

Mardi 4 février : Début du projet. Répartition des tâches dans le projet.

Mercredi 5 février : Documents fournis pour création de diagramme pour le projet. Début de création de diagramme de classe et du cahier des charges.

Jeudi 6 février : Reprise du diagramme de classe et du cahier des charges + création de diagramme d'utilisation et de contexte. Création d'un suivi de projet.

Vendredi 7 février : Reprise diagramme. Réunion avec prof pour explication. Envoi mail de réunion avec professionnel.

Du 17 au 21 : Réalisation de l'Étude de projet. Analyse personnelle pour comprendre certains aspects du projet. Etude des possibilités de configuration wifi, réflexion sur le contenu des trames réseau, réflexion sur l'IHM.

Mardi 25 février : Partage de "l'étude de projet" au groupe de projet. Réunion avec le prof à ce sujet, des remarques sont faites. Reprise du cahier des charges pour les prendre en compte.

Mercredi 26 février : Suite du cahier des charges. Réalisation d'un diagramme de contexte

Jeudi 27 février : re-conception du diagramme de contexte et enregistrement. Réalisation d'un diagramme de cas d'utilisation.

Vendredi 28 février : Remarque du prof qui nous conseille de créer des diagrammes pour chaque partie et non faire des diagrammes "globaux". Création d'un diagramme de contexte partie IHM. Création d'un diagramme de cas d'utilisation partie IHM. Début du sommaire document.

Du 28 au 1 : Reprise du diagramme de contexte partie IHM. Reprise du diagramme de cas d'utilisation partie IHM. Création d'un diagramme d'activité partie IHM.

Lundi 2 mars : Suite du sommaire d'activité.

Mardi 3 mars : Reprise du sommaire d'activité, en y ajoutant le diagramme de contexte partie IHM, le diagramme de cas d'utilisation partie IHM et le diagramme d'activité partie IHM. Création d'une première version d'une IHM.

Mercredi 4 mars : Suite de création de prototype d'IHM. La première page réalisée sera la page "connexion". 2 versions ont été créées.

Jeudi 5 mars : Suite de création d'IHM. Création d'une troisième version sur MFC.

Vendredi 6 mars : Suite de la création de la troisième version d'IHM sur MFC. Création d'une liste de matériel avec les références, données... Modification du cahier des charges.

Lundi 9 mars : recherche sur les bases de données et sur la communication client-serveur.

Mardi 10 mars : suite des recherches sur les bases de données et sur la communication client-serveur. Lecture de tutoriaux.

Mercredi 11 mars : suite des recherches sur les bases de données. J'ai trouvé qu'il existe une librairie MySQL pour un programme C++ : l'API MySQL. Capture d'écran des pages de mon prototype d'IHM.

Jeudi 12 mars : recherche sur les bases de données. Nouveau tuto. Relecture du TP sur l'API MySQL et l'environnement WAMP.

Vendredi 13 mars : suite du travail sur les bases de données. Ai refait le TP sur l'API MySQL vu en cours.

Samedi et Dimanche 15 mars : installation et configuration à la maison d'un pc de travail dédié au projet.

Du 16 au 19 mars : Confinement. J'ai installé et configuré un serveur MySQL et PHPmyAdmin (environnement WAMP) pour pouvoir créer une base de données sur mon PC. Découverte du site "Openclassrooms" dont je trouve une leçon sur les bases de données qui m'aide à créer la structure de ma base de données : le concept, les notions qui me manquaient, et les commandes SQL. J'utilise la console SQL manuellement pour créer ma base et mes tables. Le lien du site :

<https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql>

Création des premières tables avec les commandes suivantes :

- `CREATE DATABASE RMS CHARACTER set 'utf8';`
- `USE RMS;`
- `CREATE TABLE Client ( IdNom smallint unsigned auto_increment, Nom varchar(255) not null, Prenom varchar(255) not null, PRIMARY KEY (IdNom) );`
- `CREATE TABLE Config_bouton ( NumeroBouton tinyint unsigned, Rouge tinyint unsigned, Vert tinyint unsigned, Bleu tinyint unsigned );`
- `CREATE TABLE Sequence ( IdSequence smallint unsigned auto_increment, ListeEtapes varchar(255) not null, NbreRepetition tinyint unsigned, PRIMARY KEY (IdSequence) );`
- `CREATE TABLE Tests ( IdTestClient int unsigned auto_increment, Resultat varchar(255), Date datetime not null, PRIMARY KEY (IdTestClient) );`

Du 20 au 22 : Modification des tables et des commandes SQL associées :

