules()
{
    ui->lw_listeVehicule->clear();
    ui->le_nomDuVehicule->clear();
    ui->le_nombresEssieux->clear();
    ui->le_typeDeVehicule->clear();
    ui->tw_empattement->clear();
    std::vector<CVehicule *> listeVehicule = pVehiculeManager-
>listeVehicule();
    if (listeVehicule.size() > 0) {
        for (CVehicule *vehicule : listeVehicule) {
            ui->lw_listeVehicule->addItem(vehicule->getNom());
        }
    } else {
        QMessageBox::information(this, "Listes des véhicules", "La table
n'a aucun véhicule");
    }
}

/*!
* \brief MainWindow::on_lw_listeVehicule_itemSelectionChanged
* \details Affiche les infos de la voiture sélectionnée
*/
void MainWindow::on_lw_listeVehicule_itemSelectionChanged()
{
    if (ui->lw_listeVehicule->currentRow() > -1) {
        QString nomSelect = ui->lw_listeVehicule->currentItem()->text();
        //récupération des infos du badge dont le nom est sélectionné
        CVehicule *vehicule = pVehiculeManager-
>getVehiculeByNom(nomSelect);
        //initialisation des différents dans le groupe box
        "Caractéristique du véhicule"
        if (vehicule != nullptr) {
            ui->le_nomDuVehicule->setText(vehicule->getNom());
            ui->le_typeDeVehicule->setText(vehicule-
>verifierTypeVehicule(vehicule->getType()));
            ui->le_nombresEssieux->setText(QString::number(vehicule-
>getEssieux()));
            //maj tableWidget
            QStringList titre;
            QString inter_roues;
            ui->tw_empattement->setEditTriggers(QAbstractItemView
::NoEditTriggers); //verouille tab;
    }
}

```

```

        ui->tw_empatement->setRowCount(1); //creer une ligne
        ui->tw_empatement->setColumnCount(vehicule->getEssieux() -
1); //creer les colonnes en fonction des essieux
        for (int i = 0;i < vehicule->getEssieux() - 1 ; i++) {
//boucle qui tourne pendant x fois le nombre essieux -1
            QTableWidgetItem *empatement = new
QTableWidgetItem(QString::number(vehicule->getEmpatement()[i])+" m");
//creer une table pour empatement
            empatement->setTextAlignment(Qt::AlignHCenter);//alligne
au centre
            titre += inter_roues = QString("%1 inter_roues").arg(i +
1); //créer le titre
            ui->tw_empatement->setItem(0, i, empatement);//ajoute les
valeurs dans les tables
        }
        ui->tw_empatement->setHorizontalHeaderLabels(titre);//ajoute
les titres
        delete (vehicule);
    }
} else if (ui->lw_listeVehicule->count() > 0)
    ui->lw_listeVehicule->setCurrentRow(0);
}
/*!
* \brief MainWindow::on_pb_supprimerVehicule_released
* \details à partir du bouton, supprime un vehicule
*/
void MainWindow::on_pb_supprimerVehicule_released()
{
    if (ui->lw_listeVehicule->currentRow() > -1)
    {
        QString nomSelect = ui->lw_listeVehicule->currentItem()->text();
        int ret = QMessageBox::warning(this,
                                      "Suppression vehicule",
                                      "Voulez-vous vraiment supprimer le
vehicule : "+nomSelect,
                                      QMessageBox::Cancel,
                                      QMessageBox::Ok);
        if (ret == QMessageBox::Ok) {
            pVehiculeManager->delVehicule(nomSelect);
            listerLesVehicules();
            on_lw_listeVehicule_itemSelectionChanged();
        }
    } else
        QMessageBox::information(this, "Suppression vehicule", "Vous
devez sélectionner au moins un vehicule");
}

//Page Scenario

/*!
* \brief MainWindow::afficheScenario
* \details Affiche les scenarios
*/
void MainWindow::listerScenario()
{
    ui->lw_listeScenarios->setCurrentRow(-1);
    ui->lw_listeDesScenarios->setCurrentRow(-1);

    ui->lw_listeScenarios->clear();
    ui->lw_listeDesScenarios->clear();
    ui->tw_listePassage->clear();
}

```

```

    std::vector<CScenario *> listeScenario = pScenarioManager-
>listeScenarios();
    if (listeScenario.size() > 0) {
        for (CScenario *scenario : listeScenario) {
            ui->lw_listeScenarios->addItem(scenario->getNom());
            ui->lw_listeDesScenarios->addItem(scenario->getNom());
        }
    } else {
        QMessageBox::information(this, "Listes des scenarios", "La table
n'a aucun scenarios");
    }
}

/*!
* \brief MainWindow::on_lw_scenarios_itemSelectionChanged
* \details Affiche les passages d'un scenario selectionné
*/
void MainWindow::on_lw_listeScenarios_itemSelectionChanged()
{
    if (ui->lw_listeScenarios->currentRow() > -1) {
        ui->tw_listePassage->clear();
        QString nomSelec = ui->lw_listeScenarios->currentItem()->text();
        //récupération des infos du scenario dont le nom est sélectionné
        CScenario *Scenario = pScenarioManager-
>getScenarioByNom(nomSelec);
        //initialisation des différents dans le groupe box
"listesPassage"
        if(Scenario != nullptr) {
            //maj tableWidget
            QStringList titre;
            titre << "Nom" << "Vitesse" << "Temps inter-vehicule";
            ui->tw_listePassage->setEditTriggers(QAbstractItemView
::NoEditTriggers); //verouille tab;
            ui->tw_listePassage->setRowCount(Scenario-
>getVectPassage().size()); //creer des lignes en fonctions de
passages
            ui->tw_listePassage->setColumnCount(3);
            //ajout dans les tableaux
            for (long unsigned i = 0; i < Scenario-
>getVectPassage().size(); i++) {
                QTableWidgetItem *vehicule = new
QTableWidgetItem(Scenario->getVectPassage()[i]->getVehicule()->getNom());
                QTableWidgetItem *vitesse = new
QTableWidgetItem(QString::number(Scenario->getVectPassage()[i]-
>getVitesse()));
                QTableWidgetItem *interVehicule = new
QTableWidgetItem(QString::number(Scenario->getVectPassage()[i]-
>getInterVehicle()));
                vehicule->setTextAlignment(Qt::AlignHCenter);
                vitesse->setTextAlignment(Qt::AlignHCenter);
                interVehicule->setTextAlignment(Qt::AlignHCenter);
                ui->tw_listePassage->setItem(i,0,vehicule);
                ui->tw_listePassage->setItem(i,1,vitesse);
                ui->tw_listePassage->setItem(i,2,interVehicule);
            }
            ui->tw_listePassage->setHorizontalHeaderLabels(titre);
            delete (Scenario);
        }
    }
}
/*!
* \brief MainWindow::on_pb_CreerSCenario_released
* \details avec le bouton, ouvre la fénêtre dialoge scenario pour creer
un scenario

```

```

/*
void MainWindow::on_pb_CreerSCenario_released()
{
    if (pDataBase != nullptr) {
        CDialoScenario *scenarioDialog = new CDialoScenario(this);
        scenarioDialog->setNomBouton(ui->pb_CreerSCenario->text());
        std::vector<QString> listVehicule;
        for (int i = 0; i < ui->lw_listeVehicule->count(); i++) {
            listVehicule.push_back(ui->lw_listeVehicule->item(i)->text());
        }
        CScenario *scenario = new CScenario(this);
        scenarioDialog->setScenario(scenario);
        scenarioDialog->setVehiculeDispo(listVehicule);
        DialogueConfigScenario *scenarioDialogueConfScenario = new
        DialogueConfigScenario(
            scenarioDialog);
        int ret = scenarioDialogueConfScenario->exec();

        if (ret == QDialog::Accepted) {
            //recupération de l'object scenario
            CScenario *newScenario = scenarioDialog->getScenario();
            //ajout scenario ds la bdd
            pScenarioManager->addScenario(*newScenario);
            //actualisation lw_scenario
            listerScenario();
            delete newScenario;
        }
        delete scenario;
        delete scenarioDialog;
        delete scenarioDialogueConfScenario;
    } else
        QMessageBox::information(this, "Ajouter scenario", "Vous devez
ajouter une base de donnée");
}

/*
 * \brief MainWindow::on_pb_modifierScenario_released
 * \details avec le bouton, ouvre la fenêtre dialogue scenario pour creer
un scenario
 */
void MainWindow::on_pb_modifierScenario_released()
{
    if(ui->lw_listeScenarios->currentRow() > -1){
        QString nomSelect = ui->lw_listeScenarios->currentItem()->text(); //selec le nom
        CScenario *ancienScenario = pScenarioManager->getScenarioByNom(nomSelect);
        //recuperation des parametres du scenario selec
        CDialoScenario *scenarioDialog = new CDialoScenario(this);
        scenarioDialog->setNomBouton(ui->pb_modifierScenario->text());
        std::vector<QString> listVehicule;
        for (int i = 0; i < ui->lw_listeVehicule->count(); i++) {
            listVehicule.push_back(ui->lw_listeVehicule->item(i)->text());
        }
        scenarioDialog->setScenario(ancienScenario);
        scenarioDialog->setVehiculeDispo(listVehicule);

        DialogueConfigScenario *scenarioDialogueConfScenario = new
        DialogueConfigScenario(scenarioDialog);

        int ret = scenarioDialogueConfScenario->exec();
        if (ret == QDialog::Accepted) {
}

```

```

        //suppression ancien scenario et ses passages
        pScenarioManager->delScenario(nomSelect);
        //recupération de l'object scenario
        CSscenario *newScenario = scenarioDialog->getScenario();
        //ajout nouveau scenario ds la bdd
        pScenarioManager->addScenario(*newScenario);
        //actualisation lw_scenario
        listerScenario();
        delete newScenario;

    }

    delete scenarioDialog;
    delete scenarioDialogueConfScenario;
}
else
    QMessageBox::information(this,
                           "Modification scénario",
                           "Vous devez sélectionner un scénario");
}

/*!
* \brief MainWindow::on_pb_supprimerScenario_released
* \details à partir du bouton, supprime un scenario et ses passages
*/
void MainWindow::on_pb_supprimerScenario_released()
{
    if (ui->lw_listeScenarios->currentRow() > -1)
    {
        QString nomSelect = ui->lw_listeScenarios->currentItem()->text();
        int ret = QMessageBox::warning(this,
                                      "Suppression scenario",
                                      "Voulez-vous vraiment supprimer le
scenario : "+nomSelect,
                                      QMessageBox::Ok |
                                      QMessageBox::Cancel,
                                      QMessageBox::Ok);
        if (ret == QMessageBox::Ok) {
            //supprime un scenario et ses passages
            pScenarioManager->delScenario(nomSelect);
            //actualise la list widget
            ui->lw_listeScenarios->setCurrentRow(-1);
            listerScenario();
            on_lw_listeScenarios_itemSelectionChanged();

        }
    } else
        QMessageBox::information(this, "Suppression scenario", "Vous
devez sélectionner au moins un scenario") ;
    }

/*!
* \brief MainWindow::on_pb_transfertScenario_released
* \details avec le bouton transfert, envoie les scenarios
*/
void MainWindow::on_pb_transfertScenario_released()
{
    //emplasement json
    pPathJson = mQSettings->value("cheminJSON",pPathJson).toString();
    if (pPathJson != nullptr) {
        if (ui->lw_listeScenarios->currentRow() > -1) {
            QString nomSelec = ui->lw_listeScenarios->currentItem()->text();
}

```

```

        //récupération des infos du scenario dont le nom est
        sélectionné
        CScenario *scenario = pScenarioManager-
>getScenarioByNom(nomSelec);
        //récupération du nom du port est sélectionné
        QString portSelec = ui->cb_portDispo->currentText();
        //etape d'envoie
        if (pScenarioManager->creeJson(pPathJson, scenario) == true)
{ //si json complete :
        ui->lb_etape1->setStyleSheet("background-color: lime ;");
        ui->lb_etape1->setText("Json\na jour");
        //envoie sur port serie
        int envoi = pScenarioManager->transfertJson(pPathJson,
portSelec);
        if (envoi == 1) { //probleme port
            ui->lb_etape2->setText("Port\nd'envoi");
            ui->lb_etape2->setStyleSheet("background-color: red
;");
        } else if (envoi == 2) { // probleme emplacement
            ui->lb_etape2->setText("Emplacement\nnJson");
            ui->lb_etape2->setStyleSheet("background-color: red
;");
        } else if (envoi == 3) { //probleme json vide
            ui->lb_etape1->setStyleSheet("background-color: red
;");
            ui->lb_etape1->setText("Json\nvide");
        } else { //envoie fichier
            ui->lb_etape2->setStyleSheet("background-color: lime
");
            ui->lb_etape2->setText("Json\nenvoyé");
        }
    } else { //probleme emplacement json
        ui->lb_etape1->setStyleSheet("background-color: red;");
        ui->lb_etape1->setText("Emplacement\nnJson");
        ui->lb_etape2->setText("Non\ntraité");
        ui->lb_etape2->setStyleSheet("background-color: red ;");
    }
} else
    QMessageBox::information(this,
                            "Transfere",
                            "Vous devez sélectionner au moins un
scénario");
} else
    QMessageBox::information(this, "Transfere", "Vous devez
sélectionner le fichier json à remplir");
}

/*
* \brief MainWindow::affichePortDisponible
* \details Affiche port disponible cb_portDispo
*/
void MainWindow::affichePortDisponible()
{
    QList<QSerialPortInfo> listePorts;
    QStringList listePortsDisponibles;
    listePortsDisponibles << "Aucun";

    listePorts = QSerialPortInfo::availablePorts();
    for (int i=0; i < listePorts.size();i++)
    {
        QSerialPortInfo info = listePorts.at(i);
        //if(info.portName().contains("ACM") ||
info.portName().contains("USB"))
        {

```

```

        if(!info.manufacturer().isEmpty())
            listePortsDisponibles << /*info.manufacturer() + " (" +
*/info.portName() /*+ ")"*/;
        else
            listePortsDisponibles << info.portName();
    }
}

ui->cb_portDispo->addItems(listePortsDisponibles);
}

//Page Stats

/*!
 * \brief MainWindow::on_lw_listeDesScenarios_itemSelectionChanged
 * \details Affiche les stats d'un scenario selectionné
 */
void MainWindow::on_lw_listeDesScenarios_itemSelectionChanged()
{
    if (ui->lw_listeDesScenarios->currentRow() > -1) {
        QString nomSelec = ui->lw_listeDesScenarios->currentItem()-
>text();
        //récupération des infos du scenario dont le nom est sélectionné
        CScenario *Scenario = pScenarioManager-
>getScenarioByNom(nomSelec);
        //initialisation des différentes stats dans la page IHM stats
        if (Scenario != nullptr) {
            //stats v1
            ui->le_NombresPassesVL->setText(QString::number(Scenario-
>calculerStatsVL()->getNbreVehicule()));
            ui->le_vitesseMoyVL->setText(QString::number(Scenario-
>calculerStatsVL()->getVitesseMoy()));
            ui->le_vitesseMaxVL->setText(QString::number(Scenario-
>calculerStatsVL()->getVitesseMax()));
            //stats pl
            ui->le_NombresPassesPL->setText(QString::number(Scenario-
>calculerStatsPL()->getNbreVehicule()));
            ui->le_VitesseMoyPL->setText(QString::number(Scenario-
>calculerStatsPL()->getVitesseMoy()));
            ui->le_vitesseMaxPL->setText(QString::number(Scenario-
>calculerStatsPL()->getVitesseMax()));
            //stats general
            ui->le_dureeScenario->setText(QString::number(Scenario-
>calculerStatsVL()->getDuree())+Scenario->calculerStatsPL()->getDuree()));
            ui->le_nombresVehiculePasss-
>setText(QString::number(Scenario->calculerStatsVL()-
>getNbreVehicule())+Scenario->calculerStatsPL()->getNbreVehicule()));
            if(Scenario->calculerStatsVL()->getVitesseMoy() > 0 &&
Scenario->calculerStatsPL()->getVitesseMoy() > 0)
                ui->le_vitesseMoy->setText(QString::number((Scenario-
>calculerStatsVL()->getVitesseMoy())+Scenario->calculerStatsPL()-
>getVitesseMoy())/2));
            else
                ui->le_vitesseMoy->setText(QString::number((Scenario-
>calculerStatsVL()->getVitesseMoy())+Scenario->calculerStatsPL()-
>getVitesseMoy())));
        }
    }
}

```

[SigfoxMessageController.php](#)

```
<?php

namespace App\Http\Controllers;

use App\Models\Sigfoxmessage;
use Illuminate\Http\Request;
use App\Statistique;
use App\Campagnedemesure;
use App\Boitier;

class SigfoxMessageController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        return Sigfoxmessage::all();
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        if (!$request->input('device')) {
            return response()->json('ERREUR : champ device vide!', 500);
        }

        $sigfoxMessage = Sigfoxmessage::create([
            'device' => $request->input('device'),
            'data'   => $request->input('data'),
            'time'   => $request->input('time'),
        ]);
    }
}
```

```

]);
```

```

// décode la payload
// conversion de la payload en binaire codé ASCII
$binData = "";
```

```

$info = $sigfoxMessage->data;
if(strlen($sigfoxMessage->data) == 12)
{
```

```

} else {
    for ($i = 0; $i < strlen($sigfoxMessage->data); $i++) {
        // convertit chaque chiffre hexa en un quartet binaire
        $quartet = base_convert($sigfoxMessage->data[$i], 16, 2);
        $quartet = str_pad($quartet, 4, "0", STR_PAD_LEFT); // format fixe en 4
chiffres binaires
        $binData .= $quartet; // on concatène au résultat
    }
}
```

```

$info2 = $binData;
```

```

// extraction des données binaires
$typeV = extraitChampBinaire($binData, 0, 1);
$NbVehicule = extraitChampBinaire($binData, 1, 11);
$NbEssieu = extraitChampBinaire($binData, 12, 13);
$VitMoyenne = extraitChampBinaire($binData, 25, 8);
$VitMax = extraitChampBinaire($binData, 33, 8);
$VitesseInferieureOuEgale = extraitChampBinaire($binData, 41, 11);
$VitesseLimitMoins20 = extraitChampBinaire($binData, 52, 11);
$VitesseLimitPlus20 = extraitChampBinaire($binData, 63, 10);
$VitesseLimitPlus30 = extraitChampBinaire($binData, 73, 9);
$VitesseLimitPlus40 = extraitChampBinaire($binData, 82, 7);
$VitesseLimitPlus50 = extraitChampBinaire($binData, 89, 7); //96
```

```

$typeV = dechex($typeV);
$NbVehicule = dechex($NbVehicule);
$NbEssieu = dechex($NbEssieu);
$VitMoyenne = dechex($VitMoyenne);
$VitMax = dechex($VitMax);
$VitesseInferieureOuEgale = dechex($VitesseInferieureOuEgale);
$VitesseLimitMoins20 = dechex($VitesseLimitMoins20);
$VitesseLimitPlus20 = dechex($VitesseLimitPlus20);
```

```

$VitesseLimitPlus30 = dechex($VitesseLimitPlus30);
$VitesseLimitPlus40 = dechex($VitesseLimitPlus40);
$VitesseLimitPlus50 = dechex($VitesseLimitPlus50);

$boitier = Boitier::where('adrSigfox', $sigfoxMessage->device)->first();

if (empty($boitier)) {
    return response()->json('ERREUR : boitier inexistant', 500);
}
$campagne = Campagnedemesure::where('boitier_id', $boitier->id)->first();
if (empty($campagne)) {
    return response()->json('ERREUR : campagne inexisteante', 500);
}

if ($typeV == 0) {
    $type_vehicule = "voiture";
} else {
    $type_vehicule = "camion";
}
// création du relevé
$stat = Statistique::create([
'campagnId' => $campagne->id,
'typeV' => $type_vehicule,
'timestamp' => date('Y-m-d H:i:s', $sigfoxMessage->time),
'VitMax' => $VitMax,
'NbEssieux' => $NbEssieu,
'VitMoyenne' => $VitMoyenne,
'NbVehicule' => $NbVehicule,
'VitesseInferieureOuEgale' => $VitesseInferieureOuEgale,
'VitesseLimitMoins20' => $VitesseLimitMoins20,
'VitesseLimitPlus20' => $VitesseLimitPlus20,
'VitesseLimitPlus30' => $VitesseLimitPlus30,
'VitesseLimitPlus40' => $VitesseLimitPlus40,
'VitesseLimitPlus50' => $VitesseLimitPlus50,
]);
return (array(['info' => $info, 'info2' => $info2]));
}
}
/** 
 * Display the specified resource.
 *
 * @param \App\Models\Sigfoxmessage $sigfoxmessage

```

```
* @return \Illuminate\Http\Response
*/
public function show($id)
{
    //
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Models\Sigfoxmessage $sigfoxmessage
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, Sigfoxmessage $sigfoxmessage)
{
    //
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\Sigfoxmessage $sigfoxmessage
 * @return \Illuminate\Http\Response
 */
public function destroy(Sigfoxmessage $sigfoxmessage)
{
    //
}
function extraitChampBinaire($chaine, $index, $taille)
{
    $bin = substr($chaine, $index, $taille);
    if (strpos($bin, "0") === false) {
        // aucun zéro trouvé => tous bits à 1 => NULL
        $val = null;
    } else {
        $val = base_convert($bin, 2, 10);
    }
    return $val;
}
```

CollecteSigfoxController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Statistique;

class CollecteSigfoxController extends Controller
{
    public function index()
    {
        $Stats = Statistique::all();
        return view('pages/Statistiques')->with('Stats', $Stats);
    }
}
```

[Web.php](#)

```
<?php

use Illuminate\Support\Facades\Route;

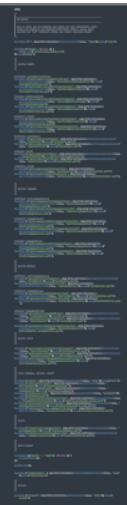
/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/
Route::get('/', function () {
    return view('welcome');
});

Route::group(['prefix' => 'admin'], function () {
    Voyager::routes();
});

Route::get('/Statistiques', [
    'as'=>'Statistiques',
    'uses'=>'App\Http\Controllers\CollecteSigfoxController@index'
]) ;

//Route::get('/Controller', [
//    '/as'=>'Controller',
//    //return view('welcome');
//    '/uses'=>'App\Http\Controllers\Controller@index'
//]) ;

Route::get('/UpdateStats', [
    'as'=>'UpdateStats',
    'uses'=>'App\Http\Controllers\Controller@index'
]);
```

ARMAND HOUPIN	web.php	1/3
<pre> 1 <?php 2 3 /* 4 ----- 5 Web Routes 6 ----- 7 8 Here is where you can register web routes for your application. These 9 routes are loaded by the RouteServiceProvider within a group which 10 contains the "web" middleware group. Now create something great! 11 12 */ 13 14 Route::get('/', [App\Http\Controllers\PagesController::class, 'login'])->name('login'); 15 16 17 Route::get('/popup', function () { 18 return view('/Client/popup_highscore'); 19 })->name('popup'); 20 21 // 22 // 23 // Gestion Compte 24 // 25 // 26 27 28 //Afficher ListeAdminTechnicien 29 Route::get('/gerercompte/ListeAdminTechnicien', [App\Http\Controllers\ gestioncompteController::class, 'listeAdminTechnicienmodifier'])->name(' listeAdminTechnicien_path'); 30 Route::get('/gerercompte>ListeAdminTechnicien/{id}', [App\Http\Controllers\ gestioncompteController::class, 'supprimerAdminTechnicien'])->name(' deleteAdminTechnicien_path'); 31 32 //Ajouter AdminTechnicien 33 Route::get('/gerercompte/inscriptionAdminTechnicien', [App\Http\Controllers\ gestioncompteController::class, 'ajouterAdminTechnicienget'])->name(' inscriptionCompteAdmin'); 34 Route::post('/gerercompte/inscriptionAdminTechnicien', [App\Http\Controllers\ gestioncompteController::class, 'ajouterAdminTechnicien'])->name(' ajouterAdminTechnicien_path'); 35 36 //Modifier Client 37 Route::get('/gerercompte/ModifierAdminTechnicien/{id}', [App\Http\Controllers\ gestioncompteController::class, 'modifierAdminTechnicien'])->name(' modifierAdminTechnicien_path'); 38 Route::post('/gerercompte/ModifierAdminTechnicien/{id}', [App\Http\Controllers\ gestioncompteController::class, 'modifierAdminTechnicienConfirm'])->name(' modifierAdminTechnicien_path'); 39 40 //Afficher ListeClient 41 Route::get('/gerercompte>ListeClient', [App\Http\Controllers\gestioncompteController:: class, 'listeClient'])->name(' listeClient_path'); 42 Route::get('/gerercompte>ListeClient/{id}', [App\Http\Controllers\gestioncompteController:: class, 'supprimerClient'])->name('deleteClient_path'); 43 44 //Ajouter Client 45 Route::get('/gerercompte/inscription', [App\Http\Controllers\gestioncompteController::class , 'ajouterClientget'])->name(' inscriptionCompteClient'); 46 Route::post('/gerercompte/inscription', [App\Http\Controllers\gestioncompteController:: class, 'ajouterClient'])->name('ajouterClient_path'); 47 48 //Modifier Client 49 Route::get('/gerercompte/ModifierClient/{id}', [App\Http\Controllers\ gestioncompteController::class, 'modifierClient'])->name('modifierClient_path'); 50 Route::post('/gerercompte/ModifierClient/{id}', [App\Http\Controllers\ gestioncompteController::class, 'modifierClientConfirm'])->name('modifierClient_path'); 51 </pre>		

ARMAND HOUPIN	web.php	2/3
<pre> 53 // 54 // 55 // Gestion Campagne 56 // 57 // 58 59 //Afficher ListeCampagneMesure 60 Route::get('/gerercampagne/ListeCampagneMesure', [App\Http\Controllers\gestioncampagnemesureController::class, 'listeCampagneMesure'])->name('listeCampagneMesure_path'); 61 Route::get('/gerercampagne>ListeCampagneMesure/{id}', [App\Http\Controllers\gestioncampagnemesureController::class, 'supprimerCampagneMesure'])->name('deleteCampagneMesure_path'); 62 63 //Modifier CampagneMesure 64 Route::get('/gerercampagne/modifierCampagneMesure/{id}', [App\Http\Controllers\gestioncampagnemesureController::class, 'modifierCampagneMesure'])->name('modifierCampagneMesure_path'); 65 Route::post('/gerercampagne/modifierCampagneMesure/{id}', [App\Http\Controllers\gestioncampagnemesureController::class, 'modifierCampagneMesureConfirm'])->name('modifierCampagneMesure_path'); 66 67 68 //Ajouter CampagneMesure 69 Route::get('/gerercampagne/creationCampagneMesure', [App\Http\Controllers\gestioncampagnemesureController::class, 'creationCampagneMesure'])->name('creationCampagneMesure_path'); 70 Route::post('/gerercampagne/creationCampagneMesure', [App\Http\Controllers\gestioncampagnemesureController::class, 'ajouterCampagneMesure'])->name('creationCampagneMesure_path'); 71 72 // 73 // 74 // Gestion Boitier 75 // 76 // 77 78 //Afficher ListeCampagneMesure 79 Route::get('/gererboitier>ListeBoitier', [App\Http\Controllers\gestionboitierController::class, 'listeBoitier'])->name('listeBoitier_path'); 80 Route::get('/gererboitier>ListeBoitier/{id}', [App\Http\Controllers\gestionboitierController::class, 'supprimerBoitier'])->name('deleteBoitier_path'); 81 82 //Modifier CampagneMesure 83 Route::get('/gererboitier/modifierBoitier/{id}', [App\Http\Controllers\gestionboitierController::class, 'modifierBoitier'])->name('modifierBoitier_path'); 84 Route::post('/gererboitier/modifierBoitier/{id}', [App\Http\Controllers\gestionboitierController::class, 'modifierBoitierConfirm'])->name('modifierBoitier_path'); 85 86 87 //Ajouter CampagneBoitier 88 Route::get('/gererboitier/creationBoitier', [App\Http\Controllers\gestionboitierController::class, 'creationBoitier'])->name('creationBoitier_path'); 89 Route::post('/gererboitier/creationBoitier', [App\Http\Controllers\gestionboitierController::class, 'ajouterBoitier'])->name('creationBoitier_path'); 90 91 Route::get('/gererboitier/BoitierAlarmeBatterie{id}', [App\Http\Controllers\gestionboitierController::class, 'BoitierAlarmeBatterie'])->name('modifierBoitier_alarmebatterie_path'); 92 // 93 // 94 // Gestion Alert 95 // 96 // 97 98 Route::get('/gerercompte>ListeClient/alert/{alert}', [App\Http\Controllers\AlertController::class, 'listeClientalert'])->name('listeClientalert_path'); 99 Route::get('/gerercompte>ListeBoitier/alert/{alert}', [App\Http\Controllers\AlertController::class, 'listeboitieralert'])->name('listeboitieralert_path'); 100 Route::get('/gerercompte>ListeCampagneMesure/alert/{alert}', [App\Http\Controllers\AlertController::class, 'listecampagnemesurealert'])->name('listecampagnemesurealert_path'); 101 102 // 103 // 104 // Liste campagne, boitier, client 105 // 106 // </pre>		

E7 : HOUPIN Armand

ARMAND HOUPIN	web.php	3/3
<pre> 107 Route::get('/liste', [App\Http\Controllers\HomeController::class, 'liste'])->name('liste'); 108 Route::get('/liste/boitier/{id}', [App\Http\Controllers\HomeController::class, 'boitierdelete'])->name('listedeleteboitier'); 109 Route::get('/liste/campagne/{id}', [App\Http\Controllers\HomeController::class, 'campagnedelete'])->name('listedeletecampagne'); 110 Route::get('/liste/user/{id}', [App\Http\Controllers\HomeController::class, 'userdelete']) ->name('listedeleteuser'); 111 Route::get('/liste/modifierStatutencours/{id}', [App\Http\Controllers\HomeController::class, 'listemodifierstatutEncours'])->name('modifierListestatut_encours_path'); 112 Route::get('/liste/modifierStatutfini/{id}', [App\Http\Controllers\HomeController::class, 'listemodifierstatutfini'])->name('modifierListestatut_fini_path'); 113 Route::get('/liste/ajouter/{id_campagne}/{id_boitier}', [App\Http\Controllers\ HomeController::class, 'listeaajouterboitier'])->name('listeaajouterboitier_path'); 114 // 115 // 116 // Client 117 // 118 // 119 // 120 Route::get('/client/listecampagneclient', [App\Http\Controllers\clientController::class, 'mescampagnesliste'])->name('clientmescampagnes_path'); 121 Route::get('/client/consultationcampagne/{id}', [App\Http\Controllers\clientController:: class, 'campagneconsultation'])->name('clientconsultation_path'); 122 // 123 // 124 // Voyager 125 // 126 // 127 // 128 129 Route::group(['prefix' => 'admin'], function () { 130 Voyager::routes(); 131 }); 132 133 Auth::routes(); 134 135 136 Route::get('/Statistiques', [App\Http\Controllers\CollecteSigfoxController::class, 'index'])->name('Statistiques'); 137 // 138 // 139 // Accueil 140 // 141 // 142 // 143 144 Route::get('/accueil', [App\Http\Controllers\HomeController::class, 'index'])->name(' accueil'); </pre>		

ARMAND HOUPIN	boitier.php	1/1
<pre> 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 class boitier extends Model 8 { 9 protected \$fillable = [10 'adrSigfox', 11 'alarmeBatterie', 12 'statut' 13]; 14 } </pre>		

E7 : HOUPIN Armand

ARMAND HOUPIN	campagne.php	1/1
<pre> 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 class campagnemesure extends Model 8 { 9 protected \$fillable = [10 'limitationvitesse', 11 'id_user', 12 'DebutCampagne', 13 'FinCampagne', 14 'adresseCampagne', 15 'latitudecampagne', 16 'longitudeCampagne', 17 'statut', 18 'codePostal', 19 'ville', 20 'numeroRoute', 21 'Direction' 22]; 23 }</pre>		

ARMAND HOUPIN	statistique.php	1/1
<pre> 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Database\Eloquent\Model; 6 7 8 class Statistique extends Model 9 { 10 protected \$fillable = [11 'typeV', 12 'VitMax', 13 'NbEssieux', 14 'VitMoyenne', 15 'NbVehicule', 16 'VitesseInferieureOuEgale', 17 'VitesseLimitMoins20', 18 'VitesseLimitPlus20', 19 'VitesseLimitPlus30', 20 'VitesseLimitPlus40', 21 'VitesseLimitPlus50', 22 'campagneId' 23]; 24 }</pre>		

ARMAND HOUPIN	User.php	1/1
<pre> 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Contracts\Auth\MustVerifyEmail; 6 use Illuminate\Database\Eloquent\Factories\HasFactory; 7 use Illuminate\Foundation\Auth\User as Authenticatable; 8 use Illuminate\Notifications\Notifiable; 9 10 class User extends \TCG\Voyager\Models\User 11 { 12 use HasFactory, Notifiable; 13 14 /** 15 * The attributes that are mass assignable. 16 * 17 * @var array 18 */ 19 protected \$fillable = [20 'name', 21 'email', 22 'role_id', 23 'password', 24 'prenom', 25 'identifiant', 26 'numeroTel', 27 'adresseClient', 28 'codePostal', 29 'ville', 30 'region', 31 'entreprise', 32 'nomEntreprise', 33 'preference' 34]; 35 36 /** 37 * The attributes that should be hidden for arrays. 38 * 39 * @var array 40 */ 41 protected \$hidden = [42 'password', 43 'remember_token', 44]; 45 46 /** 47 * The attributes that should be cast to native types. 48 * 49 * @var array 50 */ 51 protected \$casts = [52 'email_verified_at' => 'datetime', 53]; 54 } 55 </pre>		

ARMAND HOUPIN	alertController.php	1/2
<pre> 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 use App\Models\User; 8 use App\Models\campagnemesure; 9 use App\Models\boitier; 10 11 class AlertController extends Controller 12 { 13 public function listeClientalert(\$alert) 14 { 15 \$listeclient = User::all(); 16 17 if(\$alert == 1) 18 { 19 \$alert = 1; 20 \$messagealert = "Le compte client a bien été crée !"; 21 } else if(\$alert == 2) 22 { 23 \$alert = 1; 24 \$messagealert = "Le compte client a bien été modifiée !"; 25 } else if(\$alert == 3) 26 { 27 \$alert = 1; 28 \$messagealert = "Le compte client a bien été supprimée !"; 29 } 30 return view('/gerercompte/ListeClient')->with('client', \$listeclient)->with('alert', 31 \$alert)->with('messagealert', \$messagealert); 32 } 33 34 public function listeboitieralert(\$alert) 35 { 36 \$listebotier = boitier::all(); 37 38 if(\$alert == 1) 39 { 40 \$alert = 1; 41 \$messagealert = "Le boitier a bien été crée !"; 42 } else if(\$alert == 2) 43 { 44 \$alert = 1; 45 \$messagealert = "Le boitier a bien été modifiée !"; 46 } else if(\$alert == 3) 47 { 48 \$alert = 1; 49 \$messagealert = "Le boitier a bien été supprimée !"; 50 } else if(\$alert == 5) 51 { 52 \$alert = 1; 53 \$messagealert = "L'alarme de la batterie à bien été modifié"; 54 } 55 56 57 return view('/gererboitier>ListeBoitier')->with('boitier', \$listebotier)->with(58 'alert', \$alert)->with('messagealert', \$messagealert); 59 } </pre>		

ARMAND HOUPIN	alertController.php	2/2
<pre>60 public function listecampagnemesurealert(\$alert) 61 { 62 \$listecampagne = campagnemesure::all(); 63 64 65 if(\$alert == 1) 66 { 67 \$alert = 1; 68 \$messagealert = "La campagne de mesure a bien été créée !"; 69 } else if(\$alert == 2) 70 { 71 \$alert = 1; 72 \$messagealert = "La campagne de mesure a bien été modifiée !"; 73 } else if(\$alert == 3) 74 { 75 \$alert = 1; 76 \$messagealert = "La campagne de mesure a bien été supprimée !"; 77 } 78 79 return view('/gerercampagne>ListeCampagneMesure')->with('campagne', \$listecampagne) 80 ->with('alert', \$alert)->with('messagealert', \$messagealert); 81 } 82 83 public function demandeOK_path(\$alert) 84 { 85 \$alert = 1; 86 \$messagealert = "La demande a été envoyée !"; 87 return view('/forms/message_demande')->with('alert', \$alert)->with('messagealert', \$messagealert); 88 } 89 } 90 }</pre>		

ARMAND HOUPIN	clientController.php	1/3
<pre> 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 use Illuminate\Notifications\Notifiable; 8 use TCG\Voyager\Traits\VoyagerUser; 9 use Illuminate\Foundation\Auth\User as Authenticatable; 10 use Illuminate\Support\Facades\Auth; 11 12 use App\Models\User; 13 use App\Models\campagnemesure; 14 use App\Models\boitier; 15 use App\Models\Statistique; 16 17 use App\Models\demande; 18 19 use Illuminate\Support\Facades\Mail; 20 use App\Mail\MessageGoogle; 21 22 23 class clientController extends Controller 24 { 25 public function mescampagnesliste() 26 { 27 28 if (! auth()->check()) { 29 return redirect('/login'); 30 } 31 32 \$listecampagne = campagnemesure::all(); 33 34 return view('/Client/listecampagneclient')->with('listecampagne', \$listecampagne); 35 } 36 37 public function campagneconsultation(\$id) 38 { 39 if (! auth()->check()) { 40 return redirect('/login'); 41 } 42 43 \$campagne = campagnemesure::find(\$id); 44 \$stat = Statistique::all(); 45 46 \$nom_file = 'resultmesure.csv'; 47 48 if(file_exists('csv/' . \$nom_file) == 1) 49 { 50 unlink('csv/' . \$nom_file); 51 } 52 53 \$vit_m_20 = \$campagne->limitationvitesse - 20; 54 \$vit_p_20 = \$campagne->limitationvitesse + 20; 55 \$vit_p_30 = \$campagne->limitationvitesse + 30; 56 \$vit_p_40 = \$campagne->limitationvitesse + 40; 57 \$vit_p_50 = \$campagne->limitationvitesse + 50; 58 59 \$f = fopen('csv/' . \$nom_file, "x+"); 60 \$texte = 'Type de véhicule ; Heure ; Limité à ' . 'km/h;Nombre de véhicules qui 61 sont passés ;La vitesse Moyenne des véhicules ;Nombre de véhicules 62 inférieure ou égale à ' . \$campagne->limitationvitesse . ' km/h ;Nombre de 63 véhicules inférieure à ' . \$vit_m_20 . ' km/h ;Nombre de véhicules supérieur à 64 ' . \$vit_p_20 . ' km/h ;Nombre de véhicules supérieur à ' . \$vit_p_30 . ' km/h 65 ;Nombre de véhicules supérieur à ' . \$vit_p_40 . ' km/h ;Nombre de véhicules 66 supérieur à ' . \$vit_p_50 . ' km/h :'. "\n"; 67 // écriture 68 fputs(\$f, \$texte); 69 70 } 71 72 } </pre>		

ARMAND HOUPIN	clientController.php	2/3
<pre> 64 foreach(\$stat as \$data){ 65 if(\$data->campagneId == \$campagne->id) 66 \$texte = \$data->typeV . ';' . \$data->created_at . ';' . \$campagne-> 67 limitationvitesse . ';' . \$data->NbVehicule . ';' . \$data->VitMoyenne . ';' . \$ 68 data->VitesseInferieureOuEgale . ';' . \$data->VitesseLimitMoins20 . ';' . \$data- 69 ->VitesseLimitPlus20 . ';' . \$data->VitesseLimitPlus30 . ';' . \$data-> 70 VitesseLimitPlus40 . ';' . \$data->VitesseLimitPlus50 . "\n"; 71 fputs(\$f, \$texte); 72 } 73 // fermeture 74 fclose(\$f); 75 76 \$vitmax = 0; 77 78 foreach(\$stat as \$data) 79 { 80 if(\$data->VitMax > \$vitmax) 81 { 82 \$vitmax = \$data->VitMax; 83 } 84 } 85 86 \$stat_true = 0; 87 88 foreach(\$stat as \$data){ 89 if(\$data->campagneId == \$campagne->id) 90 { 91 if(\$data->typeV == "camion") 92 { 93 \$statVitMoyen_PL[] = \$data->VitMoyenne; 94 \$statHeureStat_PL[] = \$data->created_at; 95 \$statVitesseInferieurOuEgale_PL[] = \$data->VitesseInferieureOuEgale; 96 \$statVitesseLimitMoins20_PL[] = \$data->VitesseLimitMoins20; 97 \$statVitesseLimitPlus20_PL[] = \$data->VitesseLimitPlus20; 98 \$statVitesseLimitPlus30_PL[] = \$data->VitesseLimitPlus30; 99 \$statVitesseLimitPlus40_PL[] = \$data->VitesseLimitPlus40; 100 \$statVitesseLimitPlus50_PL[] = \$data->VitesseLimitPlus50; 101 } elseif(\$data->typeV == "voiture") 102 { 103 \$statVitMoyen_VL[] = \$data->VitMoyenne; 104 \$statHeureStat_VL[] = \$data->created_at; 105 \$statVitesseInferieurOuEgale_VL[] = \$data->VitesseInferieureOuEgale; 106 \$statVitesseLimitMoins20_VL[] = \$data->VitesseLimitMoins20; 107 \$statVitesseLimitPlus20_VL[] = \$data->VitesseLimitPlus20; 108 \$statVitesseLimitPlus30_VL[] = \$data->VitesseLimitPlus30; 109 \$statVitesseLimitPlus40_VL[] = \$data->VitesseLimitPlus40; 110 \$statVitesseLimitPlus50_VL[] = \$data->VitesseLimitPlus50; 111 } 112 \$stat_true = 1; 113 } else 114 { 115 \$stat_true = 0; 116 } 117 118 if(\$stat_true == 1) 119 { 120 if(empty(\$statVitMoyen_PL)){ 121 122 \$statVitMoyen_PL = NULL; 123 \$statHeureStat_PL = NULL; 124 \$statVitesseInferieurOuEgale_PL = NULL; 125 \$statVitesseLimitMoins20_PL = NULL; 126 \$statVitesseLimitPlus20_PL = NULL; 127 \$statVitesseLimitPlus30_PL = NULL; 128 \$statVitesseLimitPlus40_PL = NULL; 129 \$statVitesseLimitPlus50_PL = NULL; 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 389 390 391 392 393 394 395 396 397 398 399 399 400 401 402 403 404 405 406 407 408 409 409 410 411 412 413 414 415 416 417 418 419 419 420 421 422 423 424 425 426 427 428 429 429 430 431 432 433 434 435 436 437 438 439 439 440 441 442 443 444 445 446 447 448 449 449 450 451 452 453 454 455 456 457 458 459 459 460 461 462 463 464 465 466 467 468 469 469 470 471 472 473 474 475 476 477 478 479 479 480 481 482 483 484 485 486 487 488 489 489 490 491 492 493 494 495 496 497 498 499 499 500 501 502 503 504 505 506 507 508 509 509 510 511 512 513 514 515 516 517 518 519 519 520 521 522 523 524 525 526 527 528 529 529 530 531 532 533 534 535 536 537 538 539 539 540 541 542 543 544 545 546 547 548 549 549 550 551 552 553 554 555 556 557 558 559 559 560 561 562 563 564 565 566 567 568 569 569 570 571 572 573 574 575 576 577 578 579 579 580 581 582 583 584 585 586 587 588 589 589 590 591 592 593 594 595 596 597 598 599 599 600 601 602 603 604 605 606 607 608 609 609 610 611 612 613 614 615 616 617 618 619 619 620 621 622 623 624 625 626 627 628 629 629 630 631 632 633 634 635 636 637 638 639 639 640 641 642 643 644 645 646 647 648 649 649 650 651 652 653 654 655 656 657 658 659 659 660 661 662 663 664 665 666 667 668 669 669 670 671 672 673 674 675 676 677 678 679 679 680 681 682 683 684 685 686 687 687 688 689 689 690 691 692 693 694 695 696 697 697 698 699 699 700 701 702 703 704 705 706 707 708 709 709 710 711 712 713 714 715 716 717 717 718 719 719 720 721 722 723 724 725 726 727 728 729 729 730 731 732 733 734 735 736 737 738 739 739 740 741 742 743 744 745 746 747 748 749 749 750 751 752 753 754 755 756 757 758 759 759 760 761 762 763 764 765 766 767 768 769 769 770 771 772 773 774 775 776 777 778 779 779 780 781 782 783 784 785 786 787 787 788 789 789 790 791 792 793 794 795 796 797 797 798 799 799 800 801 802 803 804 805 806 807 808 809 809 810 811 812 813 814 815 816 817 817 818 819 819 820 821 822 823 824 825 826 827 827 828 829 829 830 831 832 833 834 835 836 837 837 838 839 839 840 841 842 843 844 845 846 847 847 848 849 849 850 851 852 853 854 855 856 857 857 858 859 859 860 861 862 863 864 865 866 867 867 868 869 869 870 871 872 873 874 875 876 876 877 878 878 879 879 880 881 882 883 884 885 886 887 887 888 889 889 890 891 892 893 894 895 896 897 897 898 899 899 900 901 902 903 904 905 906 907 908 909 909 910 911 912 913 914 915 916 917 917 918 919 919 920 921 922 923 924 925 926 927 927 928 929 929 930 931 932 933 934 935 936 937 937 938 939 939 940 941 942 943 944 945 946 946 947 948 948 949 950 951 952 953 954 955 956 957 957 958 959 959 960 961 962 963 964 965 966 966 967 968 968 969 969 970 971 972 973 974 975 976 976 977 978 978 979 979 980 981 982 983 984 985 986 986 987 988 988 989 989 990 991 992 993 994 995 996 996 997 998 998 999 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1009 1010 1011 1012 1013 1014 1015 1016 1017 1017 1018 1019 1019 1020 1021 1022 1023 1024 1025 1026 1027 1027 1028 1029 1029 1030 1031 1032 1033 1034 1035 1036 1037 1037 1038 1039 1039 1040 1041 1042 1043 1044 1045 1046 1046 1047 1048 1048 1049 1049 1050 1051 1052 1053 1054 1055 1056 1057 1057 1058 1059 1059 1060 1061 1062 1063 1064 1065 1066 1066 1067 1068 1068 1069 1069 1070 1071 1072 1073 1074 1075 1076 1076 1077 1078 1078 1079 1079 1080 1081 1082 1083 1084 1085 1086 1086 1087 1088 1088 1089 1089 1090 1091 1092 1093 1094 1095 1095 1096 1097 1097 1098 1098 1099 1099 1100 1101 1102 1103 1104 1105 1106 1106 1107 1108 1108 1109 1109 1110 1111 1112 1113 1114 1114 1115 1116 1116 1117 1117 1118 1119 1119 1120 1120 1121 1122 1122 1123 1123 1124 1124 1125 1125 1126 1126 1127 1127 1128 1128 1129 1129 1130 1131 1132 1133 1134 1135 1136 1137 1137 1138 1139 1139 1140 1141 1142 1143 1144 1144 1145 1146 1146 1147 1147 1148 1149 1149 1150 1151 1152 1153 1154 1155 1156 1156 1157 1158 1158 1159 1159 1160 1161 1162 1163 1164 1165 1165 1166 1167 1167 1168 1168 1169 1169 1170 1171 1172 1173 1174 1175 1175 1176 1177 1177 1178 1178 1179 1179 1180 1181 1182 1183 1184 1185 1185 1186 1187 1187 1188 1188 1189 1189 1190 1191 1192 1193 1194 1194 1195 1196 1196 1197 1197 1198 1199 1199 1200 1201 1202 1203 1204 1204 1205 1206 1206 1207 1207 1208 1209 1209 1210 1210 1211 1212 1213 1214 1214 1215 1216 1216 1217 1217 1218 1219 1219 1220 1220 1221 1222 1223 1224 1224 1225 1226 1226 1227 1227 1228 1229 1229 1230 1231 1232 1233 1234 1235 1235 1236 1237 1237 1238 1238 1239 1239 1240 1241 1242 1243 1244 1244 1245 1246 1246 1247 1247 1248 1249 1249 1250 1251 1252 1253 1254 1255 1255 1256 1257 1257 1258 1258 1259 1260 1261 1262 1263 1264 1264 1265 1266 1266 1267 1267 1268 1269 1269 1270 1271 1272 1273 1274 1275 1275 1276 1277 1277 1278 1278 1279 1280 1281 1282 1283 1284 1284 1285 1286 1286 1287 1287 1288 1289 1289 1290 1291 1292 1293 1294 1294 1295 1296 1296 1297 1297 1298 1299 1299 1300 1301 1302 1303 1304 1304 1305 1306 1306 1307 1307 1308 1309 1309 1310 1311 1312 1313 1314 1314 1315 1316 1316 1317 1317 1318 1319 1319 1320 1321 1322 1323 1324 1324 1325 1326 1326 1327 1327 1328 1329 1329 1330 1331 1332 1333 1334 1335 1335 1336 1337 1337 1338 1338 1339 1340 1340 1341 1342 1343 1344 1345 1345 1346 1347 1347 1348 1348 1349 1350 1351 1352 1353 1354 1354 1355 1356 1356 1357 1357 1358 1359 1359 1360 1361 1362 1363 1364 1365 1365 1366 1367 1367 1368 1368 1369 1370 1370 1371 1372 1373 1374 1375 1375 1376 1377 1377 1378 1378 1379 1380 1381 1382 1383 1384 1384 1385 1386 1386 1387 1387 1388 1389 1389 1390 1391 1392 1393 1394 1394 1395 1396 1396 1397 1397 1398 1399 1399 1400 1401 1402 1403 1404 1404 1405 1406 1406 1407 1407 1408 1409 1409 1410 1411 1412 1413 1414 1414 1415 1416 1416 1417 1417 1418 1419 1419 1420 1421 1422 1423 1424 1424 1425 1426 1426 1427 1427 1428 1429 1429 1430 1431 1432 1433 1434 1435 1435 1436 1437 1437 1438 1438 1439 1440 1440 1441 1442 1443 1444 1445 1445 1446 1447 1447 1448 1448 1449 1450 1451 1452 1453 1454 1454 1455 1456 1456 1457 1457 1458 1459 1459 1460 1461 1462 1463 1464 1465 1465 1466 1467 1467 1468 1468 1469 1470 1470 1471 1472 1473 1474 1475 1475 1476 1477 1477 1478 1478 1479 1480 1481 1482 1483 1484 1484 1485 1486 1486 1487 1487 1488 1489 1489 1490 1491 1492 1493 1494 1494 1495 1496 1496 1497 1497 1498 1499 1499 1500 1501 1502 1503 1504 1504 1505 1506 1506 1507 1507 1508 1509 1509 1510 1511 1512 1513 1514 1514 1515 1516 1516 1517 1517 1518 1519 1519 1520 1521 1522 1523 1524 1524 1525 1526 1526 1527 1527 1528 1529 1529 1530 1531 1532 1533 1534 1535 1535 1536 1537 1537 1538 1538 1539 1540 1540 1541 1542 1543 1544 1545 1545 1546 1547 1547 1548 1548 1549 1550 1551 1552 1553 1554 1554 1555 1556 1556 1557 1557 1558 1559 1559 1560 1561 1562 1563 1564 1565 1565 1566 1567 1567 1568 1568 1569 1570 1570 1571 1572 1573 1574 1575 1575 1576 1577 1577 1578 1578 1579 1580 1581 1582 1583 1584 1584 1585 1586 1586 1587 1587 1588 1589 1589 1590 1591 1592 1593 1594 1594 1595 1596 1596 1597 1597 1598 1599 1599 1600 1601 1602 1603 1604 1604 1605 1606 1606 1607 1607 1608 1609 1609 1610 1611 1612 1613 1614 1614 1615 1616 1616 1617 1617 1618 1619 1619 1620 1621 1622 1623 1624 1624 1625 1626 1626 1627 1627 1628 1629 1629 1630 1631 1632 1633 1634 1635 1635 1636 1637 1637 1638 1638 1639 1640 1640 1641 1642 1643 1644 1645 1645 1646 1647 1647 1648 1648 1649 1650 1651 1652 1653 1654 1654 1655 1656 1656 1657 1657 1658 1659 1659 1660 1661 1662 1663 1664 1665 1665 1666 1667 1667 1668 1668 1669 1670 1670 1671 1672 1673 1674 1675 1675 1676 1677 1677 1678 1678 1679 1680 1681 1682 1683 1684 1684 1685 1686 1686 1687 1687 1688 1689 1689 1690 1691 1692 1693 1694 1694 1695 1696 1696 1697 1697 1698 1699 1699 1700 1701 1702 1703 1704 1704 1705 1706 1706 1707 1707 1708 1709 1709 1710 1711 1712 1713 1714 1714 1715 1716 1716 1717 1717 1718 1719 1719 1720 1721 1722 1723 1724 1724 1725 1726 1726 1727 1727 1728 1729 1729 1730 1731 1732 1733 1734 1735 1735 1736 1737 1737 1738 1738 1739 1740 1740 1741 1742 1743 1744 1745 1745 1746 1747 1747 1748 1748 1749 1750 1751 1752 1753 1754 1754 1755 1756 1756 1757 1757 1758 1759 1759 1760 1761 1762 1763 1764 1765 1765 1766 1767 1767 1768 1768 1769 1770 1770 1771 1772 1773 1774 1775 1775 1776 1777 1777 1778 1778 1779 1780 1781 1782 1783 1784 1784 1785 1786 1786 1787 1787 1788 1789 1789 1790 1791 1792 1793 1794 1794 1795 1796 1796 1797 1797 1798 1799 1799 1800 1801 1802 1803 1804 1804 1805 1806 1806 1807 1807 1808 1809 1809 1810 1811 1812 1813 1814 1814 1815 1816 1816 1817 1817 1818 1819 1819 1820 1821 1822 1823 1824 1824 1825 1826 1826 1827 1827 1828 1829 1829 1830 1831 1832 1833 1834 1835 1835 1836 1837 1837 1838 1838 1839 1840 1840 1841 1842 1843 1844 1845 1845 1846 1847 1847 1848 1848 1849 1850 1851 1852 1853 1854 1854 1855 1856 1856 1857 1857 1858 1859 1859 1860 1861 1862 1863 1864 1865 1865 1866 1867 1867 1868 1868 1869 1870 1870 1871 1872 1873 1874 1875 1875 1876 1877 1877 1878 1878 1879 1880 1881 1882 1883 1884 1884 18</pre>		

ARMAND HOUPIN	clientController.php	3/3
<pre> 130 131 return view('/Client/consultationcampagne')->with('campagne', \$campagne)-> 132 with('stat_true', \$stat_true)->with('vitmax', \$vitmax)->with('stat', \$stat)->with('statVitMoyen_PL', \$statVitMoyen_PL)->with(133 'statHeureStat_PL', \$statHeureStat_PL)->with(134 'statVitesseInferieurOuEgale_PL', \$statVitesseInferieurOuEgale_PL)->with(135 ('statVitesseLimitMoins20_PL', \$statVitesseLimitMoins20_PL)->with(136 'statVitesseLimitPlus20_PL', \$statVitesseLimitPlus20_PL)->with(137 'statVitesseLimitPlus30_PL', \$statVitesseLimitPlus30_PL)->with(138 'statVitesseLimitPlus40_PL', \$statVitesseLimitPlus40_PL)->with(139 'statVitesseLimitPlus50_PL', \$statVitesseLimitPlus50_PL)->with(140 'statVitMoyen_VL', \$statVitMoyen_VL)->with('statHeureStat_VL', \$statHeureStat_VL)->with('statVitesseInferieurOuEgale_VL', \$statVitesseInferieurOuEgale_VL)->with('statVitesseLimitMoins20_VL', \$statVitesseLimitMoins20_VL)->with('statVitesseLimitPlus20_VL', \$statVitesseLimitPlus20_VL)->with('statVitesseLimitPlus30_VL', \$statVitesseLimitPlus30_VL)->with('statVitesseLimitPlus40_VL', \$statVitesseLimitPlus40_VL)->with('statVitesseLimitPlus50_VL', \$statVitesseLimitPlus50_VL); 141 142 } elseif(empty(\$statVitMoyen_VL)) 143 { 144 145 \$statVitMoyen_VL = NULL; 146 \$statHeureStat_VL = NULL; 147 \$statVitesseInferieurOuEgale_VL = NULL; 148 \$statVitesseLimitMoins20_VL = NULL; 149 \$statVitesseLimitPlus20_VL = NULL; 150 \$statVitesseLimitPlus30_VL = NULL; 151 \$statVitesseLimitPlus40_VL = NULL; 152 \$statVitesseLimitPlus50_VL = NULL; 153 154 return view('/Client/consultationcampagne')->with('campagne', \$campagne)-> 155 with('stat_true', \$stat_true)->with('vitmax', \$vitmax)->with('stat', \$stat)->with('statVitMoyen_PL', \$statVitMoyen_PL)->with(156 'statHeureStat_PL', \$statHeureStat_PL)->with(157 'statVitesseInferieurOuEgale_PL', \$statVitesseInferieurOuEgale_PL)->with(158 ('statVitesseLimitMoins20_PL', \$statVitesseLimitMoins20_PL)->with(159 'statVitesseLimitPlus20_PL', \$statVitesseLimitPlus20_PL)->with(160 'statVitesseLimitPlus30_PL', \$statVitesseLimitPlus30_PL)->with(161 'statVitesseLimitPlus40_PL', \$statVitesseLimitPlus40_PL)->with(162 'statVitesseLimitPlus50_PL', \$statVitesseLimitPlus50_PL)->with(163 'statVitMoyen_VL', \$statVitMoyen_VL)->with('statHeureStat_VL', \$statHeureStat_VL)->with('statVitesseInferieurOuEgale_VL', \$statVitesseInferieurOuEgale_VL)->with('statVitesseLimitMoins20_VL', \$statVitesseLimitMoins20_VL)->with('statVitesseLimitPlus20_VL', \$statVitesseLimitPlus20_VL)->with('statVitesseLimitPlus30_VL', \$statVitesseLimitPlus30_VL)->with('statVitesseLimitPlus40_VL', \$statVitesseLimitPlus40_VL)->with('statVitesseLimitPlus50_VL', \$statVitesseLimitPlus50_VL); 164 165 } else{ 166 167 return view('/Client/consultationcampagne')->with('campagne', \$campagne)-> 168 with('stat_true', \$stat_true)->with('vitmax', \$vitmax)->with('stat', \$stat)->with('statVitMoyen_PL', \$statVitMoyen_PL)->with(169 'statHeureStat_PL', \$statHeureStat_PL)->with(170 'statVitesseInferieurOuEgale_PL', \$statVitesseInferieurOuEgale_PL)->with(171 ('statVitesseLimitMoins20_PL', \$statVitesseLimitMoins20_PL)->with(172 'statVitesseLimitPlus20_PL', \$statVitesseLimitPlus20_PL)->with(173 'statVitesseLimitPlus30_PL', \$statVitesseLimitPlus30_PL)->with(174 'statVitesseLimitPlus40_PL', \$statVitesseLimitPlus40_PL)->with(175 'statVitesseLimitPlus50_PL', \$statVitesseLimitPlus50_PL)->with(176 'statVitMoyen_VL', \$statVitMoyen_VL)->with('statHeureStat_VL', \$statHeureStat_VL)->with('statVitesseInferieurOuEgale_VL', \$statVitesseInferieurOuEgale_VL)->with('statVitesseLimitMoins20_VL', \$statVitesseLimitMoins20_VL)->with('statVitesseLimitPlus20_VL', \$statVitesseLimitPlus20_VL)->with('statVitesseLimitPlus30_VL', \$statVitesseLimitPlus30_VL)->with('statVitesseLimitPlus40_VL', \$statVitesseLimitPlus40_VL)->with('statVitesseLimitPlus50_VL', \$statVitesseLimitPlus50_VL); 177 178 } 179 } elseif(\$stat_true == 0) 180 { 181 182 return view('/Client/consultationcampagne')->with('campagne', \$campagne)->with(183 'stat_true', \$stat_true); 184 185 } 186 187 }</pre>		