`CREATE DATABASE RMS CHARACTER set 'utf8';`

`USE RMS;`

`CREATE TABLE Client ( IdNom smallint unsigned auto_increment, Nom varchar(255) not null, Prenom varchar(255) not null, PRIMARY KEY (IdNom) );`

`CREATE TABLE Config_bouton ( NumeroBouton tinyint unsigned, Rouge tinyint unsigned, Vert tinyint unsigned, Bleu tinyint unsigned );`

`CREATE TABLE Sequence ( IdSequence smallint unsigned auto_increment, ListeEtapes varchar(255) not null, NbreRepetition tinyint unsigned, PRIMARY KEY (IdSequence) );`

`CREATE TABLE Tests ( IdTestClient int unsigned auto_increment, IdNom smallint unsigned, IdSequence smallint unsigned, Resultat varchar(255), Date datetime not null, PRIMARY KEY (IdTestClient) );`

Du 23 au 1<sup>er</sup> avril : Mise en pause du projet car je dois faire plein de devoir à rendre en ligne pour la semaine + retravailler ma présentation pour la revue de stage qui est normalement le 1 avril. Puis abandon de la soutenance de stage du au confinement.

Jeudi 2 avril : Conception du support de ma première revue de projet.

Vendredi 3 avril : Préparation de ma revue de projet : j'ai amélioré ma première partie (les généralités) et j'attaque la suite sur ma partie du projet. Envoi au professeur la première partie pour correction.

Du 4 au 6 : Pause vacances.

Mardi 7 avril : Reprise de la partie "Base de données" du projet. Suite de l'étude des notions de base de données, pour comprendre l'indexation, les clés primaires, et les relations extérieures, etc.

Mercredi 8 avril : Suite de l'étude de MYSQL. Création d'une deuxième version de ma base de données comprenant en plus des clés étrangères qui permettent de créer des relations entre les tables de ma base. Ce qui me donne ces nouvelles lignes de commandes SQL :

```
CREATE DATABASE rms CHARACTER set 'utf8';
```

```
USE rms;
```

```
CREATE TABLE client ( IdNom smallint unsigned auto_increment, Nom varchar(255) not null, Prenom  
varchar(255) not null, PRIMARY KEY (IdNom) )ENGINE=INNODB;
```

```
CREATE TABLE config_bouton ( NumeroBouton tinyint unsigned, Rouge tinyint unsigned, Vert tinyint  
unsigned, Bleu tinyint unsigned )ENGINE=INNODB;
```

```
CREATE TABLE sequence ( IdSequence smallint unsigned auto_increment, ListeEtapes varchar(255) not  
null, NbreRepetition tinyint unsigned, PRIMARY KEY (IdSequence) )ENGINE=INNODB;
```

```
CREATE TABLE tests ( IdTestClient int unsigned auto_increment, IdNom smallint unsigned, IdSequence  
smallint unsigned, Resultat varchar(255), Date datetime not null, PRIMARY KEY  
(IdTestClient) )ENGINE=INNODB;
```

```
ALTER TABLE tests ADD CONSTRAINT relationIdNom FOREIGN KEY (IdNom) REFERENCES  
Client(IdNom);
```

```
ALTER TABLE tests ADD CONSTRAINT relationIdSeq FOREIGN KEY (IdSequence) REFERENCES  
sequence(IdSequence);
```

Réalisation de première ligne de commande pour insérer des données dans les tables (champ dans les tables):

```
INSERT INTO `client`(`IdNom`,`Nom`,`Prenom`) VALUES (NULL,'Durant','David');
```

```
INSERT INTO `sequence`(`IdSequence`,`ListeEtapes`,`NbreRepetition`) VALUES  
(NULL,'0302010405',0);
```

```
INSERT INTO `tests`(`IdTestClient`,`IdNom`,`IdSequence`,`Resultat`,`Date`) VALUES  
(NULL,1,1,'100090','2020-04-08 13:43:00');
```

```
INSERT INTO `tests`(`IdTestClient`,`IdNom`,`IdSequence`,`Resultat`,`Date`) SELECT NULL, IdNom,  
NULL, '100099031', '2020-04-08 13:30:00' FROM client WHERE Nom = 'Durant' AND Prenom = 'David'
```

9 et 10 avril : Vérification et validation de mes tables avec une mise à jour des schémas. Ajout de données dans mes tables pour les tests. Ecriture en C++ du programme de création de la base et de ses tables. Je suis aidé de ces liens suivants pour réaliser le code :

<https://dev.mysql.com/doc/connector-cpp/1.1/en/connector-cpp-examples-complete-example-2.html>