ARMAND HOUPIN	gestionboitierController.php	1/2
<pre> 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 use App\Models\User; 8 use App\Models\campagnemesure; 9 use App\Models\boitier; 10 use App\Models\demande; 11 class gestionboitierController extends Controller 12 { 13 14 // 15 // 16 // Gestion Boitier 17 // 18 // 19 20 //Affiche la liste des boitiers 21 public function listeBoitier() 22 { 23 24 if (! auth()->check()) 25 return redirect()->route('login'); 26 } 27 28 \$listeboitier = boitier::all(); 29 \$alarme = boitier::all(); 30 31 \$alarme_popup = NULL; 32 foreach(\$alarme as \$alarmes) 33 { 34 if(!empty(\$alarmes->alarmeBatterie)) 35 { 36 \$alarme_popup = 1; 37 } 38 } 39 40 \$demande = demande::all(); 41 42 \$demande_popup = NULL; 43 44 foreach(\$demande as \$demandes) 45 { 46 \$demande_popup += 1; 47 } 48 return view('/gererboitier/ListeBoitier')->with('boitier', \$listeboitier)->with(49 'alarme', \$alarme)->with('alarme_popup', \$alarme_popup); 50 } 51 52 //Affiche la page d'ajout des boitiers 53 public function creationBoitier() 54 { 55 if (! auth()->check()) 56 return redirect()->route('login'); 57 } 58 59 return view('/gererboitier/creationBoitier'); 60 } 61 62 //Ajouter des boitiers 63 public function ajouterBoitier() 64 { 65 if (! auth()->check()) 66 return redirect()->route('login'); 67 68 \$boitier = new boitier; 69 \$boitier->adrSigfox = request('sigfox'); 70 \$boitier->statut = request('statut'); 71 \$boitier->alarmeBatterie = NULL; 72 73 \$boitier->save(); 74 75 return redirect()->route('listeboitieralert_path', ['alert' => 1]); 76 } 77 </pre>		

ARMAND HOUPIN	gestionboitierController.php	2/2
<pre> 78 79 //Supprimer la liste des boitiers 80 public function supprimerBoitier(\$id) 81 { 82 if (! auth()->check()) { 83 return redirect()->route('login'); 84 } 85 86 \$boitier = boitier::find(\$id); 87 88 \$campagne = campagnemesure::where('id_boitier', \$boitier->id)->first(); 89 if(!empty(\$campagne)) 90 { 91 \$campagne->id_boitier = NULL; 92 \$campagne->update(); 93 } 94 95 \$boitier->delete(); 96 97 return redirect()->route('listeboitieralert_path', ['alert' => 3]); 98 99 } 100 101 102 //Affiche la page modifier des boitiers 103 public function modifierBoitier(\$id) 104 { 105 106 if (! auth()->check()) { 107 return redirect()->route('login'); 108 } 109 110 \$boitier = boitier::find(\$id); 111 112 \$campagne = campagnemesure::where('id_boitier', \$boitier->id)->first(); 113 //\$campagne = campagnemesure::find(\$boitier->id); 114 115 return view('/gererboitier/modifierBoitier')->with('boitier', \$boitier)->with(116 'campagne', \$campagne); 117 118 } 119 120 //Modifie des boitiers 121 public function modifierBoitierConfirm(\$id) 122 { 123 124 if (! auth()->check()) { 125 return redirect()->route('login'); 126 } 127 128 \$boitier = boitier::find(\$id); 129 \$boitier->adrSigfox = request('sigfox'); 130 \$boitier->alarmeBatterie = request('alarmeBatterie'); 131 if(!empty(request('statut'))){ 132 \$boitier->statut = request('statut'); 133 } 134 135 \$boitier->update(); 136 137 return redirect()->route('listeboitieralert_path', ['alert' => 2]); 138 } 139 140 public function BoitierAlarmeBatterie(\$id) 141 { 142 143 if (! auth()->check()) { 144 return redirect()->route('login'); 145 } 146 147 \$boitier = boitier::find(\$id); 148 \$boitier->alarmeBatterie = NULL; 149 150 \$boitier->update(); 151 152 return redirect()->route('listeboitieralert_path', ['alert' => 5]); 153 } 154 </pre>		

ARMAND HOUPIN	gestioncampagneController.php	1/3
<pre> 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 use App\Models\User; 8 use App\Models\campagnemesure; 9 use App\Models\boitier; 10 use App\Models\demande; 11 use Geocoder\Laravel\ProviderAndDumperAggregator as Geocoder; 12 13 class gestioncampagnemesureController extends Controller 14 { 15 // 16 // 17 // Gestion Campagne de mesure 18 // 19 // 20 21 22 //Affiche la liste des campagnes 23 public function listeCampagneMesure() 24 { 25 if (! auth()->check()) { 26 return redirect()->route('login'); 27 } 28 29 \$listecampagne = campagnemesure::all(); 30 \$alarme = boitier::all(); 31 32 \$alarme_popup = NULL; 33 foreach(\$alarme as \$alarmes) 34 { 35 if(!empty(\$alarmes->alarmeBatterie)) 36 { 37 \$alarme_popup = 1; 38 } 39 } 40 41 \$demande = demande::all(); 42 43 \$demande_popup = NULL; 44 45 foreach(\$demande as \$demandes) 46 { 47 \$demande_popup += 1; 48 } 49 return view('/gerercampagne>ListeCampagneMesure')->with('campagne', \$listecampagne) 50 ->with('alarme', \$alarme)->with('alarme_popup', \$alarme_popup); 51 } 52 53 //Affiche la page d'ajout des campagnes 54 public function creationCampagneMesure() 55 { 56 if (! auth()->check()) { 57 return redirect()->route('login'); 58 } 59 60 \$listeclient = User::all(); 61 \$listeboitier = boitier::all(); 62 return view('/gerercampagne/creationCampagneMesure')->with('client', \$listeclient) 63 ->with('boitier', \$listeboitier); 64 } </pre>		

```

54
55
56
57
58
59
60
61
62
63
64
65 //Ajouter des campagnes avec Eloquent
66 public function ajouterCampagneMesure()
67 {
68     if (! auth()->check()) {
69         return redirect()->route('login');
70     }
71
72     $campagne = new campagnemesure;
73     $campagne->adresseCampagne = request('adresseCampagne');
74     $campagne->statut = request('statut');
75     $campagne->codePostal = request('codePostal');
76     $campagne->ville = request('ville');
77     $campagne->numeroRoute = request('numeroRoute');
78     $campagne->Direction = request('Direction');
79     $campagne->DébutCampagne = request('DebutCampagne');
80     $campagne->FinCampagne = request('FinCampagne');
81     $campagne->limitationvitesse = request('limitationvitesse');
82     $campagne->id_user = request('id_user');
83
84     if(!empty(request('id_boitier')))
85     {
86         $campagne->id_boitier = request('id_boitier');
87         $boitier = boitier::find(request('id_boitier'));
88         $boitier->statut = "utilisé";
89         $boitier->update();
90     }
91
92     $campagne->save();
93
94     return redirect()->route('listecampagnemesurealert_path', ['alert' => 1]);
95 }
96
97
98 //Supprimer des campagnes avec Eloquent
99 public function supprimerCampagneMesure($id)
100 {
101     if (! auth()->check()) {
102         return redirect()->route('login');
103     }
104
105     $campagne = campagnemesure::find($id);
106
107     $campagne->delete();
108
109     return redirect()->route('listecampagnemesurealert_path', ['alert' => 3]);
110 }
111
112
113 //Afficher la page modifier des campagnes
114 public function modifierCampagneMesure($id)
115 {
116     if (! auth()->check()) {
117         return redirect()->route('login');
118     }
119
120     $listeclient = User::all();
121     $listebloitier = boitier::all();
122
123     $campagne = campagnemesure::find($id);
124
125     $client = $campagne->id_user;
126     $clientdefault = User::find($client);
127
128     $boitier = $campagne->id_boitier;
129     $boitierdefault = boitier::find($boitier);
130
131
132     return view('/gerercampagne/modifierCampagneMesure')->with('client', $listeclient)
133         ->with('campagne', $campagne)->with('clientdefault', $clientdefault)->with(
134             'boitier', $listebloitier)->with('boitierdefault', $boitierdefault);
135 }
```

ARMAND HOUPIN	gestioncampagneController.php	3/3
<pre> 135 //Modifier des campagnes avec Eloquent 136 public function modifierCampagneMesureConfirm(\$id) 137 { 138 139 if (! auth()->check()) { 140 return redirect()->route('login'); 141 } 142 143 144 \$campagne = campagnemesure::find(\$id); 145 \$campagne->adresseCampagne = request('adresseCampagne'); 146 \$campagne->statut = request('statut'); 147 \$campagne->codePostal = request('codePostal'); 148 \$campagne->ville = request('ville'); 149 \$campagne->numeroRoute = request('numeroRoute'); 150 \$campagne->Direction = request('Direction'); 151 \$campagne->DebutCampagne = request('DebutCampagne'); 152 \$campagne->FinCampagne = request('FinCampagne'); 153 \$campagne->limitationvitesse = request('limitationvitesse'); 154 \$campagne->id_user = request('id_user'); 155 156 if(!empty(request('id_boitier'))) 157 { 158 \$campagne->id_boitier = request('id_boitier'); 159 \$boitier = boitier::find(request('id_boitier')); 160 \$boitier->statut = "utilisé"; 161 \$boitier->update(); 162 } else 163 { 164 \$boitier = boitier::find(\$campagne->id_boitier); 165 if(!empty(\$boitier)){ 166 \$boitier->statut = "non utilisé"; 167 \$boitier->update(); 168 \$campagne->id_boitier = NULL; 169 } 170 } 171 172 173 \$campagne->update(); 174 175 return redirect()->route('listecampagnemesurealert_path', ['alert' => 2]); 176 177 } 178 </pre>		

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  use App\Models\User;
8  use Illuminate\Support\Facades\Hash;
9  use App\Models\boitier;
10 use App\Models\demande;
11 use Illuminate\Support\Facades\Mail;
12 use App\Mail\MessageGoogle;
13
14 class gestioncompteController extends Controller
15 {
16
17
18
19     //
20     //
21     // Gestion Client
22     //
23     //
24
25     //methode qui hash le mot de passe
26     protected function validator(array $data)
27     {
28         return Validator::make($data, [
29             'name' => ['required', 'string', 'max:255'],
30             'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
31             'password' => ['required', 'string', 'min:8', 'confirmed'],
32         ]);
33     }
34
35
36     //Afficher la page inscription client
37     public function ajouterClientget()
38     {
39         if (! auth()->check()) {
40             return redirect('/login');
41         }
42
43         return view('/gerercompte/inscription');
44     }
45
46
47     //Afficher la page inscription Admin/Technicien
48     public function ajouterAdminTechnicienget()
49     {
50         if (! auth()->check()) {
51             return redirect('/login');
52         }
53
54         return view('/gerercompte/inscriptionAdmin');
55     }
56
57

```

```

58     //Afficher la liste des clients
59     public function listeClient()
60     {
61         if (! auth()->check())
62             return redirect('/login');
63     }
64
65     $listeclient = User::all();
66     $alarme = boitier::all();
67
68     $alarme_popup = NULL;
69     foreach($alarme as $alarmes)
70     {
71         if(!empty($alarmes->alarmeBatterie))
72         {
73             $alarme_popup = 1;
74         }
75     }
76
77     $demande = demande::all();
78
79     $demande_popup = NULL;
80
81     foreach($demande as $demandes)
82     {
83         $demande_popup += 1;
84     }
85
86     return view('/gerercompte/ListeClient')->with('client', $listeclient)->with('alarme',
87         ', $alarme)->with('alarme_popup', $alarme_popup);
88 }
89
90 //Ajouter Client avec Eloquent
91 public function ajouterClient(Request $request)
92 {
93     if (! auth()->check())
94         return redirect('/login');
95
96     $email_existant = User::where('email', $request['email'])->first();
97
98     if(!empty($email_existant))
99     {
100        $alert = 1;
101        $messagealert = "L'adresse mail existe deja !";
102
103        return view('/gerercompte/inscription')->with('alert', $alert)->with(
104            'messagealert', $messagealert);
105    }
106
107    $client = new User;
108    $client->email = $request['email'];
109    $client->role_id = 2;
110    $client->name = $request['name'];
111    $client->password = Hash::make($request['password']);
112    $client->prenom = $request['prenom'];
113    $client->identifiant = $request['identifiant'];
114    $client->numeroTel = $request['numeroTel'];
115    $client->adresseClient = $request['adresseClient'];
116    $client->codePostal = $request['codePostal'];
117    $client->ville = $request['ville'];
118    $client->region = $request['region'];
119    $client->preference = "theme_dark";
120
121    if( $request['entreprise'] == "oui") {
122        $client->nomEntreprise = $request['entreprise'];
123        $client->nomEntreprise = $request['nomEntreprise'];
124    }
125
126    Mail::to($client)->bcc("kolantr2021snir@gmail.com")
127        ->queue(new MessageGoogle($request->all()));
128
129    $client->save();
130
131    return redirect()->route('listeClientalert_path', ['alert' => 1]);
132
133 }
134

```

ARMAND HOUPIN	gestioncompteController.php	3/5
	<pre> 136 //Supprimer Client avec Eloquent 137 public function supprimerClient(\$id) 138 { 139 140 if (! auth()->check()) { 141 return redirect('/login'); 142 } 143 144 \$client = User::find(\$id); 145 146 \$client->delete(); 147 148 return redirect()->route('listeClientalert_path', ['alert' => 3]); 149 } 150 151 //Affichage de la page Modifier client 152 public function modifierClient(\$id) 153 { 154 155 if (! auth()->check()) { 156 return redirect('/login'); 157 } 158 159 \$client = User::find(\$id); 160 161 return view('/gerercompte/ModifierClient')->with('client', \$client); 162 } 163 164 //Modifier Client avec Eloquent 165 public function modifierClientConfirm(\$id) 166 { 167 168 if (! auth()->check()) { 169 return redirect('/login'); 170 } 171 172 \$client = User::find(\$id); 173 \$client->email = request('email'); 174 \$client->name = request('name'); 175 \$client->prenom = request('prenom'); 176 177 178 if(!empty(request('numeroTel'))){ 179 \$client->numeroTel = request('numeroTel'); 180 } 181 182 if(!empty(request('adresseClient'))){ 183 \$client->adresseClient = request('adresseClient'); 184 } 185 186 if(!empty(request('codePostal'))){ 187 \$client->codePostal = request('codePostal'); 188 } 189 190 if(!empty(request('ville'))){ 191 \$client->ville = request('ville'); 192 } 193 194 if(!empty(request('region'))){ 195 \$client->region = request('region'); 196 } 197 198 if(!empty(request('nomEntreprise'))){ 199 \$client->nomEntreprise = request('nomEntreprise'); 200 } else{ 201 \$client->nomEntreprise = "NULL"; 202 } 203 204 \$client->update(); 205 206 return redirect()->route('listeClientalert_path', ['alert' => 2]); 207 } 208 }</pre>	

```

213     //  

214     //  

215     // Gestion Admin / Technicien  

216     //  

217     //  

218  

219  

220     //Afficher la liste Admin / Technicien  

221     public function listeAdminTechnicienmodifier()  

222     {  

223         if (! auth()->check()) {  

224             return redirect('/login');  

225         }  

226  

227         $listeclient = User::all();  

228         $alarme = boitier::all();  

229  

230         $alarme_popup = NULL;  

231         foreach($alarme as $alarmes)  

232         {  

233             if(!empty($alarmes->alarmeBatterie))  

234             {  

235                 $alarme_popup = 1;  

236             }  

237         }  

238  

239         $demande = demande::all();  

240  

241         $demande_popup = NULL;  

242  

243         foreach($demande as $demandes)  

244         {  

245             $demande_popup += 1;  

246         }  

247         return view('/gerercompte/ListeAdminTechnicien')->with('client', $listeclient)->  

248             with('alarme', $alarme)->with('alarme_popup', $alarme_popup);  

249  

250  

251     //Ajouter Admin / Technicien avec Eloquent  

252     public function ajouterAdminTechnicien(Request $request)  

253     {  

254  

255         if (! auth()->check()) {  

256             return redirect('/login');  

257         }  

258  

259         $email_existant = User::where('email', $request['email'])->first();  

260  

261         if(!empty($email_existant))  

262         {  

263             $alert = 1;  

264             $messagealert = "L'adresse mail existe déjà !";  

265  

266             return view('/gerercompte/ListeAdminTechnicien')->with('alert', $alert)->with(''  

267                 messagealert', $messagealert);  

268         }  

269  

270         $client = new User;  

271         $client->email = $request['email'];  

272         $client->role_id = $request['role'];  

273         $client->name = $request['name'];  

274         $client->prenom = $request['prenom'];  

275         $client->password = Hash::make($request['password']);  

276         $client->identifiant = $request['identifiant'];  

277         $client->preference = "theme_dark";  

278  

279         $client->save();  

280  

281         $alert = 1;  

282         $messagealert = "Le compte Admin / Technicien a bien été créé !";  

283  

284         Mail::to($client)->bcc("kolantr2021snir@gmail.com")  

285             ->queue(new MessageGoogle($request->all()));  

286  

287         $listeclient = User::all();  

288         return view('/gerercompte/ListeAdminTechnicien')->with('client', $listeclient)->  

289             with('alert', $alert)->with('messagealert', $messagealert);  

290     }

```

ARMAND HOUPIN	gestioncompteController.php	5/5
	<pre> 291 //Supprimer Admin / Technicien avec Eloquent 292 public function supprimerAdminTechnicien(\$id) 293 { 294 295 if (! auth()->check()) { 296 return redirect('/login'); 297 } 298 299 \$client = User::find(\$id); 300 301 if(!empty(\$client)) 302 { 303 \$client->delete(); 304 305 \$alert = 1; 306 \$messagealert = "Le compte Admin / Technicien a bien été supprimé !"; 307 308 \$listeclient = User::all(); 309 return view('/gerercompte/ListeAdminTechnicien')->with('client', \$listeclient)-> 310 with('alert', \$alert)->with('messagealert', \$messagealert); 311 } else { 312 313 \$listeclient = User::all(); 314 return view('/gerercompte/ListeAdminTechnicien')->with('client', \$listeclient); 315 } 316 317 //Afficher la page Modifier Admin / Technicien 318 public function modifierAdminTechnicien(\$id) 319 { 320 321 if (! auth()->check()) { 322 return redirect('/login'); 323 } 324 325 \$client = User::find(\$id); 326 327 return view('/gerercompte/ModifierAdminTechnicien')->with('client', \$client); 328 329 } 330 331 332 //Modifier Admin / Technicien avec Eloquent 333 public function modifierAdminTechnicienConfirm(\$id) 334 { 335 if (! auth()->check()) { 336 return redirect('/login'); 337 } 338 339 \$client = User::find(\$id); 340 \$client->email = request('email'); 341 \$client->name = request('name'); 342 \$client->prenom = request('prenom'); 343 \$client->role_id = request('role'); 344 345 \$client->update(); 346 347 \$alert = 1; 348 \$messagealert = "Le compte Admin / Technicien a bien été modifié !"; 349 350 \$listeclient = User::all(); 351 return view('/gerercompte/ListeAdminTechnicien')->with('client', \$listeclient)-> 352 with('alert', \$alert)->with('messagealert', \$messagealert); 353 } </pre>	

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  use App\Models\User;
8  use App\Models\campagnemesure;
9  use App\Models\boitier;
10
11 use App\Models\demande;
12
13 use Illuminate\Foundation\Auth\User as Authenticatable;
14 use Illuminate\Support\Facades\Auth;
15
16 use Illuminate\Support\Facades\Hash;
17
18 class HomeController extends Controller
19 {
20     /**
21      * Create a new controller instance.
22      *
23      * @return void
24      */
25     public function __construct()
26     {
27         $this->middleware('auth');
28     }
29
30     /**
31      * Show the application dashboard.
32      *
33      * @return \Illuminate\Contracts\Support\Renderable
34      */
35     public function index()
36     {
37
38         if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien')){
39             $alarme = boitier::all();
40
41             $alarme_popup = NULL;
42
43             foreach($alarme as $alarmes)
44             {
45                 if(!empty($alarmes->alarmeBatterie))
46                 {
47                     $alarme_popup = 1;
48                 }
49             }
50
51             $nb_boitier = 0;
52             $nb_campagne = 0;
53             $nb_compte = 0;
54             $nb_client = 0;
55             $nb_admin = 0;
56             $nb_technicien = 0;
57             $camp_cours = 0;
58             $camp_fini = 0;
59             $boitier_utiliser = 0;
60             $boitier_repos = 0;
61
62             $boitier = boitier::all();
63             foreach($boitier as $boitiers)
64             {
65                 $nb_boitier = $nb_boitier + 1;
66                 if ($boitiers->statut == "non utilisé") {
67                     $boitier_repos = $boitier_repos + 1;
68                 }else{
69                     $boitier_utiliser = $boitier_utiliser + 1;
70                 }
71             }
72         }
73     }
74 }

```



```

73     $campagne = campagnemesure::all();
74     foreach($campagne as $campagnes)
75     {
76         $nb_campagne = $nb_campagne + 1;
77         if ($campagnes->statut == "fini") {
78             $camp_fini = $camp_fini + 1;
79         }else{
80             $camp_cours = $camp_cours + 1;
81         }
82     }
83
84     $user = User::all();
85     foreach($user as $users)
86     {
87         $nb_compte = $nb_compte + 1;
88         if ($users->role_id == 1) {
89             $nb_admin = $nb_admin + 1;
90         }else if($users->role_id == 3){
91             $nb_client = $nb_client + 1;
92         }else if($users->role_id == 2){
93             $nb_technicien = $nb_technicien +1;
94         }
95     }
96
97     return view('/accueil')->with('alarme', $alarme)->with('alarme_popup', $alarme_popup)->with("nb_compte", $nb_compte)->with("nb_admin", $nb_admin)->with("nb_client", $nb_client)->with("nb_technicien", $nb_technicien)->with("nb_campagne", $nb_campagne)->with("camp_fini", $camp_fini)->with("camp_cours", $camp_cours)->with("nb_boitier", $nb_boitier)->with("boitier_repos", $boitier_repos)->with("boiter_utiliser", $boiter_utiliser);
98 }
99 else
100 {
101     $id = Auth::user()->id;
102
103     $campagne = campagnemesure::all();
104
105     $nb_campagne = 0;
106     $nb_camp_cours = 0;
107     $nb_camp_fini = 0;
108
109     foreach($campagne as $campagnes){
110         if($campagnes->id_user == $id)
111         {
112             $nb_campagne = $nb_campagne + 1;
113             if($campagnes->statut == "fini"){
114                 $nb_camp_fini = $nb_camp_fini + 1;
115             } else {
116                 $nb_camp_cours = $nb_camp_cours + 1;
117             }
118         }
119     }
120
121     return view('/accueil')->with('nb_campagne', $nb_campagne)->with('nb_camp_cours', $nb_camp_cours)->with('nb_camp_fini', $nb_camp_fini);
122 }
123
124
125
126     protected function validator(array $data)
127     {
128         return Validator::make($data, [
129             'name' => ['required', 'string', 'max:255'],
130             'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
131             'password' => ['required', 'string', 'min:8', 'confirmed'],
132         ]);
133     }
134

```

```

136     public function liste()
137     {
138         $listecampagne = campagnemesure::all();
139         $listeboitier = boitier::all();
140         $listeclient = User::all();
141
142         $alarme = boitier::all();
143
144         $alarme_popup = NULL;
145         foreach($alarme as $alarmes)
146         {
147             if(!empty($alarmes->alarmeBatterie))
148             {
149                 $alarme_popup += 1;
150             }
151         }
152
153         $demande = demande::all();
154
155         $demande_popup = NULL;
156
157         foreach($demande as $demandes)
158         {
159             $demande_popup += 1;
160         }
161
162         return view('/liste')->with('campagne', $listecampagne)->with('listeboitier', $listeboitier)->with('listeclient', $listeclient)->with('alarme', $alarme)->with('alarme_popup', $alarme_popup);
163     }
164
165     public function boitierdelete($id)
166     {
167
168         $campagne = campagnemesure::where('id_boitier', $id)->first();
169         if(!empty($campagne))
170         {
171             $campagne->id_boitier = NULL;
172             $campagne->update();
173
174             $boitier = boitier::find($id);
175             $boitier->statut = "non utilisé";
176             $boitier->update();
177         }
178
179         return redirect()->route('liste');
180     }
181
182     public function campagnedelete($id)
183     {
184         $campagne = campagnemesure::find($id);
185
186         $campagne->delete();
187
188         return redirect()->route('liste');
189     }
190
191     public function userdelete($id)
192     {
193         $campagne = campagnemesure::where('id_user', $id)->first();
194         if(!empty($campagne))
195         {
196             $campagne->id_user = NULL;
197             $campagne->update();
198         }
199
200         return redirect()->route('liste');
201     }
202 }
```

ARMAND HOUPIN	HomeController.php	4/4
<pre> 204 public function listemodifierstatutEncours(\$id) 205 { 206 \$campagne = campagnemesure::find(\$id); 207 \$campagne->statut = "en cours"; 208 \$campagne->update(); 209 210 return redirect()->route('liste'); 211 } 212 213 public function listemodifierstatutfini(\$id) 214 { 215 \$campagne = campagnemesure::find(\$id); 216 \$campagne->statut = "fini"; 217 \$campagne->update(); 218 219 220 return redirect()->route('liste'); 221 } 222 223 public function listeajouterboitier(\$id_campagne, \$id_boitier) 224 { 225 \$campagne = campagnemesure::find(\$id_campagne); 226 \$campagne->id_boitier = \$id_boitier; 227 \$campagne->update(); 228 229 \$boitier = boitier::find(\$id_boitier); 230 \$boitier->statut = "utilisé"; 231 \$boitier->update(); 232 233 234 return redirect()->route('liste'); 235 } 236 237 } 238 </pre>		

ARMAND HOUPIN	PagesController.php	1/1
<pre> 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class PagesController extends Controller 8 { 9 public function home() 10 { 11 return \view('pages.login'); 12 } 13 14 public function login() 15 { 16 17 if (auth()->check()) { 18 return redirect()->route('accueil'); 19 } else { 20 return view('auth.login'); 21 } 22 } 23 } 24 </pre>		

ARMAND HOUPIN	Home.blade.php	1/4
<pre> 1 <!DOCTYPE html> 2 <html lang="fr"> 3 <head> 4 <meta charset="utf-8"> 5 <meta name="viewport" content="width=device-width, initial-scale=1"> 6 <meta name="description" content=""> 7 <title> {{ \$title }} </title> 8 9 10 <link rel="shortcut icon" type="image/png" href="https://encrypted-tbn0.gstatic.com/ 11 images? 12 q=tbn:ANd9GcS9i_4sX9b1uIV_QXHx9HmXOyJkftoM_LC94wQh000ncqvh6DT9wsFa_xN_Nt4shAcdKgc& 13 usqp=CAU" /> 14 15 <!-- Bootstrap core CSS --> 16 <link href="{{ asset('/bootstrap/bootstrap.min.css')}}" rel="stylesheet"> 17 <link href="{{ asset('/bootstrap/navbar.css')}}" rel="stylesheet"> 18 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/ 19 font-awesome.min.css" crossorigin="anonymous"> 20 <script src="https://kit.fontawesome.com/64d58efce2.js" crossorigin="anonymous"></script> 21 22 <style> 23 .bd-placeholder-img { 24 font-size: 1.125rem; 25 text-anchor: middle; 26 -webkit-user-select: none; 27 -moz-user-select: none; 28 user-select: none; 29 } 30 31 @media (min-width: 768px) { 32 .bd-placeholder-img-lg { 33 font-size: 3.5rem; 34 } 35 36 .bg-violet{ 37 background-color: #201c33; 38 } 39 40 .dropdown-itemArmand{display:block;width:100%;padding:.25rem 1rem;clear:both; 41 font-weight:400;color: white;text-align:inherit;text-decoration:none;white-space: 42 nowrap;background-color:transparent;border:0} 43 44 .nav-linkArmand{display:block;padding:.1rem 0.5rem;text-decoration:none;transition: 45 color .15s ease-in-out,background-color .15s ease-in-out,border-color .15s 46 ease-in-out} 47 48 .dropdown-itemblack{display:block;width:100%;padding:.25rem 1rem;clear:both;font-weight 49 :400;color: black;text-align:inherit;text-decoration:none;white-space:nowrap; 50 background-color:transparent;border:0} 51 52 .dropdown-itemblack:focus,.dropdown-itemblack:hover { 53 color: white; 54 background-color:#BA8989; 55 } 56 57 .tx-white 58 { 59 color: #eceeef; 60 } 61 62 .tx-black 63 { 64 color: #2E2E2E; 65 } 66 67 .alert { 68 padding: 10px; 69 background-color: #f44336; 70 color: white; 71 opacity: 0.75; 72 transition: opacity 0.6s; 73 margin-bottom: 15px; 74 } </pre>		

```

71  .charging-container {
72    max-width: 150px;
73    width: 85px;
74    height: 35px;
75    border: 3px solid #2196F3;
76    border-radius: 5px;
77    position: relative;
78    cursor: pointer;
79    padding: 10px;
80  }
81
82  .charging-container::before {
83    content: '';
84    position: absolute;
85    width: 6px;
86    height: 16px;
87    background: #2196F3;
88    right: -7px;
89    top: 50%;
90    margin-top: -8px;
91    border-radius: 2px;
92  }
93
94  .charging-container::after {
95    content: '';
96    position: absolute;
97    top: 5px;
98    bottom: 5px;
99    left: 5px;
100   /* right: 10px; */
101   background: #2196F3;
102   transition: all .3s;
103   -webkit-animation: charging 1s infinite;
104   -moz-animation: charging 1s infinite;
105   animation: charging 1s infinite;
106   animation: charging 1s infinite;
107 }
108
109  @-webkit-keyframes charging {
110    from {
111      width: 15%;
112    }
113    to {
114      width: 5%;
115    }
116  }
117
118  @-moz-keyframes charging {
119    from {
120      width: 15%;
121    }
122    to {
123      width: 5%;
124    }
125  }
126  @keyframes charging {
127    from {
128      width: 15%;
129    }
130    to {
131      width: 5%;
132    }
133  }
134
135
136
137  .alert.success {background-color: #4CAF50; float: right;}
138  .alert.info {background-color: #2196F3;}
139  .alert.warning {background-color: #ff0000;}
140
141  .closebtn {
142    margin-left: 15px;
143    color: white;
144    font-weight: bold;
145    float: right;
146    font-size: 22px;
147    line-height: 20px;
148    cursor: pointer;
149    transition: 0.3s;
150  }
151

```

ARMAND HOUPIN	Home.blade.php	3/4
<pre> 152 .closebtn:hover { 153 color: black; 154 } 155 156 #map{ 157 height:400px; 158 } 159 160 #bloc 161 { 162 display: none; 163 } 164 165 .box { 166 width: 40%; 167 margin: 0 auto; 168 background: rgba(255,255,255,0.2); 169 padding: 35px; 170 border: 2px solid #fff; 171 border-radius: 20px/50px; 172 background-clip: padding-box; 173 text-align: center; 174 } 175 176 .button { 177 font-size: 1em; 178 padding: 10px; 179 color: #fff; 180 border: 2px solid #06D85F; 181 border-radius: 20px/50px; 182 text-decoration: none; 183 cursor: pointer; 184 transition: all 0.3s ease-out; 185 } 186 .button:hover { 187 background: #06D85F; 188 } 189 190 .overlay { 191 position: fixed; 192 top: 0; 193 bottom: 0; 194 left: 0; 195 right: 0; 196 background: rgba(0, 0, 0, 0.7); 197 transition: opacity 500ms; 198 visibility: hidden; 199 opacity: 0; 200 } 201 .overlay:target { 202 visibility: visible; 203 opacity: 1; 204 } 205 206 .popup { 207 margin: 70px auto; 208 padding: 20px; 209 background: #fff; 210 border-radius: 5px; 211 width: 30%; 212 position: relative; 213 transition: all 5s ease-in-out; 214 } 215 216 .popup h2 { 217 margin-top: 0; 218 color: #333; 219 font-family: Tahoma, Arial, sans-serif; 220 } 221 .popup .close { 222 position: absolute; 223 top: 20px; 224 right: 30px; 225 transition: all 200ms; 226 font-size: 30px; 227 font-weight: bold; 228 text-decoration: none; 229 color: #333; 230 } 231 .popup .close:hover { 232 color: #06D85F; 233 } 234 .popup .content { 235 max-height: 30%; 236 overflow: auto; 237 } 238 </pre>		

```

239 ▼ .navbar-toggler-armand {
240   padding:.25rem .75rem;
241   font-size:1.25rem;
242   line-height:1;
243   background-color:transparent;
244   border:1px solid gray;
245   border-radius:.25rem;
246   transition:box-shadow .15s ease-in-out
247 }
248
249 ▼ @media screen and (max-width: 700px){
250   .box{
251     width: 70%;
252   }
253   .popup{
254     width: 70%;
255   }
256 }
257
258 </style>
259
260 @yield('style')
261
262 </head>
263
264 <body class="bg-light">
265
266 @include('layouts.partials.navbar')
267
268 @yield('content')
269
270 <script src="{{ asset('/js/bootstrap.bundle.min.js')}}"></script>
271 <script src="{{ asset('/js/bootstrap.bundle.js')}}"></script>
272
273 @yield('scriptAlert')
274 @yield('script')
275
276 ▼ <script>
277   window.addEventListener('scroll', function(){
278     var header = document.querySelector('header');
279     header.classList.toggle('sticky', window.scrollY > 0);
280   });
281
282   function toggleMenu(){
283     var menuToggle = document.querySelector('.toggle');
284     var menu = document.querySelector('.menu');
285     menuToggle.classList.toggle('active');
286     menu.classList.toggle('active');
287   }
288 </script>
289
290 </body>
291
292
293 </html>
294

```

ARMAND HOUPIN	master.blade.php	1/1
<pre> 1 <!DOCTYPE html> 2 <html lang="fr"> 3 <head> 4 <meta charset="utf-8"> 5 <meta name="viewport" content="width=device-width, initial-scale=1"> 6 <meta name="description" content=""> 7 <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors"> 8 <meta name="generator" content="Hugo 0.79.0"> 9 <title> {{ \$title }} </title> 10 <link rel="shortcut icon" type="image/png" href="https://encrypted-tbn0.gstatic.com/ 11 images?q=tbn:ANd9GCrT538TjduUwc9IMe7dM-REsBgyFnMhw6dmA&usqp=CAU" /> 12 <link rel="canonical" href="https://getbootstrap.com/docs/5.0/examples/sign-in/"> 13 14 15 <!-- Bootstrap core CSS --> 16 <link href="{{ asset('/bootstrap/bootstrap.min.css')}}" rel="stylesheet"> 17 18 <style> 19 .bd-placeholder-img { 20 font-size: 1.125rem; 21 text-anchor: middle; 22 -webkit-user-select: none; 23 -moz-user-select: none; 24 user-select: none; 25 } 26 27 @media (min-width: 768px) { 28 .bd-placeholder-img-lg { 29 font-size: 3.5rem; 30 } 31 } 32 33 .bg-violet{ 34 background-color: #201c33; 35 } 36 37 .featurette-divider { 38 margin: 2rem 0; 39 color: #5a5a5a /* Space out the Bootstrap <hr> more */ 40 } 41 42 .featurette-divider-armand { 43 margin: 0.5rem 0; 44 color: #5a5a5a /* Space out the Bootstrap <hr> more */ 45 } 46 47 .dropdown-itemArmand{display:block;width:100%;padding:.25rem 1rem;clear:both;font-weight 48 :400;color:#212529;text-align:inherit;text-decoration:none;white-space:nowrap; 49 background-color:transparent;border:0} 50 51 .nav-linkArmand{display:block;padding:.1rem 0.5rem;text-decoration:none;transition:color 52 .15s ease-in-out,background-color .15s ease-in-out,border-color .15s ease-in-out} 53 54 .dropdown-itemArmand:focus,.dropdown-itemArmand:hover { 55 color:#1e2125; 56 background-color:#BA8989; 57 } 58 59 <!-- Custom styles for this template --> 60 <link href="{{ asset('/bootstrap/signin.css')}}" rel="stylesheet"> 61 <link href="{{ asset('/bootstrap/navbar.css')}}" rel="stylesheet"> 62 </head> 63 <body class="text-center bg-light"> 64 65 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/ 66 font-awesome.min.css" crossorigin="anonymous"> 67 <script src="https://kit.fontawesome.com/64d58efce2.js" crossorigin="anonymous"></script> 68 69 @yield('navbar') 70 71 @yield('content') 72 73 74 75 <script src="{{ asset('/js/bootstrap.bundle.min.js')}}"></script> 76 77 </body> 78 </html></pre>		

ARMAND HOUPIN	navbar.blade.php	1/1
---------------	------------------	-----

```

1 <header>
2   <a href="{{ route('accueil') }}" class="logo">KOLANTR</a>
3   <div class="toggle" onclick="toggleMenu();"></div>
4   <ul class="menu">
5     @guest
6       <li><a href="#Liste" onclick="toggleMenu();" class="login"><i class="fa fa-user" aria-hidden="true"></i></a></li>
7     @endguest
8
9     @auth
10    @if (Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') )
11
12      @if (Auth::user()->hasRole('admin'))
13        <li><a href="{{ route('listeClient_path') }}" onclick="toggleMenu();">Client
14        </a></li>
15      @endif
16
17      <li><a href="{{ route('listeCampagneMesure_path') }}" onclick="toggleMenu();"
18        >Campagne de mesure</a></li>
19      <li><a href="{{ route('listeBoitier_path') }}" onclick="toggleMenu();"
20        >Boitier</a></li>
21      <li><a href="{{ route('liste') }}" onclick="toggleMenu();">Liste</a></li>
22      <!--<li><a href="" onclick="toggleMenu(); class="login"><i class="fa fa-user" aria-hidden="true"></i></a></li>-->
23
24    @elseif(Auth::user()->hasRole('user'))
25      <li><a href="{{ route('clientmescampagnes_path') }}">Mes campagnes de
26        mesures</a></li>
27    @endif
28
29    <li><a href="{{ route('logout') }}" onclick="event.preventDefault(); document.
30      getElementById('logout-form').submit(); class="logout"><i class="fas fa-power-off"></i></a><form id="logout-form" action="{{ route('logout') }}"
31      method="POST" class="d-none">
32        @csrf
33      </form> </li>
34    @endauth
35  </ul>
36 </header>

```

ARMAND HOUPIN	footer.blade.php	1/1
---------------	------------------	-----

```

1 <footer class="copyright">
2   <p>copyright © 2021 KOLANTR. All Right Reserved.</p>
3 </footer>

```

ARMAND HOUPIN	login.blade.php	1/1
<pre> 1 @extends('layouts.master', ['title' => 'Login']) 2 3 @section('navbar') 4 <header> 5 KOLANTR 6 </header> 7 @stop 8 9 @section('content') 10 <main class="form-signin shadow p-3"> 11 <h2 class="featurette-heading"> KOLANTR </h2> 12 <form method="POST" action="{{ route('login') }}> 13 @csrf 14 <hr class="featurette-divider"> 15 <h2 class="featurette-heading fw-normal mb-3 h3">Se 16 connecter</h2> 17 <label for="inputText" class="visually-hidden">Votre Identifiant</label> 18 <input id="identifiant" type="text" class="form-control @error('identifiant' 19) is-invalid @enderror" name="identifiant" value="{{ old('identifiant') }}" 20 required autocomplete="email" autofocus placeholder="Identifiant" style=" 21 border-radius: 3px"> 22 @error('identifiant') 23 24 {{ \$message }} 25 26 @enderror 27 <hr class="featurette-divider-armand"> <!--required autofocus--> 28 <label for="inputPassword" class="visually-hidden">Mot de passe</label> 29 <input id="password" type="password" class="form-control @error('password') 30 is-invalid @enderror" name="password" required autocomplete="current-password" 31 placeholder="Mot de passe" style="border-radius: 3px"> 32 33 @error('password') 34 35 {{ \$message }} 36 37 @enderror <!--required--> 38 39 <div class="checkbox mb-3"> 40 <label> 41 <input class="form-check-input" type="checkbox" name="remember" id="remember" 42 {{ old('remember') ? 'checked' : '' }}> 43 44 <label class="form-check-label text-muted" for="remember"> 45 {{ __('Remember Me') }} 46 </label> 47 </label> 48 </div> 49 <button type="submit" class="w-100 btn btn-lg btn-primary"> 50 {{ __('Login') }} 51 </button> 52 53 @if (Route::has('password.request')) 54 56 {{ __('Forgot Your Password?') }} 57 58 @endif 59 60 </form> 61 62 </main> 63 @endsection </pre>		

```

1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta name="viewport" content="width=device-width, initial-scale=1.0">
5      <meta charset="UTF-8">
6      <title> Accueil </title>
7      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.7.0/css/bootstrap.min.css" crossorigin="anonymous">
8      <script src="https://kit.fontawesome.com/64d58efce2.js" crossorigin="anonymous"></script>
9
10     <link href="{{ asset('/bootstrap/style_accueil.css')}}" rel="stylesheet">
11     <link href="{{ asset('/bootstrap/css_animation_route.css')}}" rel="stylesheet">
12
13 </head>
14 <body>
15
16 <header>
17     <a href="{{ route('accueil') }}" class="logo">KOLANTR</a>
18     <div class="toggle" onclick="toggleMenu();"></div>
19     <ul class="menu">
20         @guest
21             <li><a href="#Liste" onclick="toggleMenu(); class="login"><i class="fa fa-user" aria-hidden="true"></i></a></li>
22         @endguest
23
24         @auth
25             @if (Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien'))
26
27                 @if (Auth::user()->hasRole('admin'))
28                     <li><a href="{{ route('listeClient_path') }}" onclick="toggleMenu();>Client</a></li>
29                 @endif
30
31                     <li><a href="{{ route('listeCampagneMesure_path') }}" onclick="toggleMenu() ;">Campagne de mesure</a></li>
32                     <li><a href="{{ route('listeBoitier_path') }}" onclick="toggleMenu();>Boitier</a></li>
33                     <li><a href="#" onclick="toggleMenu();>Information</a></li>
34                     <li><a href="#" onclick="toggleMenu();>Compte</a></li>
35                     <li><a href="#" onclick="toggleMenu();>Application</a></li>
36                     <li><a href="{{ route('liste') }}" onclick="toggleMenu();>Liste</a></li>
37                     <!--<li><a href="#" onclick="toggleMenu(); class="login"><i class="fa fa-user" aria-hidden="true"></i></a></li>-->
38
39             @elseif(Auth::user()->hasRole('user'))
40                 <li><a href="{{ route('clientmescampagnes_path') }}>Mes campagnes de mesures</a></li>
41                 <li><a href="#" onclick="toggleMenu();>Information</a></li>
42                 <li><a href="#" onclick="toggleMenu();>Compte</a></li>
43                 <li><a href="#" onclick="toggleMenu();>Contact</a></li>
44             @endif
45
46             <li><a href="{{ route('logout') }}" onclick="event.preventDefault(); document.getElementById('logout-form').submit(); class="logout"><i class="fas fa-power-off"></i></a><form id="logout-form" action="{{ route('logout') }}" method="POST" class="d-none">
47                 @csrf
48             </form> </li>
49         @endauth
50     </ul>
51 </header>
52
53 @if(Auth::user()->hasRole('admin') || Auth::user()->hasRole('technicien'))
54 <section class="banner" id="home">
55     <div class="textBx">
56         <h2> Bienvenue sur <br><span>KOLANTR</span></h2>
57         <!--<h3>Développeur Web Front end.</h3>-->
58         <a href="#client" class="btn"> C'est parti </a>
59     </div>
60 </section>
61

```

```

63      @if(Auth::user()->hasRole('admin'))
64      <section class="about" id="client">
65          <div class="heading" >
66              <h2> Client </h2>
67          </div>
68          <div class="content">
69              <div class="contentBx w50 center" >
70                  <h3> Gérer les comptes clients </h3>
71                  <p> Créer, Modifier, Supprimer </p>
72                  <div class="downloadBx">
73                      <button type="button" class="btn" name=""> <a href="{{ route('
74                          listeClient_path') }}" class="lien"> Accéder </a></button>
75                  </div>
76              </div>
77              <div class="w50 center_content">
78                  
79              </div>
80          </div>
81      </section>
82
82      <section class="about background_gray">
83          <div class="heading" >
84              <h2> Administrateur et technicien </h2>
85          </div>
86          <div class="content">
87              <div class="w50 center_content" >
88                  
89              </div>
90              <div class="contentBx w50 center" class="contentBx w50 center" >
91                  <h3> Gérer les comptes clients Administrateur et technicien</h3>
92                  <p> Créer, Modifier, Supprimer </p>
93                  <div class="downloadBx">
94                      <button type="button" class="btn" name=""> <a href="{{ route('
95                          listeAdminTechnicien_path') }}" class="lien">Accéder </a></
96                      button>
97                  </div>
98          </div>
99      </section>
100     @endif
101
102    <section class="about" id="campagne">
103        <div class="heading" >
104            <h2> Campagne de mesure </h2>
105        </div>
106        <div class="content">
107            <div class="contentBx w50 center" class="contentBx w50 center" >
108                <h3> Gérer les campagnes de mesure </h3>
109                <p> Créer, Modifier, Supprimer </p>
110                <div class="downloadBx">
111                    <button type="button" class="btn" name=""> <a href="{{ route('
112                        listeCampagneMesure_path') }}" class="lien">Accéder</a></
113                    button>
114                </div>
115                <div class="w50 center_content" >
116                    
117                </div>
118            </div>
119        </section>
120
121    <section class="about background_gray" id="boitier">
122        <div class="heading" >
123            <h2> Boitier </h2>
124        </div>
125        <div class="content">
126            <div class="w50 center_content" >
127                
128            </div>
129            <div class="contentBx w50 center" class="contentBx w50 center" >
130                <h3> Gérer les boitiers </h3>
131                <p> Créer, Modifier, Supprimer </p>
132                <div class="downloadBx">
133                    <button type="button" class="btn" name=""> <a href="{{ route('
134                        listeBoitier_path') }}" class="lien">Accéder</a> </button>
135                </div>
136            </div>
137        </section>

```

ARMAND HOUPIN	accueil.blade.php	3/6
<pre> 137 </section> 138 139 <!-- Information --> 140 <section class="services" id="information" > 141 <div class="heading white" > 142 <h2>Information</h2> 143 <p> ... </p> 144 </div> 145 <div class="content"> 146 <div class="servicesBx"> 147 148 <h2>Clients</h2> 149 <div class="content"> 150 <div class="infoBx"> 151 <h2> {{ \$nb_compte }} </h2> 152 <h2> {{ \$nb_admin }} </h2> 153 <h2> {{ \$nb_technicien }} </h2> 154 <h2> {{ \$nb_client }} </h2> 155 </div> 156 <div class="infoBx"> 157 <p> Compte Crées </p> 158 <p> Administrateur </p> 159 <p> Technicien </p> 160 <p> Client </p> 161 </div> 162 </div> 163 </div> 164 <div class="servicesBx"> 165 166 <h2>Campagnes de mesure</h2> 167 <div class="content"> 168 <div class="infoBx"> 169 <h2> {{ \$nb_campagne }} </h2> 170 <h2> {{ \$camp_cours }} </h2> 171 <h2> {{ \$camp_fini }} </h2> 172 </div> 173 <div class="infoBx"> 174 <p> Campagne </p> 175 <p> En cours </p> 176 <p> Fini </p> 177 </div> 178 </div> 179 </div> 180 <div class="servicesBx" > 181 182 <h2>Boitiers</h2> 183 <div class="content"> 184 <div class="infoBx"> 185 <h2> {{ \$nb_boitier }} </h2> 186 <h2> {{ \$boiter_utiliser }} </h2> 187 <h2> {{ \$boitier_repos }} </h2> 188 </div> 189 <div class="infoBx"> 190 <p> Boitier </p> 191 <p> En utilisation </p> 192 <p> En pause </p> 193 </div> 194 </div> 195 </div> 196 </div> 197 </section> 198 199 <!-- Compte --> 200 <section class="testimonial" id="compte" > 201 <div class="heading"> 202 <h2>Compte</h2> 203 <p>Retrouver vos informations de compte</p> 204 </div> 205 <div class="content" > 206 <div class="testimonialBx"> 207 <p> Nom : {{ Auth::user()->name }} </p> 208 <p> Prénom : {{ Auth::user()->prenom }} </p> 209 <p> Identifiant : {{ Auth::user()->identifiant }} </p> 210 <p> Adresse Mail : {{ Auth::user()->email }} </p> 211 </div> 212 </div> 213 </section> 214 </pre>		

```

216 ▼      <!-- Application -->
217      <section class="about" id="Application">
218      <div class="content">
219      <div class="contentBx w50 center">
220          <h3> Application </h3>
221          <h2> Télécharger l'application Kolantr config simulation </h2>
222          <p> Développé par Arthur Labarre </p>
223
224      <div class="downloadBx">
225          <p> Version : <span> 0.0.1 </span> </p>
226          <button type="button" class="btn" name=""> <a href="{{ asset('
227              application_simulation_kolantr/
228              Application_simulation_kolantr.rar') }}" download class="lien">
229              Télécharger </a></button>
230      </div>
231
232      <div class="container">
233          <div class="infinite">
234              <div class="shadow"></div>
235          </div>
236          <div class="box_route">
237              <div class="square"> 
238                  class="inverse" > </div>
239          </div>
240      </div>
241
242      </div>
243
244      @include('layouts.partials.footer')
245
246 ▼ @else
247      <section class="banner" id="home">
248      <div class="textBx">
249          <h2> Bienvenue sur <br><span>KOLANTR</span></h2>
250          <!--<h3>Développeur Web Front end.</h3>-->
251          <a href="#client" class="btn"> C'est parti </a>
252      </div>
253  </section>
254
255  <section class="about" id="client">
256      <div class="heading" >
257          <h2> Mes Campagnes de mesures </h2>
258      </div>
259      <div class="content">
260          <div class="contentBx w50 center" >
261              <h3> Visualiser mes campagnes de mesure </h3>
262              <p> Regarder, étudier, télécharger </p>
263              <div class="downloadBx">
264                  <button type="button" class="btn" name=""> <a href="{{ route('
265                      clientmescampagnes_path') }}" class="lien">Accéder </a></button>
266              </div>
267          </div>
268          <div class="w50 center_content" >
269               class="img img_display_600"
270          </div>
271  </section>
272

```

```

273      <!-- Information -->
274      <section class="services" id="information" >
275          <div class="heading white">
276              <h2>Information</h2>
277              <p> ... </p>
278          </div>
279          <div class="content">
280              <div class="servicesBx">
281                  
282                  <h2>Mes campagnes de mesure</h2>
283                  <div class="content">
284                      <div class="infoBx">
285                          <h2> {{ $nb_campagne }} </h2>
286                          <h2> {{ $nb_camp_cours }} </h2>
287                          <h2> {{ $nb_camp_fini }} </h2>
288                      </div>
289                      <div class="infoBx">
290                          <p> Mes campagnes </p>
291                          <p> En cours </p>
292                          <p> Fini </p>
293                      </div>
294                  </div>
295              </div>
296          </div>
297      </section>
298
299      <!-- Compte -->
300      <section class="testimonial" id="compte">
301          <div class="heading">
302              <h2>Compte</h2>
303              <p>Retrouver vos informations de compte</p>
304          </div>
305          <div class="content">
306              <div class="testimonialBx">
307                  <p> Nom : <span> {{ Auth::user()->name }} </span></p>
308                  <p> Prénom : <span> {{ Auth::user()->prenom }} </span></p>
309                  <p> Identifiant : <span>{{ Auth::user()->identifiant }}</span></p>
310                  <p> Adresse Mail : <span>{{ Auth::user()->email }}</span></p>
311                  <p> Numéro téléphone : <span>{{ Auth::user()->numeroTel }}</span></p>
312                  <p> Adresse : <span>{{ Auth::user()->adresseClient }}</span></p>
313                  <p> Code Postal : <span>{{ Auth::user()->codePostal }}</span></p>
314                  <p> Région : <span>{{ Auth::user()->region }}</span></p>
315                  <p> Ville : <span>{{ Auth::user()->ville }}</span></p>
316              </div>
317          </div>
318      </section>
319
320
321      <section class="contact" id="contact">
322          <div class="heading white">
323              <h2>Contact</h2>
324          </div>
325          <div class="content">
326              <div class="contactInfo">
327                  <h3>Contact Info</h3>
328                  <div class="ContactInfoBx">
329                      <div class="box">
330                          <div class="icon">
331                              <i class="fa fa-map-marker"></i>
332                          </div>
333                          <div class="text">
334                              <h3>Adresse</h3>
335                              <p>14 Avenue de l'Université<br>Talence<br>33400 </p>
336                          </div>
337                      </div>
338                      <div class="box">
339                          <div class="icon">
340                              <i class="fa fa-envelope-o"></i>
341                          </div>
342                          <div class="text">
343                              <h3>Email</h3>
344                              <p>kolantr2021snir@gmail.com</p>
345                          </div>
346                      </div>
347                      <div class="box">
348                          <div class="icon">
349                              <i class="fa fa-envelope-o"></i>
350                          </div>
351                          <div class="text">
352                              <h3>Site Web du lycée</h3>
353                              <p><a href="http://www.lyceekastler.fr">http://www.lyceekastler.fr</a></p>
354                          </div>
355                      </div>
356                  </div>
357              </div>
358          </div>

```

ARMAND HOUPIN	accueil.blade.php	6/6
<pre> 359 <div class="formBx"> 360 <form> 361 <h3>Contactez nous (en construction)</h3> 362 <input type="text" name="" placeholder="Prénom" disabled> 363 <input type="email" name="" placeholder="Email" disabled> 364 <textarea placeholder="Votre Message" disabled></textarea> 365 <input type="submit" value="Envoyer" disabled> 366 </form> 367 </div> 368 </div> 369 </section> 370 371 @include('layouts.partials.footer') 372 @endif 373 374 <script> 375 window.addEventListener('scroll', function(){ 376 var header = document.querySelector('header'); 377 header.classList.toggle('sticky', window.scrollY > 0); 378 }); 379 380 function toggleMenu(){ 381 var menuToggle = document.querySelector('.toggle'); 382 var menu = document.querySelector('.menu'); 383 menuToggle.classList.toggle('active'); 384 menu.classList.toggle('active'); 385 } 386 </script> 387 388 </body> 389 </html> </pre>		

ARMAND HOUPIN	alertbatterie.blade.php	1/1
<pre> 1 ▼ <div id="popup1" class="overlay"> 2 ▼ <div class="popup"> 3 <h2>Batterie Faible</h2> 4 &times; 5 ▼ <div class="content"> 6 Les batteries citées en dessous sont bientôt vides ! 7
 8 ▼ @if(!empty(\$alarme)) 9 @foreach(\$alarme as \$alarmes) 10 @if(!empty(\$alarmes->alarmeBatterie)) 11 - {{\$alarmes->adrSigFox}}
 12 @endif 13 @endforeach 14 @endif 15 </div> 16 </div> 17 </div> </pre>		

ARMAND HOUPIN	liste.blade.php	1/4
<pre> 1 @extends('layouts.home', ['title' => 'Liste']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css')}}" rel="stylesheet"> 6 <link href="{{ asset('/bootstrap/dashboard.css')}}" rel="stylesheet"> 7 <link href="{{ asset('/bootstrap/navbar.css')}}" rel="stylesheet"> 8 @stop 9 10 @section('content') 11 12 13 @if(!empty(\$alert)) 14 <div class="alert success text-center" id="alert"> 15 &times; 16 Succès ! 17 <hr> 18 {!! \$messagealert !!} 19 </div> 20 @endif 21 22 <!--<div class="containerForm-Validation" id="chargement"> 23 <main> 24 25 <div class="py-5 text-center"> 26 <label for="firstName" class="mx-auto mb-1 py-5 featurette-heading"><div 27 class="spinner-border text-light" role="status" id="bar"> 28 Loading... 29 </div> 30 </main> 31 </div>--> 32 33 <div class="containerForm-Validation"> 34 <main> 35 36

 37 <div class="row g-3 shadow p-3"> 38 <div class="col-md-7 col-lg-12 mx-auto "> 39 <div class="row g-3"> 40 <div class="col-md-8"> 41 <svg xmlns= 42 "http://www.w3.org/2000/svg" width="30" height="35" fill="currentColor" 43 class="bi bi-person-plus" viewBox="0 0 16 16"> 44 <path d="M6 8a3 3 0 1 0 0-6 3 3 0 0 0 6zm2-3a2 2 0 1 1-4 0 2 2 0 0 1 4 45 0zm4 8c0 1-1 1-1 1H1s-1 0-1-1 1-4 6-4 6 3 6 46 4zm-1-.004c-.001-.246-.154-.986-.832-1.664C9.516 10.68 8.289 10 6 10c-2.29 47 0-3.516.68-4.168 1.332-.678.678-.83 1.418-.832 1.664h10z"/> 48 <path fill-rule="evenodd" d="M13.5 5a.5.5 0 0 1 .5.5V7h1.5a.5.5 0 0 1 0 49 1H14v1.5a.5.5 0 0 1-1 0V8h-1.5a.5.5 0 0 1 0-1H13V5.5a.5.5 0 0 1 .5-.5z"/> 50 </svg> 51 </div> 52 <div class="col-md-1"> 53 <svg xmlns="http://www.w3.org/2000/svg" width="30" height="35" fill= 54 "currentColor" class="bi bi-search tx-black" viewBox="0 0 16 16"> 55 <path d="M11.742 10.344a6.5 6.5 0 1 0-1.397 56 1.398h-.001c.03.04.062.078.098.115l3.85 3.85a1 1 0 0 0 57 1.415-1.414l-3.85-3.85a1.007 1.007 0 0 0-.115-.1zM12 6.5a5.5 5.5 0 1 58 0-11 0 5.5 5.5 0 0 1 11 0z"/> 59 </div> 60 </div> 61 </div> 62 </div> 63 <hr> 64 <div class="row g-3 shadow p-3"> 65 <div class="col-md-7 col-lg-12 mx-auto "> 66 <div class="row g-3"> </pre>		

ARMAND HOUPIN	liste.blade.php	2/4
51 52 <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eighth navbar example" 53 > 54 <div class="container"> 55 <div class="col-md-4"> 56 <label for="firstName" class="form-label text-muted">Adresse Campagne</label> 57 58 </div> 59 60 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarsExample00" aria-controls="navbarsExample00" aria-expanded="false" aria-label="Toggle navigation"> 61 62 </button> 63 <div class="collapse navbar-collapse" id="navbarsExample00"> 64 <div class="col-md-5"> 65 <label for="firstName" class="form-label text-muted">@sigfox</label> 66 </div> 67 <div class="col-md-5"> 68 <label for="firstName" class="form-label text-muted">Client</label> 69 </div> 70 <div class="col-md-2"> 71 <label for="firstName" class="form-label text-muted">statut</label> 72 </div> 73 </div> 74 <hr class="my-4 tx-black"> 75 @foreach (\$campagne as \$campagne) 76 <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eighth navbar example"> 77 > 78 <div class="container" style="border-bottom: 1px solid @if(\$campagne->statut == 'fini') red @else green @endif"> 79 <div class="col-md-4"> 80 <label for="firstName" class="form-label tx-black"> {!! \$campagne-> 81 adresseCampagne !!} \$campagne->id])}}><i class="far fa-edit"></i> \$campagne 84 ->id])}}" style="color: #FC0000"><i class="fas fa-trash"></i> </label> 85 > 86 </div> 87 88 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarsExample" aria-controls="navbarsExample" aria-expanded="false" aria-label="Toggle navigation"> 89 90 </button> 91 <div class="collapse navbar-collapse" id="navbarsExample"> 92 <div class="col-md-5"> 93 @if(!empty(\$campagne->id_boitier)) 94 <label for="firstName" class="form-label tx-black">> 95 @foreach(\$listeboitier as \$listeboitiers) 96 @if(\$campagne->id_boitier == \$listeboitiers->id) 97 {{ \$listeboitiers->adrSigfox }} 98 @endif 99 @endforeach 100 \$campagne->id_boitier])}}> 101 <i class="far fa-edit"></i> 102 \$campagne->id_boitier])}}" style="color: #FC0000"><i class="fas 104 fa-trash"></i> 105 </label> 106 @else 107 @endif 108 @endfor		

```

114         <div class="dropdown">
115             <a class="nav-linkArmand" href="#" id="dropdown03" data-bs-toggle="dropdown" aria-expanded="false">
116                 <i class="far fa-plus-square"></i>
117             </a>
118             <ul class="dropdown-menu mb-5" aria-labelledby="dropdown03">
119                 @foreach($listeboitier as $listeboitiers)
120                     @if($listeboitiers->statut == "non utilisé")
121                         <li><a class="dropdown-itemblack" href="{{ route('
122                             listeajouterboitier_path', ['id_campagne' => $campagne->id,
123                             'id_boitier' => $listeboitiers ])}}"> {{ $listeboitiers->adrSigfox
124                         }}</a></li>
125                     @endif
126                 @endforeach
127             </ul>
128         </div>
129         <div class="col-md-5">
130             <label for="firstName" class="form-label tx-black">
131                 @if(!empty($campagne->id_user)) @foreach($listeclient as $listeclients)
132                     @if($campagne->id_user == $listeclients->id)
133                         {{ $listeclients->name}} {{ $listeclients->prenom}}
134                     @endif
135                 @endforeach @else Aucun client associer @endif</label>
136             </div>
137             <div class="col-md-2">
138
139                 <div class="dropdown">
140                     <a class="nav-linkArmand" href="#" id="dropdown03" data-bs-toggle="dropdown" aria-expanded="false">
141                         <span class="tx-black" >{!! $campagne->statut !!}</span> <i class="fas
142                         fa-angle-down"></i>
143                     </a>
144                     <ul class="dropdown-menu mb-5" aria-labelledby="dropdown03">
145                         <li><a class="dropdown-itemblack tx-black" href="{{ route('
146                             modifierListestatut_encours_path', ['id' => $campagne->id ])}}">
147                             En Cours</a></li>
148                         <li><a class="dropdown-itemblack tx-black" href="{{ route('
149                             modifierListestatut_fini_path', ['id' => $campagne->id ])}}">
150                             Fini</a></li>
151                     </ul>
152                 </div>
153             </div>
154             <hr class="my-4 text-muted">
155         </div>
156     </div>
157     </div>
158
159 </main>
160
161     @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or $title !=
162         "Home" )
163
164         @include('layouts.partials.footer')
165
166     @endif
167
168 </div>
169
170 @stop

```

ARMAND HOUPIN	liste.blade.php	4/4
<pre>172 @section('script') 173 <script> 174 175 function confirmdelete() { 176 if(!confirm("Êtes-vous sûr de vouloir le supprimer ?")) 177 event.preventDefault(); 178 } 179 180 var close = document.getElementsByClassName("closebtn"); 181 var i; 182 183 for (i = 0; i < close.length; i++) { 184 close[i].onclick = function(){ 185 var div = this.parentElement; 186 div.style.opacity = "0"; 187 setTimeout(function(){ div.style.display = "none"; }, 600); 188 } 189 } 190 191 var myVar = setInterval(myTimer, 2500); 192 193 function myTimer() { 194 var div = document.getElementById("alert"); 195 div.style.opacity = "0"; 196 setTimeout(function(){ div.style.display = "none"; }, 600); 197 } 198 </script> 199 200 <script src="{{ asset('/js/form-validation.js')}}"></script> 201 <!!--<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/ jquery.min.js"></script>--> 202 <!--<script> 203 \$(document).ready(function(){ 204 \$("#chargement").fadeOut(1500, function(){ 205 \$("#bloc").fadeIn(1000) 206 }); 207 }); 208 </script>--> 209 @stop</pre>		

ARMAND HOUPIN	consultationcampagne.blade.php	1/3
<pre> 1 @extends('layouts.home', ['title' => 'Gérer Campagne Mesure']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css')}}" rel="stylesheet"> 6 <link rel="preconnect" href="https://fonts.gstatic.com"> 7 <link href="https://fonts.googleapis.com/css2?family=Chango&display=swap" rel="stylesheet" 8 "> 9 10 <style> 11 .modal-window{ 12 display: none; 13 position: fixed; 14 z-index: 2; 15 left: 0; 16 top: 0; 17 right: 0; 18 bottom: 0; 19 text-align: center; 20 background: rgba(0,0,0,0.5); 21 } 22 23 .modal-window .iframe{ 24 position: absolute; 25 z-index: 3; 26 margin: auto; 27 cursor: crosshair; 28 width: 100%; 29 height: 100%; 30 } 31 32 .show-popup{ 33 display: block; 34 } 35 36 canvas{ 37 display: block; 38 text-align: center; 39 } 40 .iframe h1 { 41 position: absolute; 42 color: #fff; 43 top: 50%; 44 left: 50%; 45 transform: translate(-50%, -50%); 46 font-family: "Source Sans Pro"; 47 font-size: 75px; 48 font-weight: 1000; 49 -webkit-user-select: none; 50 user-select: none; 51 } 52 53 .iframe h6 { 54 position: absolute; 55 top: 60%; 56 left: 50%; 57 transform: translate(-50%, -50%); 58 color: #fff; 59 font-family: 'Chango', cursive; 60 font-size: 75px; 61 font-weight: 900; 62 -webkit-user-select: none; 63 user-select: none; 64 } </pre>		

```

66 .iframe a{
67   position: absolute;
68   top: 70%;
69   left: 50%;
70   transform: translate(-50%, -50%);
71   color: #fff;
72   outline: none;
73   text-decoration: none;
74   font-family: "Source Sans Pro";
75   font-size: 35px;
76   font-weight: 900;
77   -webkit-user-select: none;
78   user-select: none;
79 }
80
81 @media(max-width: 600px)
82 {
83   .iframe h1 {
84     font-size: 2em;
85     font-weight: 500;
86   }
87
88 .iframe h6 {
89   top: 65%;
90   font-size: 2em;
91   font-weight: 900;
92 }
93
94 .iframe a{
95   top: 73%;
96   font-size: 1.5em;
97   font-weight: 900;
98 }
99
100 }
101
102 @media (max-width: 1500px) {
103   .iframe h1 {
104     font-size: 2.5em;
105     font-weight: 500;
106   }
107
108 .iframe h6 {
109   font-size: 2.5em;
110   font-weight: 900;
111 }
112
113 .iframe a{
114   font-size: 2em;
115   font-weight: 900;
116 }
117
118 }
119
120 </style>
121 @stop
122
123 @section('content')
124
125 <script src="https://code.highcharts.com/highcharts.js"></script>
126 <script src="https://code.highcharts.com/highcharts-more.js"></script>
127 <script src="https://code.highcharts.com/modules/data.js"></script>
128 <script src="https://code.highcharts.com/modules/exporting.js"></script>
129 <script src="https://code.highcharts.com/modules/export-data.js"></script>
130 <script src="https://code.highcharts.com/modules/accessibility.js"></script>
131 <script src="https://code.highcharts.com/highcharts-3d.js"></script>
132 <script src="https://code.highcharts.com/stock/highstock.js"></script>
133 <script src="https://code.highcharts.com/stock/modules/data.js"></script>
134 <script src="https://code.highcharts.com/stock/modules/exporting.js"></script>
135 <script src="https://code.highcharts.com/stock/modules/export-data.js"></script>
136

```



ARMAND HOUPIN	consultationcampagne.blade.php	3/3
<pre>138 @if(!empty(\$stat_true)) 139 @include('Client/stat_true') 140 @else 141 @include('Client/stat_false') 142 @endif 143 144 @stop 145 146 ▼ @section('script') 147 148 <script src="{{ asset('/js/form-validation.js')}}"></script> 149 150 @stop</pre>		

```

1 ▼ <div class="row featurette shadow p-3 mb-5">
2 ▼   <div class="col-md-13">
3     <h2 class="featurette-heading tx-black"> Graphique camion</h2>
4     <hr>
5 ▼   <div class="row">
6     <div class="col-lg-4" style="width: 33%;>
7       <button type="button" class="w-100 btn btn-sm btn-outline-secondary"
8         onclick="AfficherGraphiqueLigne_camion();"> Graphique Ligne</button>
9     </div>
10    <div class="col-lg-4" style="width: 33%;>
11      <button type="button" class="w-100 btn btn-sm btn-outline-secondary"
12        onclick="AfficherGraphiqueCercle_camion();"> Graphique Cercle </button>
13    </div>
14    <div class="col-lg-4" style="width: 33%;>
15      <button type="button" class="w-100 btn btn-sm btn-outline-secondary"
16        onclick="AfficherGraphiqueBarre_camion();"> Graphique Barre </button>
17    </div>
18  </div>
19  <hr>
20    <div id="graphiques_lignes_camion" style="display: block;"></div>
21    <div id="graphiques_cercle_camion" style="display: none;"></div>
22    <div id="graphiques_barres_camion" style="display: none;"></div>
23
24 <script type="text/javascript">
25
26 var Campagne = {!! json_encode($campagne->adresseCampagne) !!};
27 var Limitation = {!! json_encode($campagne->limitationvitesse) !!};
28 var statVitMoyen = {!! json_encode($statVitMoyen_PL) !!};
29 var statHeureStat = {!! json_encode($statHeureStat_PL) !!};
30 var statVitesseInferieurOuEgale = {!! json_encode($statVitesseInferieurOuEgale_PL) !!};
31 var statVitesseLimitMoins20 = {!! json_encode($statVitesseLimitMoins20_PL) !!};
32 var statVitesseLimitPlus20 = {!! json_encode($statVitesseLimitPlus20_PL) !!};
33 var statVitesseLimitPlus30 = {!! json_encode($statVitesseLimitPlus30_PL) !!};
34 var statVitesseLimitPlus40 = {!! json_encode($statVitesseLimitPlus40_PL) !!};
35 var statVitesseLimitPlus50 = {!! json_encode($statVitesseLimitPlus50_PL) !!};
36 var vit1 = 0;
37 var vit2 = 0;
38 var vit3 = 0;
39 var vit4 = 0;
40 var vit5 = 0;
41 var vit6 = 0;
42
43 var compt = 0;
44
45 for (var i in statVitesseInferieurOuEgale)
46 {
47   vit1 = vit1 + statVitesseInferieurOuEgale[i];
48   compt = compt + 1;
49 }
50 for (var i in statVitesseLimitMoins20)
51 {
52   vit2 = vit2 + statVitesseLimitMoins20[i];
53 }
54 for (var i in statVitesseLimitPlus20)
55 {
56   vit3 = vit3 + statVitesseLimitPlus20[i];
57 }
58 for (var i in statVitesseLimitPlus30)
59 {
60   vit4 = vit4 + statVitesseLimitPlus30[i];
61 }
62 for (var i in statVitesseLimitPlus40)
63 {
64   vit5 = vit5 + statVitesseLimitPlus40[i];
65 }
66 for (var i in statVitesseLimitPlus50)
67 {
68   vit6 = vit6 + statVitesseLimitPlus50[i];
69 }
70

```



```

71 vit1 = vit1 / compt;
72 vit2 = vit2 / compt;
73 vit3 = vit3 / compt;
74 vit4 = vit4 / compt;
75 vit5 = vit5 / compt;
76 vit6 = vit6 / compt;
77
78
79 ▼ Highcharts.chart('graphiques_lignes_camion', {
80   chart: {
81     type: 'line'
82   },
83   title: {
84     text: 'Limitation : ' + Limitation + ' km/h',
85   },
86   xAxis: {
87     categories: statHeureStat,
88   },
89   yAxis: {
90     title: {
91       text: 'Nombre de camions'
92     },
93     plotLines: [{{
94       value: Limitation,
95       color: 'red',
96       width: 1,
97       label: {
98         text: Limitation
99       }
100     }]}
101 },
102 ▼ plotOptions: {
103   line: {
104     dataLabels: {
105       enabled: true
106     },
107     enableMouseTracking: false
108   }
109 },
110 ▼ series: [{{
111   name: 'VitesseInferieurOuEgale',
112   data: statVitesseInferieurOuEgale,
113 }, {
114   name: 'VitesseLimitMoins20',
115   data: statVitesseLimitMoins20,
116 }, {
117   name: 'VitesseLimitplus20',
118   data: statVitesseLimitPlus20,
119 }, {
120   name: 'Vitesselimitplus30',
121   data: statVitesselimitPlus30,
122 }, {
123   name: 'Vitesselimitplus40',
124   data: statVitesselimitPlus40,
125 }, {
126   name: 'Vitesselimitplus50',
127   data: statVitesselimitPlus50,
128 }]}
129 });
130

```



```

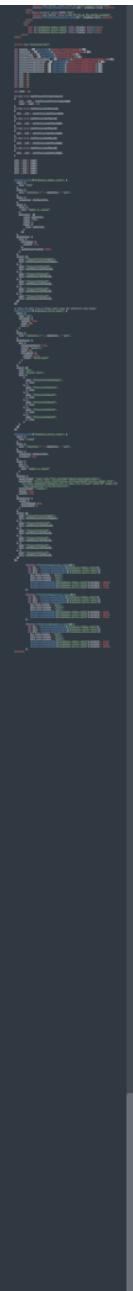
131 // Parse the data from an inline table using the Highcharts Data plugin
132 ▼ Highcharts.chart('graphiques_cercle_camion', {
133 ▼   chart: {
134     type: 'pie',
135 ▼       options3d: {
136       enabled: true,
137       alpha: 45,
138       beta: 0
139     }
140   },
141   title: {
142     text: 'Limitation : ' + Limitation + ' km/h',
143   },
144 ▼   plotOptions: {
145 ▼     pie: {
146       allowPointSelect: true,
147       cursor: 'pointer',
148       depth: 35,
149 ▼       dataLabels: {
150         enabled: true,
151         format: '{point.name}'
152       }
153     }
154   },
155 ▼   series: [
156     type: 'pie',
157     name: 'Browser share',
158 ▼     data: [
159 ▼       {
160         name: 'VitesseInferieurOuEgale',
161         y: vit1,
162       },
163       {
164         name: 'VitesseLimitMoins20',
165         y: vit2,
166       },
167       {
168         name: 'VitesseLimitplus20',
169         y: vit3,
170       },
171       {
172         name: 'VitesseLimitplus30',
173         y: vit4,
174       },
175       {
176         name: 'VitesseLimitplus40',
177         y: vit5,
178       },
179       {
180         name: 'VitesseLimitplus50',
181         y: vit6,
182       }
183     ]
184   ],
185 });
186 });
187 
```



```

182 ▼ Highcharts.chart('graphiques_barres_camion', {
183   chart: {
184     type: 'column'
185   },
186   title: {
187     text: 'Limitation : ' + Limitation + ' km/h',
188   },
189 ▼ xAxis: {
190   categories: statHeureStat,
191   crosshair: true
192 },
193 ▼ yAxis: {
194   min: 0,
195   title: {
196     text: 'Nombre de camions'
197   }
198 },
199 ▼ tooltip: {
200   headerFormat: '<span style="font-size:10px">{point.key}</span><table>',
201   pointFormat: '<tr><td style="color:{series.color};padding:0">{series.name}: </td>' +
202     '<td style="padding:0"><b>{point.y:f} <i class="fas fa-truck" width="40" height="40" fill="currentColor"></i></b></td></tr>',
203   footerFormat: '</table>',
204   shared: true,
205   useHTML: true
206 },
207 ▼ plotOptions: {
208   column: {
209     pointPadding: 0.2,
210     borderWidth: 0
211   }
212 },
213 ▼ series: [
214   {
215     name: 'VitesseInferieurOuEgale',
216     data: statVitesseInferieurOuEgale,
217   }, {
218     name: 'VitesseLimitMoins20',
219     data: statVitesseLimitMoins20,
220   }, {
221     name: 'VitesseLimitplus20',
222     data: statVitesseLimitPlus20,
223   }, {
224     name: 'VitesseLimitplus30',
225     data: statVitesseLimitPlus30,
226   }, {
227     name: 'VitesseLimitplus40',
228     data: statVitesseLimitPlus40,
229   }, {
230     name: 'VitesseLimitplus50',
231   }]
232 ▼ });
233

```



```
234     function AfficherGraphiqueLigne_camion() {
235         var div = document.getElementById("graphiques_lignes_camion");
236         var div2 = document.getElementById("graphiques_cercle_camion");
237         var div3 = document.getElementById("graphiques_barres_camion");
238         div.style.display = "block";
239         div2.style.display = "none";
240         div3.style.display = "none";
241         document.getElementById("graphiques_lignes_camion").disabled = false;
242         document.getElementById("graphiques_cercle_camion").disabled = true;
243         document.getElementById("graphiques_barres_camion").disabled = true;
244     };
245
246     function AfficherGraphiqueCercle_camion() {
247         var div = document.getElementById("graphiques_lignes_camion");
248         var div2 = document.getElementById("graphiques_cercle_camion");
249         var div3 = document.getElementById("graphiques_barres_camion");
250         div.style.display = "none";
251         div2.style.display = "block";
252         div3.style.display = "none";
253         document.getElementById("graphiques_lignes_camion").disabled = true;
254         document.getElementById("graphiques_cercle_camion").disabled = false;
255         document.getElementById("graphiques_barres_camion").disabled = true;
256     };
257
258     function AfficherGraphiqueBarre_camion() {
259         var div = document.getElementById("graphiques_lignes_camion");
260         var div2 = document.getElementById("graphiques_cercle_camion");
261         var div3 = document.getElementById("graphiques_barres_camion");
262         div.style.display = "none";
263         div2.style.display = "none";
264         div3.style.display = "block";
265         document.getElementById("graphiques_lignes_camion").disabled = true;
266         document.getElementById("graphiques_cercle_camion").disabled = true;
267         document.getElementById("graphiques_barres_camion").disabled = false;
268     };
269 </script>
```

ARMAND HOUPIN	listecampagneclient.blade.php	1/2
<pre> 1 @extends('layouts.home', ['title' => 'Gérer Campagne Mesure']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css') }}" rel="stylesheet"> 6 @stop 7 8 @section('content') 9 10 11 <div class="containerForm-Validation"> 12 <main> 13 <div class="py-5 text-center"> 14 <h2 class="mx-auto mb-1 py-5 featurette-heading"><i class="fas fa-route" width="75" height="75" fill="currentColor" ></i> Mes campagnes de 15 mesure <i class="fas fa-route" width="75" height="75" fill="currentColor" ></i> </ 16 span></h2> 17 18 </div> 19 <hr class="tx-white"> 20 <div class="row g-3 shadow p-3"> 21 <div class="col-md-7 col-lg-12 mx-auto"> 22 <div class="row g-3"> 23 24 <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eighth navbar example"> 25 <div class="container"> 26 <div class="col-md-4"> 27 <label for="firstName" class="form-label text-muted">Adresse</label>
 28 </div> 29 30 <div class="col-md-2"> 31 <label for="firstName" class="form-label text-muted">Ville</label>
 32 </div> 33 34 <button class="navbar-toggler tx-black" type="button" data-bs-toggle="collapse" 35 data-bs-target="#navbarsExample00" aria-controls="navbarsExample00" aria-expanded="false" 36 aria-label="Toggle navigation"> 37 <i class="fas fa-bars tx-black" ></i> 38 </button> 39 <div class="collapse navbar-collapse" id="navbarsExample00"> 40 <div class="col-md-5"> 41 <label for="firstName" class="form-label text-muted">Date début</label>
 42 </div> 43 <div class="col-md-3"> 44 <label for="firstName" class="form-label text-muted">Statut</label> 45 </div> 46 <div class="col-md-3"> 47 <label for="firstName" class="form-label text-muted">limitation</label> 48 </div> 49 </div> 50 </div> 51 </div> 52 <hr class="my-4 tx-black"> 53 54 <!-- foreach ici --> 55 @foreach(\$listecampagne as \$listecampagne) 56 @if(\$listecampagne->id_user == Auth::user()->id) 57 <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eighth navbar example"> 58 <div class="container"> 59 60 <div class="col-md-4"> 61 <label for="firstName" class="form-label tx-black" >{!! \$listecampagne-> 62 adresseCampagne !!}</label> 63 </div> 64 65 <div class="col-md-2"> 66 <label for="firstName" class="form-label tx-black" >{!! \$listecampagne->ville 67 !!}</label> 68 </div> 69 </div> 70 </div> 71 @endif 72 </div> 73 </div> 74 </div> 75 </div> 76 77 </pre> 	1/2	

ARMAND HOUPIN	listecampagneclient.blade.php	2/2
	<pre> 67 68 <button class="navbar-toggler tx-black" type="button" data-bs-toggle="collapse" 69 data-bs-target="#navbarsExample{{\\$listecampagne->id}}" aria-controls=" 70 navbarsExample{{\\$listecampagne->id}}" aria-expanded="false" aria-label="Toggle 71 navigation"> 72 <i class="fas fa-bars tx-black"></i> 73 </button> 74 75 <div class="collapse navbar-collapse" id="navbarsExample{{\\$listecampagne->id}}"> 76 <div class="col-md-5"> 77 <label for="firstName" class="form-label tx-black"> {!! \\$listecampagne-> 78 DebutCampagne !!}</label> 79 </div> 80 <div class="col-md-3"> 81 <label for="firstName" class="form-label tx-black"> {!! \\$listecampagne-> 82 statut !!}</label> 83 </div> 84 <div class="col-md-3"> 85 <label for="firstName" class="form-label tx-black"> {!! \\$listecampagne-> 86 limitationvitesse !!}</label> 87 </div> 88 <div class="col-md-1"> 89 <label for="firstName" class="form-label tx-black"> 90 \\$listecampagne->id])}" 91 ><i class="fas fa-search-plus"></i> 92 </label> 93 </div> 94 </div> 95 </div> 96 <hr class="my-4 text-muted"> 97 @endif 98 @endforeach 99 <!--fin foreach --> 100 101 <!--<hr class="my-4 text-muted">--> 102 </div> 103 104 105 </div> 106 107 @stop 108 109 @section('script') 110 111 <script src="{{ asset('/js/form-validation.js') }}></script> 112 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script> 113 @stop </pre>	

```

1  <button type="button" id="modal-btn" class="w-100 btn btn-sm btn-outline-secondary">
2    HighScore </button>
3  <div id="modal" class="modal-window">
4    <div class="iframe">
5      <h1 style="color:white">La vitesse max est de </h1>
6      <h6 style="color:white"> {{ $vitmax }} km/h </h6>
7      <canvas id="birthday"> </canvas>
8      <a href="#" id="btn-close"> close </a>
9    </div>
10   </div>
11
12  <script>
13    let elBtn = document.getElementById("modal-btn");
14    let elModal = document.getElementById("modal");
15    let close = document.getElementById("btn-close");
16
17    elBtn.onclick = function() {
18      console.log('click: btn: ' + this.id + ', modal: ' + elModal.id);
19      elModal.classList.toggle('show-popup');
20
21    }
22    /* to dispose the popup on click */
23    close.onclick = function() {
24      elModal.classList.toggle('show-popup');
25    }
26  </script>
27
28  <script>
29  const PI2 = Math.PI * 2
30  const random = (min, max) => Math.random() * (max - min + 1) + min | 0
31  const timestamp = _ => new Date().getTime()
32
33 // container
34  class Birthday {
35    constructor() {
36      this.resize()
37
38      // create a lovely place to store the firework
39      this.fireworks = []
40      this.counter = 0
41    }
42
43    resize() {
44      this.width = canvas.width = window.innerWidth
45      let center = this.width / 2 | 0
46      this.spawnA = center - center / 4 | 0
47      this.spawnB = center + center / 4 | 0
48
49      this.height = canvas.height = window.innerHeight
50      this.spawnC = this.height * .1
51      this.spawnD = this.height * .5
52
53    }
54
55    onClick(evt) {
56      let x = evt.clientX || evt.touches && evt.touches[0].pageX
57      let y = evt.clientY || evt.touches && evt.touches[0].pageY
58
59      let count = random(3,5)
60      for(let i = 0; i < count; i++) this.fireworks.push(new Firework(
61        random(this.spawnA, this.spawnB),
62        this.height,
63        x,
64        y,
65        random(0, 260),
66        random(30, 110)))
67
68      this.counter = -1
69
70    }
71  }
72

```

ARMAND HOUPIN	popup_highscore.blade.php	2/3
<pre> 72 73 update(delta) { 74 ctx.globalCompositeOperation = 'source-over' 75 ctx.fillStyle = `rgba(20,20,20,0.5)` 76 ctx.fillRect(0, 0, this.width, this.height) 77 78 ctx.globalCompositeOperation = 'lighter' 79 for (let firework of this.fireworks) firework.update(delta) 80 81 // if enough time passed... create new new firework 82 this.counter += delta * 3 // each second 83 if (this.counter >= 1) { 84 this.fireworks.push(new Firework(85 random(this.spawnA, this.spawnB), 86 this.height, 87 random(0, this.width), 88 random(this.spawnC, this.spawnD), 89 random(0, 360), 90 random(30, 110))) 91 this.counter = 0 92 } 93 94 // remove the dead fireworks 95 if (this.fireworks.length > 1000) this.fireworks = this.fireworks.filter(firework => ! 96 firework.dead) 97 } 98 } 99 100 class Firework { 101 constructor(x, y, targetX, targetY, shade, offsprings) { 102 this.dead = false 103 this.offsprings = offsprings 104 105 this.x = x 106 this.y = y 107 this.targetX = targetX 108 this.targetY = targetY 109 110 this.shade = shade 111 this.history = [] 112 } 113 update(delta) { 114 if (this.dead) return 115 116 let xDiff = this.targetX - this.x 117 let yDiff = this.targetY - this.y 118 if (Math.abs(xDiff) > 3 Math.abs(yDiff) > 3) { // is still moving 119 this.x += xDiff * 2 * delta 120 this.y += yDiff * 2 * delta 121 122 this.history.push({ 123 x: this.x, 124 y: this.y 125 }) 126 127 if (this.history.length > 20) this.history.shift() 128 129 } else { 130 if (this.offsprings && !this.madeChilds) { 131 132 let babies = this.offsprings / 2 133 for (let i = 0; i < babies; i++) { 134 let targetX = this.x + this.offsprings * Math.cos(PI2 * i / babies) 0 135 let targetY = this.y + this.offsprings * Math.sin(PI2 * i / babies) 0 136 137 birthday.fireworks.push(new Firework(this.x, this.y, targetX, targetY, this.shade 138 , 0)) 139 } 140 } 141 } </pre>		

ARMAND HOUPIN	popup_highscore.blade.php	3/3
<pre> 142 this.madeChilds = true 143 this.history.shift() 144 } 145 146 if (this.history.length === 0) this.dead = true 147 else if (this.offsprings) { 148 for (let i = 0; this.history.length > i; i++) { 149 let point = this.history[i] 150 ctx.beginPath() 151 ctx.fillStyle = `hsl(\${this.shade},100%,\${i + '%'})' 152 ctx.arc(point.x, point.y, 1, 0, PI2, false) 153 ctx.fill() 154 } 155 } else { 156 ctx.beginPath() 157 ctx.fillStyle = `hsl(\${this.shade},100%,50%)' 158 ctx.arc(this.x, this.y, 1, 0, PI2, false) 159 ctx.fill() 160 } 161 } 162 } 163 } 164 165 let canvas = document.getElementById('birthday') 166 let ctx = canvas.getContext('2d') 167 168 let then = timestamp() 169 170 let birthday = new Birthday() 171 172 window.onresize = () => birthday.resize() 173 document.onclick = evt => birthday.onClick(evt) 174 document.ontouchstart = evt => birthday.onClick(evt) 175 176 177 ;(function loop(){ 178 requestAnimationFrame(loop) 179 180 let now = timestamp() 181 let delta = now - then 182 183 then = now 184 birthday.update(delta / 500) 185 186 }) 187 188 </script> </pre>		

ARMAND HOUPIN	stat_false.blade.php	1/1
	<pre><main> <div id="myCarousel" class="carousel slide" data-bs-ride="carousel"> </div> <!-- Marketing messaging and featurettes ===== --> <!-- Wrap the rest of the page in another container to center all the content. --> <div class="container marketing"> <!-- Three columns of text below the carousel --> <hr class="p-3 mb-5 tx-white"> <div class="row featurette shadow p-3 mb-5"> <div class="col-md-12"> <h2 class="featurette-heading tx-black">Campagne de mesure </h2> <p class="tx-black">> Adresse de la campagne : {{ \$campagne->adresseCampagne }} </p> <p class="tx-black">> Ville de la campagne : {{ \$campagne->ville }} </p> <p class="tx-black">> Code postal de la campagne : {{ \$campagne->codePostal }} </p> <p class="tx-black">> Statut de la campagne : @if(\$campagne->statut == "en cours") < span style="color : green;"> {{ \$campagne->statut }}<0xa0> @else {{ \$campagne->statut }} @endif</p> <p class="tx-black"> Date début de la campagne : {{ \$campagne->DébutCampagne }} </p> <p class="tx-black"> Limitation vitesse : {{ \$campagne->limitationvitesse }} </p> </div> </div> <hr class="featurette-divider tx-black"> <!-- START THE FEATURETTES --> <div class="row featurette shadow p-3 mb-5"> <div class="col-md-13"> <h2 class="featurette-heading tx-black">Pas de données</h2> </div> </div> <hr class="featurette-divider tx-black"> @include('layouts.partials.footer') </div> </main></pre>	

```

1 ▼ <main>
2     |   <div id="myCarousel" class="carousel slide" data-bs-ride="carousel">
3     |   </div>
4
5
6     
8     <!-- Wrap the rest of the page in another container to center all the content. --&gt;
9
10 ▼ &lt;div class="container marketing"&gt;
11
12     <!-- Three columns of text below the carousel --&gt;
13 ▼ &lt;hr class="p-3 mb-5 tx-white"&gt;
14 ▼   &lt;div class="row featurette shadow p-3 mb-5"&gt;
15 ▼     &lt;div class="col-md-12"&gt;
16         &lt;h2 class="featurette-heading tx-black"&gt;Campagne de mesure &lt;/h2&gt;
17         &lt;p class="tx-black"&gt; Adresse de la campagne : {{ $campagne-&gt;adresseCampagne }} &lt;/p&gt;
18         &lt;p class="tx-black"&gt; Ville de la campagne : {{ $campagne-&gt;ville }} &lt;/p&gt;
19         &lt;p class="tx-black"&gt; Code postal de la campagne : {{ $campagne-&gt;codePostal }} &lt;/p&gt;
20         &lt;p class="tx-black"&gt; Statut de la campagne : @if($campagne-&gt;statut == "en cours") &lt;
21             span style="color : green;"&gt; {{ $campagne-&gt;statut }}@else &lt;span style="color
22             : red;"&gt; {{ $campagne-&gt;statut }}@endif&lt;/p&gt;
23         &lt;p class="tx-black"&gt; Date début de la campagne : {{ $campagne-&gt;DébutCampagne }} &lt;/p&gt;
24         &lt;p class="tx-black"&gt; Limitation vitesse : {{ $campagne-&gt;limitationvitesse }} &lt;/p&gt;
25     &lt;/div&gt;
26     &lt;div class="row"&gt;
27         &lt;div class="col-lg-6"&gt;
28             &lt;a class="w-100" href="{{ asset('csv/resultmesure.csv') }}"&gt; &lt;button type="button"
29             " class="w-100 btn btn-sm btn-outline-secondary"&gt; Télécharger le CSV &lt;/button&gt; &lt;/
30             a&gt;
31         &lt;/div&gt;
32     &lt;/div&gt;
33
34     <!-- START THE FEATURETTES --&gt;
35
36     &lt;hr class="featurette-divider tx-black"&gt;
37
38     @foreach($stat as $stats)
39     @if($stats-&gt;campagneId == $campagne-&gt;id)
40     &lt;div class="row featurette shadow p-3 mb-5"&gt;
41     &lt;div class="col-md-6"&gt;
42         &lt;h2 class="featurette-heading tx-black"&gt;Type de la mesure @if(!empty($stats-&gt;typeV)
43             ) {{ $stats-&gt;typeV }} @else Nombre d'essieux @endif
44         @if($stats-&gt;typeV == "camion")
45             &lt;i class='fas fa-truck' width="40" height="40" fill="currentColor"&gt;&lt;/i&gt;
46         @elseif($stats-&gt;typeV == "voiture")
47             &lt;i class='fas fa-car-side' width="40" height="40" fill="currentColor"&gt;&lt;/i&gt;
48         @endif
49         &lt;br&gt;
50     @if(empty($stats-&gt;typeV)) &lt;p class="tx-black"&gt; il n'y avait qu'un seul capteur, on
51     peut donc calculer que le nombre d'essieux &lt;/p&gt; @endif
52         &lt;p class="tx-black"&gt; Date stat : @if(!empty($stats-&gt;created_at)) {{ $stats-&gt;
53             created_at }} @endif &lt;/p&gt;
54         &lt;nav class="navbar navbar-dark bg-light" aria-label="First navbar example"&gt;
55             &lt;div class="container-fluid"&gt;
56                 &lt;h2 class="navbar-brand"&gt; &lt;span class="tx-black"&gt; Détails : &lt;/span&gt; &lt;/h2&gt;
57                 &lt;button class="navbar-toggler-armand" type="button" data-bs-toggle="collapse"
58                     data-bs-target="#navbarsExample{{ $stats-&gt;id }}" aria-controls="navbarsExample{{ $-
59                         stats-&gt;id }}"
60                     aria-expanded="false" aria-label="Toggle navigation"&gt;
61                     &lt;i class="fas fa-bars tx-black"&gt;&lt;/i&gt;
62                 &lt;/button&gt;
63             &lt;/div&gt;
64         &lt;/nav&gt;
</pre>

```

ARMAND HOUPIN	stat_true.blade.php	2/4
61 62 <div class="collapse navbar-collapse" id="navbarsExample{{\\$stats->id}}"> 63 <div class="row"> 64 <hr class="featurette-divider tx-white"> 65 <hr class="featurette-divider tx-white"> 66 <hr class="featurette-divider tx-white"> 67 <hr class="featurette-divider tx-white "> 68 <div class="col-lg-4" style="width: 100%;"> 69 81 82 <h2 class=" text-center featurette-heading tx-black" style="font-size: 1em;"> 83 Limité à {{ \\$campagne->limitationvitesse }} km/h </h2> 84 85 </div> 86 </div> 87 </div class="col-md-6"> 88 <div class="collapse navbar-collapse" id="navbarsExample{{\\$stats->id}}"> 89 <hr class="featurette-divider tx-white"> 90 <div class="row" style="text-align: center;"> 91 @if(!empty(\$stats->typeV)) 92 <div class="col-lg-4" style="width: 70%;"> 93 <h2 class=" tx-black" style="font-size: 1em;"> Nombre de véhicules qui sont passés : 94 </h2> 95 </div> 96 <div class="col-lg-4" style="width: 30%;"> 97 <h2 class=" tx-black" style="font-size: 1em;"> {{ \\$stats->NbVehicule }} @if(\\$stats-> 98 typeV == "camion")<i class='fas fa-truck' width="40" height="40" fill="currentColor 99 "/></i> @else <i class='fas fa-car-side' width="40" height="40" fill="currentColor"/>< 100 i>@endif</h2> 101 </div> 102 <hr class="featurette-divider tx-white"> 103 <div class="col-lg-4" style="width: 70%;"> 104 <h2 class=" tx-black" style="font-size: 1em;"> La vitesse Moyenne des véhicules : </ 105 h2> 106 </div> 107 <div class="col-lg-4" style="width: 30%;"> 108 <h2 class=" tx-black" style="font-size: 1em;"> {{ \\$stats->VitMoyenne }} km/h </h2> 109 </div> 110 <hr class="featurette-divider tx-white"> 111 <div class="col-lg-4" style="width: 70%;"> 112 <h2 class=" tx-black" style="font-size: 1em;"> Vitesse Maximale </h2> 113 </div> 114 <div class="col-lg-4" style="width: 30%;"> 115 <h2 class=" tx-black" style="font-size: 1em;"> {{ \\$stats->VitMax }} km/h </h2> 116 </div> 117 <hr class="featurette-divider tx-white"> 118 <div class="col-lg-4" style="width: 70%;"> 119 <h2 class=" tx-black" style="font-size: 1em;"> Nombre d'essieux </h2> 120 </div> 121 <div class="col-lg-4" style="width: 30%;"> 122 <h2 class=" tx-black" style="font-size: 1em;"> {{ \\$stats->NbEssieux }} km/h </h2> 123 </div> 124 <hr class="featurette-divider tx-white"> 125 <div class="col-lg-4" style="width: 70%;"> 126 <h2 class=" tx-black" style="font-size: 1em;"> Nombre de véhicules inférieure ou 127 égale à {{\\$campagne->limitationvitesse}} km/h : </h2> 128 </div> 129 <div class="col-lg-4" style="width: 30%;"> 130 <h2 class=" tx-black" style="font-size: 1em;"> {{ \\$stats->VitesseInferieureOuEgale }} @ 131 @if(\\$stats->typeV == "camion")<i class='fas fa-truck' width="40" height="40" fill=" 132 currentColor"/></i> @else <i class='fas fa-car-side' width="40" height="40" fill=" 133 currentColor"/></i>@endif 134 </h2> 135 </div>		

ARMAND HOUPIN	stat_true.blade.php	3/4
<pre> 118 <hr class="featurette-divider tx-white"> 119 <div class="col-lg-4" style="width: 70%;"> 120 <h2 class=" tx-black" style="font-size: 1em;"> Nombre de véhicules inférieure à {{\$campagne->limitationVitesse - 20 }} km/h : </h2> 121 </div> 122 <div class="col-lg-4" style="width: 30%;"> 123 <h2 class="tx-black" style="font-size: 1em;"> {{ \$stats->VitesseLimitMoins20 }} @if(\$stats->typeV == "camion")<i class='fas fa-truck' width="40" height="40" fill="currentColor"></i> @else <i class='fas fa-car-side' width="40" height="40" fill="currentColor"></i>@endif</h2> 124 </div> 125 <hr class="featurette-divider tx-white "> 126 <div class="col-lg-4" style="width: 70%;"> 127 <h2 class=" tx-black" style="font-size: 1em;"> Nombre de véhicules supérieur à {{\$campagne->limitationVitesse + 20 }} km/h : </h2> 128 </div> 129 <div class="col-lg-4" style="width: 30%;"> 130 <h2 class="tx-black" style="font-size: 1em;"> {{ \$stats->VitesseLimitPlus20 }} @if(\$stats->typeV == "camion")<i class='fas fa-truck' width="40" height="40" fill="currentColor"></i> @else <i class='fas fa-car-side' width="40" height="40" fill="currentColor"></i>@endif</h2> 131 </div> 132 <hr class="featurette-divider tx-white"> 133 <div class="col-lg-4" style="width: 70%;"> 134 <h2 class=" tx-black" style="font-size: 1em;"> Nombre de véhicules supérieur à {{\$campagne->limitationVitesse + 30 }} km/h : </h2> 135 </div> 136 <div class="col-lg-4" style="width: 30%;"> 137 <h2 class=" tx-black" style="font-size: 1em;"> {{ \$stats->VitesseLimitPlus30 }} @if(\$stats->typeV == "camion")<i class='fas fa-truck' width="40" height="40" fill="currentColor"></i> @else <i class='fas fa-car-side' width="40" height="40" fill="currentColor"></i>@endif</h2> 138 </div> 139 <hr class="featurette-divider tx-white"> 140 <div class="col-lg-4" style="width: 70%;"> 141 <h2 class=" @if(Auth::user()->preference == 'theme_dark') tx-white @else tx-black @endif" style="font-size: 1em;"> Nombre de véhicules supérieur à {{\$campagne->limitationVitesse + 40 }} km/h : </h2> 142 </div> 143 <div class="col-lg-4" style="width: 30%;"> 144 <h2 class=" tx-black" style="font-size: 1em;"> {{ \$stats->VitesseLimitPlus40 }} @if(\$stats->typeV == "camion")<i class='fas fa-truck' width="40" height="40" fill="currentColor"></i> @else <i class='fas fa-car-side' width="40" height="40" fill="currentColor"></i>@endif</h2> 145 </div> 146 <hr class="featurette-divider tx-white"> 147 <div class="col-lg-4" style="width: 70%;"> 148 <h2 class=" tx-black" style="font-size: 1em;"> Nombre de véhicules supérieur à {{\$campagne->limitationVitesse + 50 }} km/h : </h2> 149 </div> 150 <div class="col-lg-4" style="width: 30%;"> 151 <h2 class=" tx-black" style="font-size: 1em;"> {{ \$stats->VitesseLimitPlus50 }} @if(\$stats->typeV == "camion")<i class='fas fa-truck' width="40" height="40" fill="currentColor"></i> @else <i class='fas fa-car-side' width="40" height="40" fill="currentColor"></i>@endif</h2> 152 </div> 153

 154 @else 155 <div class="col-lg-4" style="width: 70%;"> 156 <h2 class=" tx-black" style="font-size: 1em;"> Nombre d'essieux : </h2> 157 </div> 158 <div class="col-lg-4" style="width: 30%;"> 159 <h2 class=" tx-black" style="font-size: 1em;"> {{ \$stats->NbEssieux }} </h2> 160 </div> 161

 162 @endif 163 </div> 164 </div> 165 </div> 166 </div> 167 <hr class="featurette-divider tx-black"> 168 @endif 169 @endforeach </pre>		

```
171 <hr class="featurette-divider">
172
173
174 <div class="row featurette shadow p-3 mb-5">
175   <div class="col-md-13">
176     <h2 class="featurette-heading tx-black">Googles Maps</h2>
177     <br>
178     <div class="mapouter"><div class="gmap_canvas"><iframe width="100%" height="500" id
179       ="gmap_canvas" src="https://maps.google.com/maps?q={{$campagne->adresseCampagne}}{{$campagne->ville}}{{{$campagne->codePostal}}}&t=&z=9&ie=UTF8&iwloc=&output=embed" frameborder="0" scrolling="no" marginheight="0" marginwidth="0"></iframe></div></div>
180   </div>
181
182 <hr class="featurette-divider">
183
184 @include('Client/graphiques_camion')
185 @include('Client/graphiques_voiture')
186
187 @include('layouts.partials.footer')
188
189
190
191 </div>
192
193 </main>
```

ARMAND HOUPIN	creationBoitier.blade.php	1/1
	<pre> 1 @extends('layouts.home', ['title' => 'Création Boitier Mesure']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css')}}" rel="stylesheet"> 6 @stop 7 8 @section('content') 9 <div class="containerForm-Validation"> 10 <main> 11 <div class="py-5 text-center"> 12 <h2 class="mx-auto mb-1 py-5 featurette-heading"><i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-right: 10px; margin-bottom: -10px;"></i> Création Boitier <i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-right: 10px; margin-bottom: -10px;"></i> </h2> 13 </div> 14 <div class="row g-3 shadow p-3"> 15 <div class="col-md-7 col-lg-8 mx-auto"> 16 <form class="needs-validation" novalidate method="POST" action="{{ route(' 17 creationBoitier_path')}}"> 18 @csrf 19 20 <hr class="my-4 text-muted"> 21 <div class="row g-3"> 22 <div class="col-sm-12"> 23 <label for="firstName" class="form-label text-muted">Adresse @sigfox</label> 24 <input type="text" class="form-control" id="sigfox" name="sigfox" placeholder= 25 "e641221" value="" required> 26 <div class="invalid-feedback"> 27 Une adresse sigfox valide est requis. 28 </div> 29 </div> 30 31 <hr class="my-4 text-muted"> 32 33 <label for="state" class="form-label text-muted">Statut</label> 34 35 <div class="my-3"> 36 <div class="form-check"> 37 <input name="statut" type="radio" class="form-check-input" value="non 38 utilisé" required> non utilisé 39 </div> 40 </div> 41 42 <hr class="my-4 text-muted"> 43 <button class="w-100 btn btn-primary btn-lg" type="submit">Crée le boitier</button> 44 </form> 45 <hr class="my-4 text-muted"> 46 </div> 47 </div> 48 49 </main> 50 51 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or \$title != " 52 Home") 53 54 @include('layouts.partials.footer') 55 56 @endif 57 </div> 58 @stop 59 60 @section('script') 61 <script src="{{ asset('/js/form-validation.js')}}"></script> 62 @stop 63 </pre>	

```

1  @extends('layouts.home', ['title' => 'Gérer Boîtier Mesure'])
2
3
4  @section('style')
5      <link href="{{ asset('/bootstrap/form-validation.css')}}" rel="stylesheet">
6  @stop
7
8  @section('content')
9
10
11  @if(!empty($alert))
12      <div class="alert success text-center" id="alert">
13          <span class="closebtn">&times;</span>
14          <strong>Succès !</strong>
15          <hr>
16          {!! $messagealert !!}
17      </div>
18  @endif
19
20
21  <div class="containerForm-Validation">
22      <main>
23
24          <div class="py-5 text-center">
25              <h2 class="mx-auto mb-1 py-5 featurette-heading"><span class="text-muted" style="font-size: 1.5em;"><i class="fas fa-cubes" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-right: 10px; font-weight: bold;"></i> Gérer Boîtier <i class="fas fa-cubes" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-left: 10px; font-weight: bold;"></i></span></h2>
26          </div>
27
28  <div class="row g-3 shadow p-3">
29      <div class="col-md-7 col-lg-12 mx-auto">
30          <div class="row g-3">
31              <div class="col-md-8">
32                  <a href="{{ route('creationBoitier_path')}}"><svg xmlns="http://www.w3.org/2000/svg" width="30" height="35" fill="currentColor" class="bi bi-person-plus" viewBox="0 0 16 16">
33                      <path d="M6 8a3 3 0 1 0 0-6 3 0 0 0 6zm2-3a2 2 0 1 1-4 0 2 2 0 0 1 4 0zm4 8c0 1-1 1-1 1H1c-1 0-1-1 0-4 6-4 6 3 6 3 0-3.516.68-4.168 1.332-.678.678-.83 1.418-.832 1.664C9.516 10.68 8.289 10 6 10c-2.29 0-3.516.68-4.168 1.332-.678.678-.83 1.418-.832 1.664h10z"/>
34                  <path fill-rule="evenodd" d="M13.5 5a.5.5 0 0 1 .5.5V7h1.5a.5.5 0 0 1 0 1H14v1.5a.5.5 0 0 1 0V8h-1.5a.5.5 0 0 1 0-1H13V5.5a.5.5 0 0 1 0 1 .5-.5z"/>
35                  </svg></a>
36              </div>
37          </div>
38      </div>
39  </div>
40  <hr>
41  <div class="row g-3 shadow p-3">
42      <div class="col-md-7 col-lg-12 mx-auto">
43          <div class="row g-3">
44
45              <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eighth navbar example">
46                  <div class="container">
47
48                      <div class="col-md-3">
49                          <label for="firstName" class="form-label text-muted">Adresse @Sigfox</label>
50                          <br>
51                      </div>
52
53                      <div class="col-md-3">
54                          <label for="firstName" class="form-label text-muted">Niveau Batterie</label>
55                          <br>
56                      </div>
57
58                      <button class="navbar-toggler tx-black" type="button" data-bs-toggle="collapse" data-bs-target="#navbarsExample00" aria-controls="navbarsExample00" aria-expanded="false" aria-label="Toggle navigation">
59                          <i class="fas fa-bars tx-black"></i>
60                      </button>
61                      <div class="collapse navbar-collapse" id="navbarsExample00">
62                          <div class="col-md-10">
63                              <label for="firstName" class="form-label text-muted">Statut</label><br>
64                          </div>
65                          <div class="col-md-1">
66                              <label for="firstName" class="form-label text-muted">Edit</label>
67                          </div>
68                          <div class="col-md-1">
69                              <label for="firstName" class="form-label text-muted">Delete</label>
70                          </div>
71                      </div>
72                  </div>
73                  <hr class="my-4 tx-black" />

```

ARMAND HOUPIN	ListeBoiter.blade.php	2/3
<pre> 75 76 @foreach (\$boitier as \$boitier) 77 <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eight navbar example" 78 > 79 <div class="container"> 80 81 <div class="col-md-3"> 82 <label for="firstName" class="form-label tx-black">{!! \$boitier->adrSigfox 83 !!}</label> 84 </div> 85 86 <div class="col-md-3"> 87 <label for="firstName" class="form-label tx-black"> 88 @if(!empty(\$boitier->alarmeBatterie)) 89 Batterie Faible 90 \$boitier-> 91 id])}}> 92 <svg xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill= 93 currentColor" class="bi bi-battery-full" viewBox="0 0 16 16"> 94 <path d="M2 6h10v4H2V6z"/> 95 <path d="M2 4a2 2 0 0 0-2 2v4a2 2 0 0 0 2 2h10a2 2 0 0 0 2-2V6a2 2 0 0 96 0-2-2H2zm10 1a1 1 0 0 1 1 1v4a1 1 0 0 1-1 1H2a1 1 0 0 1-1-1V6a1 1 0 0 1 97 1-1h10zm4 3a1.5 1.5 0 0 1-1.5 1.5v-3A1.5 1.5 0 0 1 16 8z"/> 98 </svg> 99 @else 100 Batterie Correct 101 @endif 102 </label> 103 </div> 104 105 <button class="navbar-toggler tx-black" type="button" data-bs-toggle="collapse" 106 data-bs-target="#navbarsExample{!! \$boitier->id !!}" aria-controls="navbarsExample 107="{!! \$boitier->id !!}" aria-expanded="false" aria-label="Toggle navigation"> 108 <i class="fas fa-bars tx-black"></i> 109 </button> 110 111 <div class="collapse navbar-collapse" id="navbarsExample{!! \$boitier->id !!}"> 112 <div class="col-md-10"> 113 <label for="firstName" class="form-label tx-black">{!! \$boitier->statut !!}</ 114 label> 115 </div> 116 <div class="col-md-1"> 117 \$boitier->id])}}><svg 118 xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill= 119 currentColor" class="bi bi-pencil-square" viewBox="0 0 16 16"> 120 <path d="M15.502 1.94a.5.5 0 0 1 .706l14.459 3.691-2-2L13.502.646a.5.5 121 0 0 1 .707 0l1.293 1.293zm-1.75 2.456l-2-2L4.939 9.21a.5.5 0 0 122 0-.121.1961-.805 2.414a.25.25 0 0 0 .316.31612.414-.805a.5.5 0 0 0 123 .196-.1216.813-6.814z"/> 124 <path fill-rule="evenodd" d="M1 13.5a1.5 1.5 0 0 0 2.5 15h11a1.5 1.5 0 0 125 0 1.5-1.5v-6a.5.5 0 0 0-1 0v6a.5.5 0 0 1-5.5h-11a.5.5 0 0 126 1-5.5v-11a.5.5 0 0 1 .5-.5H9a.5.5 0 0 0 0-1H2.5a1.5 1.5 0 0 0 1 2.5v11z 127 "/> 128 </svg> 129 130 </div> 131 <div class="col-md-1"> 132 \$boitier->id])}}> <svg xmlns="http://www.w3.org/2000/svg" width 134 = "25" height="25" fill="currentColor" class="bi bi-trash" viewBox="0 0 16 135 16"> 136 <path d="M5.5 5.5a.5.5 0 0 1 1 1 6v6a.5.5 0 0 1 1-1 0V6a.5.5 0 0 1 137 .5-.5zm2.5 0 0 1 .5v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5zm3 .5a.5.5 0 0 0 138 1 0v6a.5.5 0 0 1 1-1 0V6a.5.5 0 0 1 .5-.5zm2 1 0 0 1 0V6z"/> 139 <path fill-rule="evenodd" d="M14.5 3a1 1 0 0 1-1 1H13v9a2 2 0 0 1-2 2H5a2 140 2 0 0 1-2-2V4h-.5a1 1 0 0 1-1-1V2a1 1 0 0 1 1-1H6a1 1 0 0 1 1-1h2a1 1 0 0 141 1 1h3.5a1 1 0 0 1 1v1zM4.118 4L4 4.059V13a1 1 0 0 0 1 1h6a1 1 0 0 0 142 1-1V4.059L11.882 4H4.118zM2.5 3V2h11v1h-11z"/> 143 </svg> 144 </div> 145 </div> 146 </div> 147 </div> 148 </main> 149 150 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or \$title != " 151 Home") </pre>		

ARMAND HOUPIN	ListeBoiter.blade.php	3/3
<pre>132 @include('layouts.partials.footer') 133 134 @endif 135 </div> 136 @stop 137 138 @section('script') 139 <script> 140 141 function confirmdelete() { 142 if(!confirm("Êtes-vous sûr de vouloir le supprimer ?")) 143 event.preventDefault(); 144 } 145 146 var close = document.getElementsByClassName("closebtn"); 147 var i; 148 149 for (i = 0; i < close.length; i++) { 150 close[i].onclick = function(){ 151 var div = this.parentElement; 152 div.style.opacity = "0"; 153 setTimeout(function(){ div.style.display = "none"; }, 600); 154 } 155 } 156 157 var myVar = setInterval(myTimer, 2500); 158 159 function myTimer() { 160 var div = document.getElementById("alert"); 161 div.style.opacity = "0"; 162 setTimeout(function(){ div.style.display = "none"; }, 600); 163 } 164 </script> 165 166 <script src="{{ asset('/js/form-validation.js')}}"></script> 167 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script> 168 @stop</pre>		

ARMAND HOUPIN	ModifierBoitier.blade.php	1/2
<pre> 1 @extends('layouts.home', ['title' => 'Modifier Boitier Mesure']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css')}}" rel="stylesheet"> 6 @stop 7 8 @section('content') 9 <div class="containerForm-Validation"> 10 <main> 11 <div class="py-5 text-center"> 12 <h2 class="mx-auto mb-1 py-5 featurette-heading"><i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; margin-right: 10px; vertical-align: middle;"/> Modifier Boitier <i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle;"/></h2> 13 </div> 14 <div class="row g-3 shadow p-3"> 15 <div class="col-md-7 col-lg-8 mx-auto"> 16 <form class="needs-validation" novalidate method="POST" action="{{ route(' 17 modifierBoitier_path' , ['id' => \$boitier->id])}}> 18 @csrf 19 20 <hr class="my-4 text-muted"> 21 <div class="row g-3"> 22 <div class="col-sm-12"> 23 <label for="firstName" class="form-label text-muted">Adresse @sigfox</label> 24 <input type="text" class="form-control" id="sigfox" name="sigfox" placeholder= 25 "e641221" value="{{ \$boitier->adrSigfox }}> 26 <div class="invalid-feedback"> 27 Un prénom nom valide est requis. 28 </div> 29 </div> 30 <hr class="my-4 text-muted"> 31 32 <label for="state" class="form-label text-muted">Statut</label> 33 34 @if(\$boitier->statut == "utilisé") 35 36 <label for="firstName" class="tx-black">Le boitier est associé à une campagne</ 37 label> 38 @if(!empty(\$campagne)) 39 <label for="firstName" class="tx-black">Campagne associer : {{ \$campagne->id }} {{ 40 \$campagne->adresseCampagne }} \$campagne->id])}}><svg xmlns="http://www.w3.org/2000/svg" width="25" 42 height="25" fill="currentColor" class="bi bi-pencil-square" viewBox="0 0 16 43 16"> 44 <path d="M15.502 1.94a.5.5 0 0 1 0 .706L14.459 3.69l-2-2L13.502.646a.5.5 45 0 0 1 .707 0L1.293 1.293zm-1.75 2.456l-2-2L4.939 9.21a.5.5 0 0 46 0 -.121.196l-.805 2.414a.25.25 0 0 0 .316.316l2.414-.805a.5.5 0 0 0 47 .196-.1216.813-6.814z"/> 48 <path fill-rule="evenodd" d="M1 13.5A1.5 1.5 0 0 0 2.5 15h11a1.5 1.5 0 0 49 0 1.5-1.5v-6a.5.5 0 0 0-1.5v6a.5.5 0 0 1-.5.5h-11a.5.5 0 0 50 0 1.5-.5v-11a.5.5 0 0 1 .5-.5H9a.5.5 0 0 0 0-1H2.5A1.5 1.5 0 0 0 1 2.5v11z" 51 /> 52 </svg> 53 </label> 54 @else 55 <label for="firstName" class="tx-black">Error : Impossible de trouver la 56 campagne</label> 57 <div class="my-3"> 58 <div class="form-check"> 59 <input name="statut" type="radio" class="form-check-input" value="non 60 utilisé" @if(\$boitier->statut == "non utilisé") checked @endif > non utilisé 62 </div> 63 </div> 64 @endif 65 @else 66 <div class="my-3"> 67 <div class="form-check"> 68 <input name="statut" type="radio" class="form-check-input" value="non 69 utilisé" @if(\$boitier->statut == "non utilisé") checked @endif > non utilisé 71 </div> 72 </div> 73 <div class="my-3"> 74 Associer à une campagne de mesure 75 </div> 76 @endif 77 </div> 78 </form> 79 </div> 80 </div> 81 </pre>		

```
62         <hr class="my-4 text-muted">
63         <button class="w-100 btn btn-primary btn-lg" type="submit">Modifier le boitier</
64             button>
65         </form>
66         <hr class="my-4 text-muted">
67     </div>
68 </div>
69
70 </main>
71
72 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or $title != "Home" )
73
74 @include('layouts.partials.footer')
75
76 @endif
77 </div>
78 @stop
80
81 @section('script')
82     <script src="{{ asset('/js/form-validation.js')}}"></script>
83 @stop
```

ARMAND HOUPIN	créationCampagneMesure.blade.php	1/3
	<pre> 1 @extends('layouts.home', ['title' => 'Création Campagne Mesure']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css') }}" rel="stylesheet"> 6 @stop 7 8 @section('content') 9 <div class="containerForm-Validation"> 10 <main> 11 <div class="py-5 text-center"> 12 <h2 class="mx-auto mb-1 py-5 featurette-heading"><i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-right: 10px;"/> Crédit à la campagne Mesure <i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-right: 10px;"/></h2> 13 14 </div> 15 <div class="row g-3 shadow p-3"> 16 <div class="col-md-7 col-lg-8 mx-auto"> 17 <form class="needs-validation" novalidate method="POST" action="{{ route('creationCampagneMesure_path')}}"> 18 @csrf 19 20 21 <!--'latitudecampagne', 22 'longitudeCampagne',--> 23 <hr class="my-4 text-muted"> 24 <div class="row g-3"> 25 <div class="col-sm-12"> 26 <label for="firstName" class="form-label text-muted">Adresse de la campagne</label> 27 <input type="text" class="form-control" id="prenom" name="adresseCampagne" placeholder="Lycée Alfred Klaster" value="" required> 28 <div class="invalid-feedback"> 29 Une adresse valide est requise. 30 </div> 31 </div> 32 33 <div class="col-12"> 34 <label for="email" class="form-label text-muted">Direction (information) </label> 35 <input type="text" class="form-control" id="email" name="Direction" placeholder="Vers le Nord" required> 36 <div class="invalid-feedback"> 37 Veuillez saisir une direction valide. 38 </div> 39 </div> 40 41 <div class="col-md-4"> 42 <label for="country" class="form-label text-muted">Nº de la route</label> 43 <input type="text" class="form-control" id="region" name="numeroRoute" placeholder="Ex: D15" required> 44 <div class="invalid-feedback"> 45 Veuillez saisir un Nº de route valide. 46 </div> 47 </div> 48 49 <div class="col-md-4"> 50 <label for="state" class="form-label text-muted">Ville</label> 51 <input type="text" class="form-control" id="ville" name="ville" placeholder="Ex: Talence" required> 52 <div class="invalid-feedback"> 53 Veuillez saisir une ville valide. 54 </div> 55 </div> 56 57 <div class="col-md-4"> 58 <label for="zip" class="form-label text-muted">Code Postal</label> 59 <input type="text" class="form-control" id="CP" name="codePostal" placeholder="Ex: 33522" required> 60 <div class="invalid-feedback"> 61 Veuillez saisir un Nº de code postal valide. 62 </div> 63 </div> 64 </div> 65 </form> </pre>	

ARMAND HOUPIN	créationCampagneMesure.blade.php	2/3
<pre> 66 67 68 <hr class="my-4 text-muted"> 69 70▼ <div class="col-12"> 71 <label for="email" class="form-label text-muted">Date Début</label> 72 <input type="date" class="form-control" id="telephone" name="DebutCampagne" 73 placeholder="" required> 74 <div class="invalid-feedback"> 75 Veuillez saisir une date valide. 76 </div> 77 </div> 78▼
 79 <div class="col-12"> 80 <label class="form-label text-muted">Date Fin (laisser vide si en cours)</label> 81 <input type="date" class="form-control" id="address" name="FinCampagne" 82 placeholder=""> 83 </div> 84 85 <hr class="my-4 text-muted"> 86 87 <label for="state" class="form-label text-muted">Statut</label> 88 89 <div class="my-3"> 90 <div class="form-check"> 91 <input name="statut" type="radio" class="form-check-input" value="en 92 cours" required> En cours 93 </div> 94 <div class="form-check"> 95 <input name="statut" type="radio" class="form-check-input" value="fini" 96 required> Fini 97 </div> 98 99 <hr class="my-4 text-muted"> 100 101 <label for="state" class="form-label text-muted">Limitation Vitesse</label> 102 103 <div class="my-3"> 104 <div class="form-check"> 105 <input name="limitationvitesse" type="radio" class="form-check-input" value= 106 "30" required> 30 107 </div> 108 <div class="form-check"> 109 <input name="limitationvitesse" type="radio" class="form-check-input" value= 110 "50" required> 50 111 </div> 112 <div class="form-check"> 113 <input name="limitationvitesse" type="radio" class="form-check-input" value= 114 "70" required> 70 115 </div> 116 <div class="form-check"> 117 <input name="limitationvitesse" type="radio" class="form-check-input" value= 118 "80" required> 80 119 </div> 120 <div class="form-check"> 121 <input name="limitationvitesse" type="radio" class="form-check-input" value= 122 "90" required> 90 123 </div> 124 125 <hr class="my-4 text-muted"></pre>		

```
127 ▼      <div class="col-12">
128 ▼          <label for="address" class="form-label text-muted">Associer client ( rien
129          choisir si client introuvable )</label>
130          <select name="id_user" id="user" class="form-control">
131              <option value="0">Défault</option>
132              @foreach ($client as $client)
133                  @if($client->role->name == "user")
134                      <option value="{{ $client->id }}> {{ $client->name }} {{ $client->prenom
135                      }}</option>
136                  @endif
137                  @endforeach
138          </select>
139      </div>
140
141 ▼      <hr class="my-4 text-muted">
142 ▼          <div class="col-12">
143              <label for="address" class="form-label text-muted">Associer Boitier ( rien
144              choisir si boitier introuvable )</label>
145              <select name="id_boitier" id="boitier" class="form-control">
146                  <option value="">Défault</option>
147                  @foreach ($boitier as $boitier)
148                      @if($boitier->statut == "non utilisé")
149                          <option value="{{ $boitier->id }}> {{ $boitier->adrSigfox }}</option>
150                      @endif
151                  @endforeach
152          </select>
153      </div>
154
155      <hr class="my-4 text-muted">
156      <button class="w-100 btn btn-primary btn-lg" type="submit">Crée la campagne de
157      mesure</button>
158  </form>
159  <hr class="my-4 text-muted">
160</div>
161</main>
162
163 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or $title != "
164 Home" )
165 @include('layouts.partials.footer')
166
167 @endif
168 </div>
169 @stop
170
171 @section('script')
172     <script src="{{ asset('/js/form-validation.js')}}"></script>
173 @stop
174
```

```
1  @extends('layouts.home', ['title' => 'Gérer Campagne Mesure'])
2
3
4  @section('style')
5      <link href="{{ asset('/bootstrap/form-validation.css') }}" rel="stylesheet">
6  @stop
7
8  @section('content')
9
10
11 ▼ @if(!empty($alert))
12 ▼     <div class='alert success text-center' id="alert">
13         <span class="closebtn">&times;</span>
14         <strong>Succès !</strong>
15         <hr>
16         {!! $messagealert !!}
17     </div>
18 @endif
19
20
21 ▼ <div class="containerForm-Validation" >
22 ▼     <main>
23         <div class="py-5 text-center">
24             <h2 class="mx-auto mb-1 py-5 featurette-heading"><span class="text-muted" ><i class="fas fa-route" width="75" height="75" fill="currentColor" ></i> Gérer Campagne
25             Mesure <i class="fas fa-route" width="75" height="75" fill="currentColor" ></i> </span></h2>
26
27     </div>
28
29 ▼     <div class="row g-3 shadow p-3">
30 ▼         <div class="col-md-7 col-lg-12 mx-auto ">
31 ▼             <div class="row g-3">
32                 <div class="col-md-8">
33                     <a href="{{ route('creationCampagneMesure_path') }}><svg xmlns="http://www.w3.org/2000/svg" width="30" height="35" fill="currentColor" class="bi bi-person-plus" viewBox="0 0 16 16">
34                         <path d="M6 8a3 3 0 1 0 -6 3 0 0 0 6z" />
35                         <path d="M13.5 5a.5.5 0 0 1 .5-.5V7h1.5a.5.5 0 0 1 .5-.5" />
36                         <path fill-rule="evenodd" d="M14v1.5a.5.5 0 0 1 0V8h-1.5a.5.5 0 0 1 0-1H13V5.5a.5.5 0 0 1 0-.5z" />
37                     </a>
38                 </div>
39             </div>
40             <hr>
41         <div class="row g-3 shadow p-3">
42             <div class="col-md-7 col-lg-12 mx-auto ">
43                 <div class="row g-3">
44                     <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eighth navbar example"
45                         >
46                         <div class="container">
47                             <div class="col-md-3">
48                                 <label for="firstName" class="form-label text-muted">Adresse</label><br>
49                             </div>
50
51                             <div class="col-md-2">
52                                 <label for="firstName" class="form-label text-muted">Ville</label><br>
53                             </div>
54
55                             <div class="col-md-1">
56                                 <label for="firstName" class="form-label text-muted">Route</label><br>
57                             </div>
58
59                             <button class="navbar-toggler tx-black" type="button" data-bs-toggle="collapse"
60                                     data-bs-target="#navbarsExample00" aria-controls="navbarsExample00" aria-expanded="false"
61                                     aria-label="Toggle navigation">
62                                 <i class="fas fa-bars tx-black"></i>
63                             </button>
64                         </div>
65                     </div>
66                 </div>
67             </div>
68         </div>
69     </div>
70 
```

ARMAND HOUPIN	listeCampagneMesure.blade.php	2/3
	<pre> 63 <div class="collapse navbar-collapse" id="navbarsExample00"> 64 <div class="col-md-6"> 65 <label for="firstName" class="form-label text-muted">Date début</label>
 66 </div> 67 <div class="col-md-4"> 68 <label for="firstName" class="form-label text-muted">Statut</label> 69 </div> 70 <div class="col-md-1"> 71 <label for="firstName" class="form-label text-muted">Edit</label> 72 </div> 73 <div class="col-md-1"> 74 <label for="firstName" class="form-label text-muted">Delete</label> 75 </div> 76 </div> 77 </div> 78 </div> 79 <hr class="my-4 tx-black"> 80 81 82 @foreach (\$campagne as \$campagne) 83 <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eighth navbar example"> 84 <div class="container"> 85 86 <div class="col-md-3"> 87 <label for="firstName" class="form-label tx-black ">{!! \$campagne-> 88 adresseCampagne !!}</label> 89 </div> 90 91 <div class="col-md-2"> 92 <label for="firstName" class="form-label tx-black">{!! \$campagne->ville !!}</ 93 label> 94 </div> 95 96 <div class="col-md-1"> 97 <label for="firstName" class="form-label tx-black">{!! \$campagne->numeroRoute 98 !!}</label> 99 </div> 100 101 <button class="navbar-toggler tx-black" type="button" data-bs-toggle="collapse" 102 data-bs-target="#navbarsExample{!! \$campagne->id !!}" aria-controls="navbarsExample 103 >{!! \$campagne->id !!}" aria-expanded="false" aria-label="Toggle navigation"> 104 <i class="fas fa-bars tx-black"></i> 105 </button> 106 107 <div class="collapse navbar-collapse" id="navbarsExample{!! \$campagne->id !!}"> 108 <div class="col-md-6"> 109 <label for="firstName" class="form-label tx-black">{!! \$campagne-> 110 DébutCampagne !!}</label> 111 </div> 112 <div class="col-md-4"> 113 <label for="firstName" class="form-label tx-black">{!! \$campagne-> 114 statut !!}</label> 115 </div> 116 <div class="col-md-1"> 117 \$campagne->id])}} 118 ><svg xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill=" 119 currentColor" class="bi bi-pencil-square" viewBox="0 0 16 16"> 120 <path d="M15.502 1.94a.55 0 0 1 0 .706L14.459 3.69l-2-2L13.502.646a.55 121 0 0 1 .707 0l1.293 1.293zm-1.75 2.456l-2-2L4.939 9.21a.55 0 0 122 0-.121.196l-.805 2.414a.25.25 0 0 0 .316.316l2.414-.805a.55 0 0 0 .196-.12l6.813-6.814z"/> 123 <path fill-rule="evenodd" d="M1 13.5A1.5 1.5 0 0 0 2.5 15h11a1.5 1.5 0 124 0 1.5-1.5v-6a.55 0 0 -1 0v6a.55 0 0 1-.55h-11a.55 0 0 125 0 1-.5-.5v-11a.55 0 0 1 -.5-.5H9a.55 0 0 0 0-1H2.5A1.5 1.5 0 0 0 1 2.5v11z 126 "/> 127 128 </div> 129 <div class="col-md-1"> 130 \$campagne->id])}}><svg xmlns=" 132 http://www.w3.org/2000/svg" width="25" height="25" fill="currentColor" 133 class="bi bi-trash" viewBox="0 0 16 16"> 134 <path d="M5.5 5.5A.55 0 0 1 6 6v6a.55 0 0 1-1 0V6a.55 0 0 1 .5-.5zm2.5 135 0a.55 0 0 1 .5.5v6a.55 0 0 1-1 0V6a.55 0 0 1 .5-.5zm3 .5a.55 0 0 0-1 136 0v6a.55 0 0 1 0V6z"/> 137 <path fill-rule="evenodd" d="M14.5 3a1.1 0 0 1-1 1H13v9a2 2 0 0 1-2 2H5a2 138 2 0 0 1-2-2V4h-.5a1 1 0 0 1-1-1V2a1 1 0 0 1 1-1H6a1 1 0 0 1 1-1h2a1 1 0 0 139 1 1 1h3.5a1 1 0 0 1 1 1v1zM4.118 4L4 4.059V13a1 1 0 0 0 1 1h6a1 1 0 0 0 140 1-1V4.059L11.882 4H4.118zM2.5 3V2h11v1h-11z"/> 141 142 </div> 143 </div> 144 </div> 145 </div> 146 </pre>	

ARMAND HOUPIN	listeCampagneMesure.blade.php	3/3
<pre> 123 </div> 124 </div> 125 <hr class="my-4 text-muted"> 126 @endforeach 127 128 <!--<hr class="my-4 text-muted">--> 129 </div> 130 </div> 131 </div> 132 </main> 133 134 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or \$title != "Home") 135 136 @include('layouts.partials.footer') 137 138 @endif 139 140 141 </div> 142 143 @stop 144 145 @section('script') 146 <script> 147 148 function confirmdelete() { 149 if(!confirm("Êtes-vous sûr de vouloir le supprimer ?")) 150 event.preventDefault(); 151 } 152 153 var close = document.getElementsByClassName("closebtn"); 154 var i; 155 156 for (i = 0; i < close.length; i++) { 157 close[i].onclick = function(){ 158 var div = this.parentElement; 159 div.style.opacity = "0"; 160 setTimeout(function(){ div.style.display = "none"; }, 600); 161 } 162 } 163 164 var myVar = setInterval(myTimer, 2500); 165 166 function myTimer() { 167 var div = document.getElementById("alert"); 168 div.style.opacity = "0"; 169 setTimeout(function(){ div.style.display = "none"; }, 600); 170 } 171 </script> 172 173 <script src="{{ asset('/js/form-validation.js')}}"></script> 174 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script> 175 176 @stop </pre>		

ARMAND HOUPIN	modifierCampagneMesure.blade.php	1/4
<pre> 1 @extends('layouts.home', ['title' => 'Modifier Campagne Mesure']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css')}}" rel="stylesheet"> 6 <link rel="stylesheet" href="https://unpkg.com/leaflet@1.3.1/dist/leaflet.css" integrity= 7 "sha512-Rksm5RenBEKSKFjg13a41vrjkw4EVPlJ3+0iI65vTjIdo9brlAacEuKOiQ5OFh7cOI1bkDwLqdLw3Zg0c 8 RJAAQ==" crossorigin="" /> 9 @stop 10 11 12 @section('content') 13 <div class="containerForm-Validation"> 14 <main> 15 <div class="py-5 text-center"> 16 <h2 class="mx-auto mb-1 py-5 featurette-heading"> <i class="fas fa-user" width="75" height="75" fill="currentColor" ></i> Modifier Campagne 17 Mesure <i class="fas fa-user" width="75" height="75" fill="currentColor"></i> 18 </h2> 19 20 21 22 <!--'latitudecampagne', 23 'longitudeCampagne',--> 24 <hr class="my-4 text-muted"> 25 <div class="row g-3 shadow p-3"> 26 <div class="col-md-7 col-lg-8 mx-auto "> 27 <form class="needs-validation" novalidate method="POST" action="{{ route(28 'modifierCampagneMesure_path' , ['id' => \$campagne->id])}}"> 29 @csrf 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 </pre>		

```

68
69             <hr class="my-4 text-muted">
70
71     <div class="col-12">
72         <label for="email" class="form-label text-muted">Date Début</label>
73         <input type="date" class="form-control" id="telephone" name="DebutCampagne"
74         placeholder="" value="{{ $campagne->DébutCampagne }}>
75         <div class="invalid-feedback">
76             Veuillez saisir un numéro de téléphone valide.
77         </div>
78     <br>
79     <div class="col-12">
80         <label class="form-label text-muted">Date Fin ( laisser vide si en cours )</label>
81         <input type="date" class="form-control" id="address" name="FinCampagne"
82         placeholder="" value="{{ $campagne->FinCampagne }}>
83     </div>
84
85     <hr class="my-4 text-muted">
86
87     <label for="state" class="form-label text-muted">Statut</label>
88
89     <div class="my-3">
90         <div class="form-check">
91             <input name="statut" type="radio" class="form-check-input" value="en
92             cours" @if($campagne->statut == "en cours") checked @endif> <span class="tx-black"> En cours </span>
93         </div>
94         <div class="form-check">
95             <input name="statut" type="radio" class="form-check-input" value="fini" @
96             if($campagne->statut == "fini") checked @endif> <span class="tx-black">
97                 Fini </span>
98         </div>
99     </div>
100
101     <hr class="my-4 text-muted">
102
103     <label for="state" class="form-label text-muted">Limitation Vitesse</label>
104
105     <div class="my-3">
106         <div class="form-check">
107             <input name="limitationvitesse" type="radio" class="form-check-input" value=
108             "30" @if($campagne->limitationvitesse == "30") checked @endif> <span class="tx-black"> 30 </span>
109         </div>
110         <div class="form-check">
111             <input name="limitationvitesse" type="radio" class="form-check-input" value=
112             "50" @if($campagne->limitationvitesse == "50") checked @endif> <span class="tx-black"> 50 </span>
113         </div>
114         <div class="form-check">
115             <input name="limitationvitesse" type="radio" class="form-check-input" value=
116             "70" @if($campagne->limitationvitesse == "70") checked @endif> <span class="tx-black"> 70 </span>
117         </div>
118         <div class="form-check">
119             <input name="limitationvitesse" type="radio" class="form-check-input" value=
120             "80" @if($campagne->limitationvitesse == "80") checked @endif> <span class="tx-black"> 80 </span>
121         </div>
122         <div class="form-check">
123             <input name="limitationvitesse" type="radio" class="form-check-input" value=
124             "90" @if($campagne->limitationvitesse == "90") checked @endif> <span class="tx-black"> 90 </span>
125         </div>
126         <div class="form-check">
127             <input name="limitationvitesse" type="radio" class="form-check-input" value=
128             "110" @if($campagne->limitationvitesse == "110") checked @endif> <span class="tx-black"> 110 </span>
129         </div>
130         <div class="form-check">
131             <input name="limitationvitesse" type="radio" class="form-check-input" value=
132             "130" @if($campagne->limitationvitesse == "130") checked @endif> <span class="tx-black"> 130 </span>
133         </div>
134     </div>

```

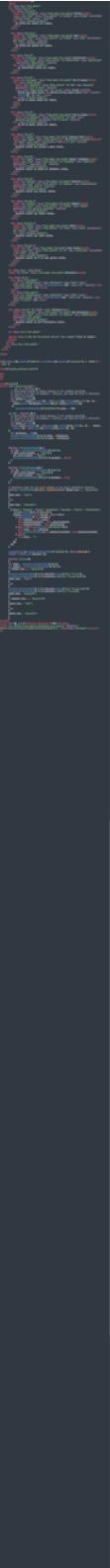


ARMAND HOUPIN	modifierCampagneMesure.blade.php	3/4
	<pre> 126 <hr class="my-4 text-muted"> 127 128 <div class="col-12"> 129 <label for="address" class="form-label text-muted">Associer client</label> 130 <select name="id_user" id="user" class="form-control"> 131 @if(!empty(\$clientdefault->id)) <option value="{{ \$clientdefault->id }}>{{ \$clientdefault->name }} {{ \$clientdefault->prenom }}</option> @endif 132 <option value=""> Aucun client </option> 133 134 @foreach (\$client as \$client) 135 136 @if(!empty(\$client)) 137 @if(\$client->role->name == "user") 138 <option value="{{ \$client->id }}> {{ \$client->name }} {{ \$client->prenom }}</option> 139 @endif 140 @else 141 @if(\$client->role->name == "user") 142 @if(\$client->name != \$clientdefault->name) 143 <option value="{{ \$client->id }}> {{ \$client->name }} {{ \$client->prenom }}</option> 144 @endif 145 @endif 146 @endif 147 @endforeach 148 </select> 149 <!--<input type="option" class="form-control" id="address" 150 name="nomEntreprise" placeholder="--> 151 <div class="invalid-feedback"> 152 Veuillez saisir un nom d'entreprise valide. 153 </div> 154 </div> 155 156 <hr class="my-4 text-muted"> 157 <div class="col-12"> 158 <label for="address" class="form-label text-muted">Associer Boitier</label> 159 <select name="id_boitier" id="boitier" class="form-control"> 160 161 @if(!empty(\$boitierdefault)) 162 <option value="{{ \$boitierdefault->id }}>{{ \$boitierdefault->adrSigfox }}</option> 163 @endif 164 165 <option value=""> Aucun boitier </option> 166 167 @foreach (\$boitier as \$boitier) 168 169 @if(!empty(\$boitierdefault)) 170 @if(\$boitier->adrSigfox != \$boitierdefault->adrSigfox) 171 @if(\$boitier->statut == "non utilisé") 172 <option value="{{ \$boitier->id }}> {{ \$boitier->adrSigfox }}</option> 173 @endif 174 @endif 175 @else 176 @if(\$boitier->statut == "non utilisé") 177 <option value="{{ \$boitier->id }}> {{ \$boitier->adrSigfox }}</option> 178 @endif 179 @endif 180 181 @endforeach 182 </select> 183 </div> 184 185 <hr class="my-4 text-muted"> 186 187 <button class="w-100 btn btn-primary btn-lg" type="submit">Modifier la campagne 188 de mesure</button> 189 190 <hr class="my-4 text-muted"></pre>	

```
191 ▼      <div class="row featurette shadow p-3 mb-5">
192 ▼          <div class="col-md-13">
193              <h2 class="featurette-heading">Google Maps</h2>
194              <br>
195              <div class="mapouter"><div class="gmap_canvas"><iframe width="100%" height="500" id="gmap_canvas" src="https://maps.google.com/maps?q={{$campagne->adresseCampagne}}{{$campagne->ville}}{{$campagne->codePostal}}&t=&z=9&ie=UTF8&iwloc=&output=embed" frameborder="0" scrolling="no" marginheight="0" marginwidth="0"></iframe></div></div>
196          </div>
197      </div>
198  </div>
199
200  </main>
201
202  @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or $title != "Home" )
203
204  @include('layouts.partials.footer')
205
206  @endif
207  </div>
208  @stop
209
210  @section('script')
211  <script src="{{ asset('/js/form-validation.js')}}"></script>
212
213  @stop
214
```

ARMAND HOUPIN	inscription.blade.php	1/3
<pre> 1 @extends('layouts.home', ['title' => 'Création Compte Client']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css') }}" rel="stylesheet"> 6 @stop 7 8 @section('content') 9 10 @if(!empty(\$alert)) 11 <div class="alert error text-center" id="alert"> 12 &times; 13 Succès ! 14 <hr> 15 {!! \$messagealert !!} 16 </div> 17 @endif 18 19 20 <div class="containerForm-Validation"> 21 <main> 22 <div class="py-5 text-center"> 23 <h2 class="mx-auto mb-1 py-5 featurette-heading"><i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-right: 10px; margin-bottom: -10px;"></i> Création Compte Client <i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-right: 10px; margin-bottom: -10px;"></i></h2> 24 25 </div> 26 <div class="row g-3 shadow p-3"> 27 <div class="col-md-7 col-lg-8 mx-auto"> 28 <form class="needs-validation" novalidate method="POST" action="{{ route('ajouterClient_path') }}> 29 @csrf 30 <hr class="my-4 text-muted"> 31 <div class="row g-3"> 32 <div class="col-sm-6"> 33 <label for="firstName" class="form-label text-muted">Prénom</label> 34 <input type="text" class="form-control" id="prenom" name="prenom" placeholder="Prénom Client" value="" required> 35 <div class="invalid-feedback"> 36 Un prénom nom valide est requis. 37 </div> 38 </div> 39 40 <div class="col-sm-6"> 41 <label for="firstName" class="form-label text-muted">Nom</label> 42 <input type="text" class="form-control" id="prenom" name="name" placeholder="Nom Client" value="" required> 43 <div class="invalid-feedback"> 44 Un prénom nom valide est requis. 45 </div> 46 </div> 47 48 <div class="col-12"> 49 <label for="username" class="form-label text-muted">Identifiant</label> 50 <input type="text" class="form-control" id="Identifiant" name="identifiant" placeholder="Identifiant" required> 51 <div class="invalid-feedback"> 52 Un identifiant valide est requis. 53 </div> 54 </div> 55 56 <div class="col-12"> 57 <label for="username" class="form-label text-muted">Mot de passe</label> 58 <div class="input-group"> 59 <input type="password" class="form-control" id="mdp" name="password" placeholder="Mot de passe" required> 60 Afficher() 61 <div class="invalid-feedback"> 62 Un mot de passe valide est requis. 63 </div> 64 </div> 65 </div> 66 </form> </pre>		

ARMAND HOUPIN	inscription.blade.php	2/3
67 ▼	<pre> 68 <div class="col-12"> 69 <label for="username" class="form-label text-muted">Mot de passe</label> 70 <input type="password" class="form-control" id="mdp2" name="password_confirmed" placeholder="Mot de passe" required> 71 <div class="invalid-feedback"> 72 Un mot de passe valide est requis. 73 </div> 74 </div> 75 ▼ 76 <div class="col-12"> 77 <label for="email" class="form-label text-muted">Adresse Mail</label> 78 <input type="email" class="form-control @error('email') is-invalid @enderror" id="email" name="email" placeholder="you@example.com" required> 79 <div class="invalid-feedback"> 80 Veuillez saisir une adresse e-mail valide. 81 </div> 82 </div> 83 ▼ 84 <div class="col-12"> 85 <label for="telephone" class="form-label text-muted">Numéro Téléphone</label> 86 <input type="text" class="form-control" id="telephone" name="numeroTel" placeholder="" maxlength="10" required> 87 <div class="invalid-feedback"> 88 Veuillez saisir un numéro de téléphone valide. 89 </div> 90 </div> 91 ▼ 92 <div class="col-12"> 93 <label for="address" class="form-label text-muted">Adresse</label> 94 <input type="text" class="form-control" id="address" name="adresseClient" placeholder="1234 Main St" required> 95 <div class="invalid-feedback"> 96 Veuillez saisir une adresse valide. 97 </div> 98 </div> 99 ▼ 100 <div class="col-md-5"> 101 <label for="country" class="form-label text-muted">Région</label> 102 <input type="text" class="form-control" id="region" name="region" placeholder="Ex: Nouvelle-Aquitaine" required> 103 <div class="invalid-feedback"> 104 Veuillez saisir une région valide. 105 </div> 106 </div> 107 ▼ 108 <div class="col-md-4"> 109 <label for="state" class="form-label text-muted">Ville</label> 110 <input type="text" class="form-control" id="ville" name="ville" placeholder="Ex: Talence" required> 111 <div class="invalid-feedback"> 112 Veuillez saisir une ville valide. 113 </div> 114 </div> 115 ▼ 116 <div class="col-md-3"> 117 <label for="zip" class="form-label text-muted">Code Postal</label> 118 <input type="text" class="form-control" id="CP" name="codePostal" placeholder="Ex: 33522" required> 119 <div class="invalid-feedback"> 120 Veuillez saisir un N° de code postal valide. 121 </div> 122 </div> 123 <hr class="my-4 text-muted"> 124 <label for="state" class="form-label text-muted">Entreprise</label> 125 <div class="my-3"> 126 <div class="form-check"> 127 <input id="ouiEntreprise" name="entreprise" type="radio" class="form-check-input" onclick="AfficherEntreprise();" value="oui" required> Oui 128 </div> 129 <div class="form-check"> 130 <input id="nonEntreprise" name="entreprise" type="radio" class="form-check-input" onclick="CacherEntreprise();" value="non" required> Non 131 </div> 132 </div> 133 </div> 134 </pre>	ARMAND HOUPIN

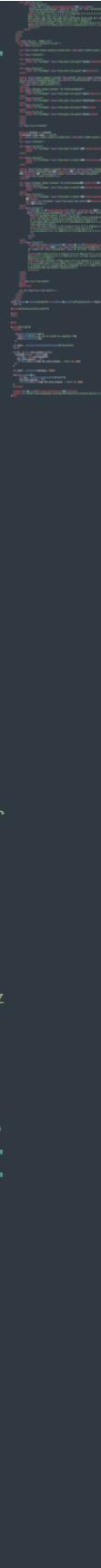
ARMAND HOUPIN	inscription.blade.php	3/3
	<pre> 136 <div class="col-12" id="maDIV" style="display:none;"> 137 <label for="address" class="form-label text-muted">Nom Entreprise</label> 138 <input type="text" class="form-control" id="address" name="nomEntreprise" 139 placeholder="Ex: google"> 140 <div class="invalid-feedback"> 141 Veuillez saisir un nom d'entreprise valide. 142 </div> 143 144 <hr class="my-4 text-muted"> 145 146 <button class="w-100 btn btn-primary btn-lg" type="submit">Créer le compte</ 147 button> 148 </form> 149 </div> 150 </div> 151 152 </main> 153 154 155 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or \$title != "Home") 156 157 @include('layouts.partials.footer') 158 159 @endif 160 </div> 161 @stop 162 163 @section('script') 164 <script type="text/javascript"> 165 var ID = function () { 166 // Math.random should be unique because of its seeding algorithm. 167 // Convert it to base 36 (numbers + letters), and grab the first 9 characters 168 // after the decimal. 169 var ladata=new Date() 170 return 'c' + ladata.getHours() + Math.random().toString(36).substr(2, 9) + 171 ladata.getDate()+(ladata.getMonth()+1)+ ladata.getMinutes(); 172 }; 173 document.getElementById("Identifiant").value = ID(); 174 175 var MDP = function () { 176 // Math.random should be unique because of its seeding algorithm. 177 // Convert it to base 36 (numbers + letters), and grab the first 9 characters 178 // after the decimal. 179 var ladata=new Date() 180 return ladata.getHours() + Math.random().toString(36).substr(2, 9) + ladata. 181 getHours() + Math.random().toString(36).substr(2, 9); 182 }; 183 var motdepasse = MDP(); 184 document.getElementById("mdp").value = motdepasse; 185 document.getElementById("mdp2").value = motdepasse; 186 187 function AfficherEntreprise() { 188 var div = document.getElementById("maDIV"); 189 if (div.style.display === "none") { 190 div.style.display = "block"; 191 document.getElementById("maDIV").disabled = false; 192 } 193 }; </pre>	

```

1  @extends('layouts.home', ['title' => 'Gérer Compte Client'])
2
3
4  @section('style')
5      <link href="{{ asset('/bootstrap/form-validation.css')}}" rel="stylesheet">
6  @stop
7
8  @section('content')
9
10 @if(!empty($alert))
11     <div class="alert success text-center" id="alert">
12         <span class="closebtn">&times;</span>
13         <strong>Succès !</strong>
14         <hr>
15         {!! $messagealert !!}
16     </div>
17 @endif
18 @endif
19
20 <div class="containerForm-Validation">
21     <main>
22         <div class="py-5 text-center">
23             <h2 class="mx-auto mb-1 py-5 featurette-heading"><span class="text-muted" style="font-size: 1.5em;"><i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-right: 10px; margin-bottom: -10px;"></i> Gérer Compte Client</span></h2>
24
25     </div>
26
27     <div class="row g-3 shadow p-3">
28         <div class="col-md-7 col-lg-12 mx-auto">
29             <div class="row g-3">
30                 <div class="col-md-8">
31                     <a href="{{ route('inscriptionCompteClient')}}"><svg xmlns="http://www.w3.org/2000/svg" width="30" height="35" fill="currentColor" class="bi bi-person-plus" viewBox="0 0 16 16">
32                         <path d="M6 8a3 3 0 1 0 0-6 3 0 0 0 0 6zm2-3a2 2 0 1 1-4 0 2 2 0 0 1 4 0zm4 8c0 1-1 1-1 1H1s-1 0-1-1 1-4 6-4 6 3 6 4zm-1-.004c-.001-.246-.154-.986-.832-1.664C9.516 10.68 8.289 10 6 10c-2.29 0-3.516 6.4-1.68 1.332-.678.678-.83 1.418-.832 1.664h10z"/>
33                         <path fill-rule="evenodd" d="M13.5 5a.5.5 0 0 1 .5.5V7h1.5a.5.5 0 0 1 .5.5V8h-1.5a.5.5 0 0 1 0-1H13V5.5a.5.5 0 0 1 .5-.5z"/>
34                     </svg></a>
35                 </div>
36             </div>
37         </div>
38     </div>
39     <hr>
40     <div class="row g-3 shadow p-3">
41         <div class="col-md-7 col-lg-12 mx-auto">
42             <div class="row g-3">
43                 <div class="col-md-2">
44                     <label for="firstName" class="form-label text-muted">Prénom</label><br>
45                 </div>
46
47                 <div class="col-md-2">
48                     <label for="firstName" class="form-label text-muted">Nom</label><br>
49                 </div>
50
51                 <button class="navbar-toggler tx-black" type="button" data-bs-toggle="collapse" data-bs-target="#navbarsExample00" aria-controls="navbarsExample00" aria-expanded="false" aria-label="Toggle navigation">
52                     <i class="fas fa-bars tx-black"></i>
53                 </button>
54                 <div class="collapse navbar-collapse" id="navbarsExample00">
55                     <div class="col-md-6">
56                         <label for="firstName" class="form-label text-muted">Email</label><br>
57                     </div>
58                     <div class="col-md-4">
59                         <label for="firstName" class="form-label text-muted">Identifiant</label>
60                     </div>
61                     <div class="col-md-1">
62                         <label for="firstName" class="form-label text-muted">Edit</label>
63                     </div>
64                     <div class="col-md-1">
65                         <label for="firstName" class="form-label text-muted">Delete</label>
66                     </div>
67                 </div>
68             <hr class="my-4 tx-black">
69         </div>
70     </div>
71     </div>
72     </div>
73     </div>
74     <hr class="my-4 tx-black">
75

```



ARMAND HOUPIN	ListeClient.blade.php	2/3
	<pre> 77 @foreach (\$client as \$client) 78 @if(\$client->role->name == "user") 79 <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eighth navbar example"> 80 <div class="container"> 81 <div class="col-md-2"> 82 <label for="firstName" class="form-label tx-black">{!! \$client->prenom !!}</ 83 label> 84 </div> 85 86 <div class="col-md-2"> 87 <label for="firstName" class="form-label tx-black">{!! \$client->name !!}</ 88 label> 89 </div> 90 91 <button class="navbar-toggler tx-black" type="button" data-bs-toggle="collapse" 92 data-bs-target="#navbarsExample{!! \$client->id !!}" aria-controls="navbarsExample 93 >{!! \$client->id !!}" aria-expanded="false" aria-label="Toggle navigation"> 94 <i class="fas fa-bars tx-black"></i> 95 </button> 96 97 <div class="collapse navbar-collapse" id="navbarsExample{!! \$client->id !!}"> 98 <div class="col-md-6"> 99 <label for="firstName" class="form-label tx-black">{!! \$client->email !!}</ 100 label> 101 </div> 102 <div class="col-md-4"> 103 <label for="firstName" class="form-label tx-black">{!! \$client->identifiant 104 !!}</label> 105 <!--<label for="firstName" class="form-label text-muted">{!! \$client->role-> 106 name !!}</label>--> 107 </div> 108 <div class="col-md-1"> 109 \$client->id])}}><svg 110 xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill="currentColor 111 " class="bi bi-pencil-square" viewBox="0 0 16 16"> 112 <path d="M15.502 1.94a.5.5 0 0 1 .706L14.459 3.69l-2-2L13.502.646a.5.5 113 0 0 1 .707 0l1.293 1.293zm-1.75 2.456l-2-2L4.939 9.21a.5.5 0 0 114 0 -.121.196l-.805 2.414a.25.25 0 0 0 .316.316l2.414-.805a.5.5 0 0 0 115 .196-.121.813-6.814z"/> 116 <path fill-rule="evenodd" d="M1 13.5A1.5 1.5 0 0 0 2.5 15h11a1.5 1.5 0 0 117 0 1.5-1.5v-6a.5.5 0 0 0-1 0v6a.5.5 0 0 1-1.5h-11a.5.5 0 0 118 0 1-1.5v-11a.5.5 0 0 1 .5-.5H9a.5.5 0 0 0 0-1H2.5A1.5 1.5 0 0 0 1 2.5v11z 119 "/> 120 121 </div> 122 <div class="col-md-1"> 123 \$client->id])}}><svg xmlns="http://www.w3.org/2000/svg" width= 125 25 height="25" fill="currentColor" class="bi bi-trash" viewBox="0 0 16 16"> 126 <path d="M5.5 5.5A.5.5 0 0 1 6 6v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5zm2.5 127 0a.5.5 0 0 1 .5.5v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5zm3 .5a.5.5 0 0 0-1 128 0v6a.5.5 0 0 1 0V6z"/> 129 <path fill-rule="evenodd" d="M14.5 3a1 1 0 0 1-1 1H13v9a2 2 0 0 1-2 2H5a2 130 2 0 0 1-2-2V4h-.5a1 1 0 0 1-1 1V2a1 1 0 0 1 1-1H6a1 1 0 0 1 1-h2a1 1 0 0 131 1 1h3.5a1 1 0 0 1 1 1v1zM4.118 4L4 4.059V13a1 1 0 0 0 1 1h6a1 1 0 0 0 132 1-1V4.059L11.882 4H4.118zM2.5 3V2h11v1h-11z"/> 133 134 </div> 135 </div> 136 </div> 137 </div> 138 <hr class="my-4 text-muted"> 139 @endif 140 @endforeach 141 </pre>	

ARMAND HOUPIN	ListeClient.blade.php	3/3
<pre> 123 </div> 124 </div> 125 </div> 126 </main> 127 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or \$title != "Home") 128 @include('layouts.partials.footer') 129 @endif 130 </div> 131 132 @stop 133 134 135 @stop 136 137 @section('script') 138 <script> 139 140 function confirmdelete() { 141 if(!confirm("Êtes-vous sûr de vouloir le supprimer ?")) 142 event.preventDefault(); 143 } 144 145 var close = document.getElementsByClassName("closebtn"); 146 var i; 147 148 for (i = 0; i < close.length; i++) { 149 close[i].onclick = function(){ 150 var div = this.parentElement; 151 div.style.opacity = "0"; 152 setTimeout(function(){ div.style.display = "none"; }, 600); 153 } 154 } 155 156 var myVar = setInterval(myTimer, 2500); 157 158 function myTimer() { 159 var div = document.getElementById("alert"); 160 div.style.opacity = "0"; 161 setTimeout(function(){ div.style.display = "none"; }, 600); 162 } 163 </script> 164 165 <script src="{{ asset('/js/form-validation.js')}}"></script> 166 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script> 167 @stop </pre>		

ARMAND HOUPIN	ModifierClient.blade.php	1/2
1 @extends('layouts.home', ['title' => 'Modifier Compte Client']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css')}}" rel="stylesheet"> 6 @stop 7 8 9 @section('content') 10 ▼ <div class="containerForm-Validation"> 11 ▼ <main> 12 <div class="py-5 text-center"> 13 <h2 class="mx-auto mb-1 py-5 featurette-heading"> <i class="fas fa-user" width="75" height="75" fill="currentColor" >/</i> Information Compte {{ \$client->role->name }} <i class="fas fa-user" width="75" height="75" fill="currentColor" >/</i> </h2> 14 </div> 15 <div class="row g-3 shadow p-3"> 16 <div class="col-md-7 col-lg-8 mx-auto "> 17 <form class="needs-validation" novalidate method="POST" action="{{ route('modifierClient_path', ['id' => \$client->id]) }}"> 18 @csrf 19 <hr class="my-4 text-muted"> 20 <div class="row g-3"> 21 <div class="col-sm-6"> 22 <label for="firstName" class="form-label text-muted">Prénom</label> 23 <input type="text" class="form-control" id="prenom" name="prenom" placeholder="Prénom Client" value="{{ \$client->prenom }}" required> 24 <div class="invalid-feedback"> 25 Un prénom valide est requis. 26 </div> 27 </div> 28 </div> 29 30 <div class="col-sm-6"> 31 <label for="firstName" class="form-label text-muted">Nom</label> 32 <input type="text" class="form-control" id="prenom" name="name" placeholder="Prénom Client" value="{{ \$client->name }}" required> 33 <div class="invalid-feedback"> 34 Un prénom valide est requis. 35 </div> 36 </div> 37 38 <div class="col-12"> 39 <label for="email" class="form-label text-muted">Adresse Mail</label> 40 <input type="email" class="form-control" id="email" name="email" value="{{ \$client->email }}" required> 41 <div class="invalid-feedback"> 42 Veuillez saisir une adresse e-mail valide. 43 </div> 44 </div> 45 46 <div class="col-12"> 47 <label for="email" class="form-label text-muted">Numéro Téléphone</label> 48 <input type="text" class="form-control" id="telephone" placeholder="" value="{{ \$client->numeroTel }}" required> 49 <div class="invalid-feedback"> 50 Veuillez saisir un numéro de téléphone valide. 51 </div> 52 </div> 53 54 <div class="col-12"> 55 <label for="address" class="form-label text-muted">Adresse</label> 56 <input type="text" class="form-control" id="address" name="adresseClient" value="{{ \$client->adresseClient }}" required> 57 <div class="invalid-feedback"> 58 Veuillez saisir une adresse valide. 59 </div> 60 </div> 61 62 <div class="col-md-5"> 63 <label for="country" class="form-label text-muted">Région</label> 64 <input type="text" class="form-control" id="region" name="region" value="{{ \$client->region }}" required> 65 <div class="invalid-feedback"> 66 Veuillez saisir une région valide. 67 </div> 68 </div> 69		

```

70      <div class="col-md-4">
71          <label for="state" class="form-label text-muted">Ville</label>
72          <input type="text" class="form-control" id="ville" name="ville" value="{{ $
73              client->ville }}" required>
74          <div class="invalid-feedback">
75              Veuillez saisir une ville valide.
76          </div>
77      </div>
78
79      <div class="col-md-3">
80          <label for="zip" class="form-label text-muted">Code Postal</label>
81          <input type="text" class="form-control" id="CP" name="codePostal" value="{{ $
82              client->codePostal }}" required>
83          <div class="invalid-feedback">
84              Veuillez saisir un N° de code postal valide.
85          </div>
86      </div>
87
88      <hr class="my-4 text-muted">
89
90      <div class="col-12">
91          <label for="address" class="form-label text-muted">Nom Entreprise</label>
92          <input type="text" class="form-control" id="address" name="nomEntreprise"
93          value="{{ $client->nomEntreprise }}">
94          <div class="invalid-feedback">
95              Veuillez saisir un nom d'entreprise valide.
96          </div>
97          <hr class="my-4 text-muted">
98
99          <button class="w-100 btn btn-primary btn-lg" type="submit">Modifier le compte</
100         button>
101     </form>
102     <hr class="my-4 text-muted">
103   </div>
104 </main>
105
106 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or $title != "
107     Home" )
108 @include('layouts.partials.footer')
109
110 @endif
111 </div>
112 @stop
113
114
115 @section('script')
116     <script src="{{ asset('/js/form-validation.js')}}"></script>
117 @stop
118

```

```

1  @extends('layouts.home', ['title' => 'Création Compte Admin / Technicien'])
2
3
4  @section('style')
5    <link href="{{ asset('/bootstrap/form-validation.css') }}" rel="stylesheet">
6  @stop
7
8  @section('content')
9  <div class="containerForm-Validation">
10 <main>
11   <div class="py-5 text-center">
12     <h2 class="mx-auto mb-1 py-5 featurette-heading"><span class="text-muted" style="font-size: 1.5em;"><i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-right: 10px;></i> Crédit Compte Admin / Technicien <i class="fas fa-user" width="75" height="75" fill="currentColor" style="font-size: 1.5em; vertical-align: middle; margin-left: 10px;></i></span></h2>
13
14 </div>
15 <div class="row g-3 shadow p-3">
16   <div class="col-md-7 col-lg-8 mx-auto">
17     <form class="needs-validation" novalidate method="POST" action="{{ route('ajouterAdminTechnicien_path')}}">
18       @csrf
19       <hr class="my-4 text-muted">
20       <div class="row g-3">
21         <div class="col-sm-6">
22           <label for="firstName" class="form-label text-muted">Prénom</label>
23           <input type="text" class="form-control" id="prenom" name="prenom" placeholder="Prénom Client" value="" required>
24           <div class="invalid-feedback">
25             Un prénom valide est requis.
26           </div>
27         </div>
28
29         <div class="col-sm-6">
30           <label for="firstName" class="form-label text-muted">Nom</label>
31           <input type="text" class="form-control" id="nom" name="nom" placeholder="Prénom Client" value="" required>
32           <div class="invalid-feedback">
33             Un nom valide est requis.
34           </div>
35         </div>
36
37         <div class="col-12">
38           <label for="username" class="form-label text-muted">Identifiant</label>
39           <input type="text" class="form-control" id="username" name="username" placeholder="Identifiant" value="" required>
40           <div class="invalid-feedback">
41             Un identifiant valide est requis.
42           </div>
43         </div>
44
45         <div class="col-12">
46           <label for="password" class="form-label text-muted">Mot de passe</label>
47           <input type="password" class="form-control" id="password" name="password" placeholder="Mot de passe" value="" required>
48           <div class="invalid-feedback">
49             Un mot de passe valide est requis.
50           </div>
51         </div>
52
53         <div class="col-12">
54           <label for="password_confirmation" class="form-label text-muted">Mot de passe</label>
55           <input type="password" class="form-control" id="password_confirmation" name="password_confirmation" placeholder="Mot de passe" value="" required>
56           <div class="invalid-feedback">
57             Un mot de passe valide est requis.
58           </div>
59         </div>
60
61         <div class="col-12">
62           <label for="email" class="form-label text-muted">Adresse Mail</label>
63           <input type="email" class="form-control" id="email" name="email" placeholder="you@example.com" value="" required>
64           <div class="invalid-feedback">
65             Veuillez saisir une adresse e-mail valide.
66           </div>
67         </div>
68
69       </div>
70       <div class="d-grid gap-2 d-md-block">
71         <button type="submit" class="btn btn-primary w-100" style="width: 100%;">Ajouter
72       </div>
73     </form>
74   </div>
75 </div>
76
```

ARMAND HOUPIN	inscriptionAdminTechnicien.blade.php	2/2
<pre> 69 <hr class="my-4 text-muted"> 70 <label for="state" class="form-label text-muted">Role</label> 71 72 <div class="my-3"> 73 <div class="form-check"> 74 <input name="role" type="radio" class="form-check-input" value="1" required checked> Admin 75 </div> 76 <div class="form-check"> 77 <input name="role" type="radio" class="form-check-input" value="3" required> Technicien 78 </div> 79 </div> 80 81 <hr class="my-4 text-muted"> 82 83 <button class="w-100 btn btn-primary btn-lg" type="submit">Crée le compte</button> 84 </form> 85 <hr class="my-4 text-muted"> 86 </div> 87 </div> 88 </div> 89 90 </main> 91 92 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or \$title != "Home") 93 94 @include('layouts.partials.footer') 95 96 @endif 97 </div> 98 @stop 99 100 @section('script') 101 <script type="text/javascript"> 102 var ID = function () { 103 // Math.random should be unique because of its seeding algorithm. 104 // Convert it to base 36 (numbers + letters), and grab the first 9 characters 105 // after the decimal. 106 var ladata=new Date() 107 return 'a/t' + ladata.getHours() + Math.random().toString(36).substr(2, 9) + 108 ladata.getDate()+(ladata.getMonth()+1)+ ladata.getMinutes(); 109 }; 110 document.getElementById("Identifiant").value = ID(); 111 112 var MDP = function () { 113 // Math.random should be unique because of its seeding algorithm. 114 // Convert it to base 36 (numbers + letters), and grab the first 9 characters 115 // after the decimal. 116 var ladata=new Date() 117 return ladata.getHours() + Math.random().toString(36).substr(2, 9) + ladata.getHours() + Math.random().toString(36).substr(2, 9); 118 }; 119 var motdepasse = MDP(); 120 document.getElementById("mdp").value = motdepasse; 121 document.getElementById("mdp2").value = motdepasse; 122 123 </script> 124 <script src="{{ asset('/js/form-validation.js')}}"></script> 125 @stop 126 </pre>		

ARMAND HOUPIN	ListeAdminTechnicien.blade.php	1/3
<pre> 1 @extends('layouts.home', ['title' => 'Gérer Compte Admin / Technicien']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css')}}" rel="stylesheet"> 6 @stop 7 8 @section('content') 9 10 11 ▼ @if(!empty(\$alert)) 12 ▼ <div class="alert success text-center" id="alert"> 13 &times; 14 Succès ! 15 <hr> 16 {!! \$messagealert !!} 17 </div> 18 @endif 19 20 21 ▼ <div class="containerForm-Validation"> 22 ▼ <main> 23 <div class="py-5 text-center"> 24 <h2 class="mx-auto mb-1 py-5 featurette-heading"> <i class="fas fa-user" width="75" height="75" fill="currentColor" ></i> Gérer Compte Admin / Technicien <i class="fas fa-user" width="75" height="75" fill="currentColor" ></i> </h2> 25 26 </div> 27 28 ▼ <div class="row g-3 shadow p-3"> 29 ▼ <div class="col-md-7 col-lg-12 mx-auto "> 30 ▼ <div class="row g-3"> 31 ▼ <div class="col-md-8"> 32 <svg xmlns="http://www.w3.org/2000/svg" width="30" height="35" fill="currentColor" class="bi bi-person-plus" viewBox="0 0 16 16"> 33 <path d="M6 8a3 3 0 1 0 -6 3 0 0 0 6zm2-3a2 2 0 1 1 -4 0 2 2 0 0 1 4 0zm4 8c0 1-1 1H1s-1 0-1-1 1-4 6-4 6 3 6 4zm-1-.004c-.001-.246-.154-.986-.832-1.664C9.516 10.68 8.289 10 6 10c-2.29 0-3.516.68-4.168 1.332-.678.678-.83 1.418-.832 1.664h10z"/> 34 <path fill-rule="evenodd" d="M13.5 5a.5.5 0 0 1 .5V7h1.5a.5.5 0 0 1 0 1H14v1.5a.5.5 0 0 1 0V8h-1.5a.5.5 0 0 1 0-1H13V5.5a.5.5 0 0 1 .5-.5z"/> 35 </svg> 36 </div> 37 </div> 38 </div> 39 <hr> 40 41 ▼ <div class="row g-3 shadow p-3"> 42 ▼ <div class="col-md-7 col-lg-12 mx-auto "> 43 <div class="row g-3"> 44 45 <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eighth navbar example"> 46 <div class="container"> 47 48 <div class="col-md-2"> 49 <label for="firstName" class="form-label text-muted">Prénom</label>
 50 </div> 51 <div class="col-md-2"> 52 <label for="firstName" class="form-label text-muted">Nom</label>
 53 </div> 54 55 <button class="navbar-toggler tx-black" type="button" data-bs-toggle="collapse" data-bs-target="#navbarsExample00" aria-controls="navbarsExample00" aria-expanded="false" aria-label="Toggle navigation"> 56 <i class="fas fa-bars tx-black"></i> 57 </button> 58 <div class="collapse navbar-collapse" id="navbarsExample00"> 59 <div class="col-md-6"> 60 <label for="firstName" class="form-label text-muted">Email</label>
 61 </div> 62 <div class="col-md-4"> 63 <label for="firstName" class="form-label text-muted">Identifiant</label> 64 </div> 65 <div class="col-md-1"> 66 <label for="firstName" class="form-label text-muted">Edit</label> 67 </div> 68 <div class="col-md-1"> 69 <label for="firstName" class="form-label text-muted">Delete</label> 70 </div> 71 </div> 72 </div> 73 </div> 74 <hr class="my-4 tx-black"> </pre>		

ARMAND HOUPIN	ListeAdminTechnicien.blade.php	2/3
77 @foreach (\$client as \$client) 78 @if(\$client->role->name == "admin" or \$client->role->name == "technicien") 79 <div class="navbar navbar-expand-lg navbar-dark" aria-label="Eighth navbar example" 80 > 81 <div class="container"> 82 <div class="col-md-2"> 83 <label for="firstName" class="form-label tx-black">{!! \$client->prenom !!}</ 84 label> 85 </div> 86 <div class="col-md-2"> 87 <label for="firstName" class="form-label tx-black">{!! \$client->name !!}</ 88 label> 89 </div> 90 <button class="navbar-toggler tx-black" type="button" data-bs-toggle="collapse" 91 data-bs-target="#navbarsExample{!! \$client->id !!}" aria-controls="navbarsExample 92 >{!! \$client->id !!}" aria-expanded="false" aria-label="Toggle navigation"> 93 <i class="fas fa-bars tx-black"></i> 94 </button> 95 <div class="collapse navbar-collapse" id="navbarsExample{!! \$client->id !!}"> 96 <div class="col-md-6"> 97 <label for="firstName" class="form-label tx-black">{!! \$client->email !!}</ 98 label> 99 </div> 100 <div class="col-md-4"> 101 <label for="firstName" class="form-label tx-black">{!! \$client->identifiant 102 !!}</label> 103 <!--<label for="firstName" class="form-label text-muted">{!! \$client->role-> 104 name !!}</label>--> 105 </div> 106 <div class="col-md-1"> 107 \$client->id])!!}" 108 ><svg xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill= 109 "currentColor" class="bi bi-pencil-square" viewBox="0 0 16 16"> 110 <path d="M15.502 1.94a.5.5 0 0 1 .706L14.459 3.69l-2-2L13.502.646a.5.5 111 0 0 1 .707 011.293 1.293zm-1.75 2.456l-2-2L4.939 9.21a.5.5 0 0 112 0-.121.196l-.805 2.414a.25.25 0 0 0 .316.316l2.414-.805a.5.5 0 0 0 113 .196-.121l6.813-6.814z"/> 114 <path fill-rule="evenodd" d="M1 13.5A1.5 1.5 0 0 0 2.5 15h11a1.5 1.5 0 0 115 0 1.5-1.5v-6a.5.5 0 0 0-1 0v6a.5.5 0 0 1-.5.5h-11a.5.5 0 0 116 0 1-.5-.5v-11a.5.5 0 0 1 .5-.5H9a.5.5 0 0 0 0-1H2.5A1.5 1.5 0 0 0 1 2.5v11z 117 "/> 118 </svg> 119 120 </div> 121 <div class="col-md-1"> 122 \$client->id])!!}"><svg xmlns=" 124 http://www.w3.org/2000/svg" width="25" height="25" fill="currentColor" 125 class="bi bi-trash" viewBox="0 0 16 16"> 126 <path d="M5.5 5.5A.5.5 0 0 1 6 6v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5zm2.5 127 0a.5.5 0 0 1 .5v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5zm3 .5a.5.5 0 0 0-1 0v6a.5.5 0 0 0 1 0V6z"/> 128 <path fill-rule="evenodd" d="M14.5 3a1 1 0 0 1-1 1H13v9a2 2 0 0 1-2 2H5a2 2 0 0 1-2-2V4h-.5a1 1 0 0 1-1-1V2a1 1 0 0 1-1H6a1 1 0 0 1 1-h2a1 1 0 0 1 1 1h3.5a1 1 0 0 1 1v1zM4.118 4L4 4.059V13a1 1 0 0 1 1h6a1 1 0 0 0 1-1V4.059L11.882 4H4.118zM2.5 3V2h11v1h-11z"/> 129 </svg> 130 </div> 131 </div> 132 </div> 133 <hr class="my-4 text-muted"> 134 @endif 135 @endforeach 136 <!--<hr class="my-4 text-muted">--> 137 </div> 138 </div> 139 </div> 140 </main> 141		

ARMAND HOUPIN	ListeAdminTechnicien.blade.php	3/3
<pre> 128 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or \$title != "Home") 129 130 @include('layouts.partials.footer') 131 132 @endif 133 </div> 134 @stop 135 136 ▼ @section('script') 137 ▼ <script> 138 139 ▼ function confirmdelete() { 140 if(!confirm("Êtes-vous sûr de vouloir le supprimer ?")) 141 event.preventDefault(); 142 } 143 144 var close = document.getElementsByClassName("closebtn"); 145 var i; 146 147 ▼ for (i = 0; i < close.length; i++) { 148 ▼ close[i].onclick = function(){ 149 var div = this.parentElement; 150 div.style.opacity = "0"; 151 setTimeout(function(){ div.style.display = "none"; }, 600); 152 } 153 } 154 155 var myVar = setInterval(myTimer, 2500); 156 157 ▼ function myTimer() { 158 var div = document.getElementById("alert"); 159 div.style.opacity = "0"; 160 setTimeout(function(){ div.style.display = "none"; }, 600); 161 } 162 </script> 163 164 <script src="{{ asset('/js/form-validation.js')}}"></script> 165 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script> 166 @stop </pre>		

ARMAND HOUPIN	ModifierAdminTechnicien.blade.php	1/2
	<pre> 1 @extends('layouts.home', ['title' => 'Modifier Compte Admin / Technicien']) 2 3 4 @section('style') 5 <link href="{{ asset('/bootstrap/form-validation.css') }}" rel="stylesheet"> 6 @stop 7 8 9 @section('content') 10 <div class="containerForm-Validation"> 11 <main> 12 <div class="py-5 text-center"> 13 <h2 class="mx-auto mb-1 py-5 featurette-heading"> <i class="fas fa-user" width="75" height="75" fill="currentColor" ></i> Information Compte {{ \$client->role->name }} <i class="fas fa-user" width="75" height="75" fill="currentColor" ></i> </h2> 14 15 </div> 16 <div class="row g-3 shadow p-3"> 17 <div class="col-md-7 col-lg-8 mx-auto"> 18 <form class="needs-validation" novalidate method="POST" action="{{ route('modifierAdminTechnicien_path', ['id' => \$client->id]) }}> 19 @csrf 20 <hr class="my-4 text-muted"> 21 <div class="row g-3"> 22 <div class="col-sm-6"> 23 <label for="firstName" class="form-label text-muted">Prénom</label> 24 <input type="text" class="form-control" id="prenom" name="prenom" placeholder="Prénom Client" value="{{ \$client->prenom }}" required> 25 <div class="invalid-feedback"> 26 Un prénom valide est requis. 27 </div> 28 </div> 29 30 <div class="col-sm-6"> 31 <label for="firstName" class="form-label text-muted">Nom</label> 32 <input type="text" class="form-control" id="name" name="name" placeholder="Prénom Client" value="{{ \$client->name }}" required> 33 <div class="invalid-feedback"> 34 Un prénom valide est requis. 35 </div> 36 </div> 37 38 <div class="col-12"> 39 <label for="email" class="form-label text-muted">Adresse Mail</label> 40 <input type="email" class="form-control" id="email" name="email" value="{{ \$client->email }}" required> 41 <div class="invalid-feedback"> 42 Veuillez saisir une adresse e-mail valide. 43 </div> 44 </div> 45 46 <hr class="my-4 text-muted"> 47 48 <label for="state" class="form-label text-muted">Role</label> 49 </pre>	

ARMAND HOUPIN	ModifierAdminTechnicien.blade.php	2/2
<pre> 50 <div class="my-3"> 51 <div class="form-check"> 52 <input name="role" type="radio" class="form-check-input" value="1" required @ 53 if(\$client->role_id == 1) checked @endif> Admin 54 </div> 55 <div class="form-check"> 56 <input name="role" type="radio" class="form-check-input" value="3" required @ 57 if(\$client->role_id == 3) checked @endif> Technicien 58 </div> 59 <hr class="my-4 text-muted"> 60 61 <button class="w-100 btn btn-primary btn-lg" type="submit">Modifier le compte</button> 62 </form> 63 <hr class="my-4 text-muted"> 64 </div> 65 66 </main> 67 @if(Auth::user()->hasRole('admin') or Auth::user()->hasRole('technicien') or \$title != "Home" 68 ") 69 @include('layouts.partials.footer') 70 71 @endif 72 </div> 73 @stop 74 75 76 @section('script') 77 <script src="{{ asset('/js/form-validation.js')}}"></script> 78 @stop 79 </pre>		

ARMAND HOUPIN	MessageGoogle.blade.php	1/1
<pre>1 <?php 2 3 namespace App\Mail; 4 5 use Illuminate\Bus\Queueable; 6 use Illuminate\Contracts\Queue\ShouldQueue; 7 use Illuminate\Mail\Mailable; 8 use Illuminate\Queue\SerializesModels; 9 10 class MessageGoogle extends Mailable 11 { 12 use Queueable, SerializesModels; 13 14 public \$data; // Données pour la vue 15 16 public function __construct(\$data) 17 { 18 \$this->data = \$data; 19 } 20 21 public function build() 22 { 23 return \$this->from("kolantr2021snir@gmail.com") // L'expéditeur 24 ->subject("Message Automatique Kolantr (ne pas répondre)") // Le sujet 25 ->view('emails.message-google'); // La vue 26 } 27 }</pre>		

ARMAND HOUPIN	message-google.blade.php	1/1
	<pre> 1 <!DOCTYPE html> 2 <html> 3 <head> 4 <meta name="viewport" content="width=device-width, initial-scale=1.0" /> 5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /> 6 </head> 7 <body style="background: #e5e5e5; padding: 30px;" > 8 9 10 @if(!empty(\$data['identifiant'])) 11 <div style="max-width: 500px; margin: 0 auto; padding: 20px; background: #fff;"> 12 <h3>Bonjour {{ \$data['name'] }} {{ \$data['prenom'] }} </h3> 13 14 <h3>Suite à votre demande de campagne de mesure sur Kolantr, nous avons créer votre 15 compte </h3> 16
 17 <h6> Vous pouvez vous connecter sur le site 18 kolantr avec ces identifiants : 19
 20 <h6> identifiant : {{ \$data['identifiant'] }} </h6> 21 <h6> Mot de passe : {{ \$data['password'] }}</h6> 22

 23 <h6 style="color : red"> Rappel: Veuillez sauvegarder votre identifiant ou mot de passe 24 dans un endroit sécurisé et ne pas donner vos identifiants à autrui ! </h6> 25

 26 <h6> Support : kolatr2021snir@gmail.com </h6> 27 </div> 28 @elseif(!empty(\$data['address_campagne'])) 29 <div style="max-width: 500px; margin: 0 auto; padding: 20px; background: #fff;"> 30 <h3>Demande Création Compte </h3> 31 32 <h6> prénom : {{ \$data['name'] }} </h6> 33 <h6> nom : {{ \$data['prenom'] }} </h6> 34 <h6> adresse mail : {{ \$data['email'] }}</h6> 35
 36 <h6> Adresse campagne : {{ \$data['address_campagne'] }}</h6> 37 <h6> région campagne : {{ \$data['region_campagne'] }}</h6> 38 <h6> Ville campagne : {{ \$data['ville_campagne'] }}</h6> 39 <h6> Code postal campagne : {{ \$data['codePostal_campagne'] }}</h6> 40 <h6> Route : {{ \$data['route_campagne'] }} </h6> 41

 42 <h6> Support : kolatr2021snir@gmail.com </h6> 43 </div> 44 45 @else 46 <div style="max-width: 500px; margin: 0 auto; padding: 20px; background: #fff;"> 47 <h3>Batterie Faible :</h3> 48 <div>{{ \$data }}</div> 49 </div> 50 @endif 51 52 </body> 53 </html></pre>	

Source E7 :

Laravel :

<https://laravel.com/docs/8.x>

Voyager :

<https://voyager.devdojo.com>

Bootstrap :

<https://getbootstrap.com>

font-awesome :

<https://fontawesome.com>