<https://dev.mysql.com/doc/connectors/en/connector-j-usagenotes-statements.html>

Le code C++ réalisé :

```

/* Standard C++ includes */
#include <stdlib.h>
#include <iostream>

/*
    Include directly the different
    headers from cppconn/ and mysql_driver.h + mysql_util.h
    (and mysql_connection.h). This will reduce your build time!
*/
#include <mysql/jdbc.h>    //1 seul include désormais

//programme principal
int main(void)
{
    //déclaration des variables utilisées dans le programme pour MySQL
    sql::Driver*      ptr_driver = nullptr;
    sql::Connection* ptr_con  = nullptr;
    sql::Statement*   ptr_stmt = nullptr;
    sql::ResultSet*    ptr_res  = nullptr;
    sql::SQLString     element, element2;

    sql::SQLString joindreServeurSQL("tcp://127.0.0.1:3306");
    sql::SQLString login("David_Durant");
    sql::SQLString pwd("david_MYSQL");

    std::cout << std::endl;
    std::cout << "Acces a la base de donnees MySQL avec connector c++ ..." << std::endl;
    std::cout << std::endl;

    try
    {
        /* Create a connection */
        ptr_driver = get_driver_instance();
        ptr_con = ptr_driver->connect(joindreServeurSQL, login, pwd);

        ptr_stmt = ptr_con->createStatement();

        ptr_stmt->execute("CREATE DATABASE rms CHARACTER set 'utf8'");
        ptr_stmt->execute("USE rms");
        ptr_stmt->execute("CREATE TABLE client ( IdNom smallint unsigned auto_increment, Nom varchar(255) not null,
        Prenom varchar(255) not null, DateNaissance date not null, PRIMARY KEY (IdNom) ) ENGINE=INNODB");
        ptr_stmt->execute("CREATE TABLE config_bouton ( NumeroBouton tinyint unsigned, Rouge tinyint unsigned, Vert
        tinyint unsigned, Bleu tinyint unsigned ) ENGINE=INNODB");
        ptr_stmt->execute("CREATE TABLE sequence ( IdSequence smallint unsigned auto_increment, ListeEtapes var-
        char(255) not null, NbreRepetition tinyint unsigned, NomSequence varchar(50) not null, PRIMARY KEY (IdSequence) ) ENGINE=IN-
        NODB");
        ptr_stmt->execute("CREATE TABLE tests ( IdTestClient int unsigned auto_increment, IdNom smallint unsigned,
        IdSequence smallint unsigned, Resultat text, TempsMinimum decimal(3,1) unsigned, TempsMaximum decimal(3,1) unsigned,
        TempsMoyen decimal(3,1) unsigned, EcartType decimal(3,1) unsigned, NbreEtapesExecutees int, DureeTotale decimal(5,1), Date
        datetime not null, PRIMARY KEY (IdTestClient) ) ENGINE=INNODB");
        ptr_stmt->execute("ALTER TABLE Tests ADD CONSTRAINT relationIdNom FOREIGN KEY (IdNom) REFERENCES Client(Id-
        Nom)");
        ptr_stmt->execute("ALTER TABLE Tests ADD CONSTRAINT relationIdSeq FOREIGN KEY (IdSequence) REFERENCES se-
        quence(IdSequence)");

        delete ptr_stmt;

        /* Connect to the MySQL database */
        ptr_con->setSchema("rms");
    }
}
//ici vous pouvez modifier le nom de la base de
données sur laquelle vous allez accéder

```

```

/* Interrogation de la base de données */
ptr_stmt = ptr_con->createStatement();
ptr_res = ptr_stmt->executeQuery("SHOW tables");           //ici vous pouvez modifier votre requête sql

while (ptr_res->next())           //parcours de la réponse à la requête sql
{
    std::cout << "\t... MySQL replies: ";
    /* Access column data by alias or column name */
    //std::cout << ptr_res->getString("User") << std::endl;

    element = ptr_res->getString("tables_in_rms");
    std::cout << element.c_str() << std::endl;

    std::cout << "\t... MySQL says it again: ";
    /* Access column data by numeric offset, 1 is the first column */
    //std::cout << ptr_res->getString(1) << std::endl;
    element2 = ptr_res->getString(1);
    std::cout << element2.c_str() << std::endl;
}

/* Libération des éléments affectés */
delete ptr_res;
delete ptr_stmt;

std::cout << std::endl;
std::cout << "Deuxieme requete" << std::endl;
std::cout << std::endl;

/* Interrogation de la base de données */
ptr_stmt = ptr_con->createStatement();
ptr_res = ptr_stmt->executeQuery("DESCRIBE tests");           //ici vous pouvez modifier votre requête

while (ptr_res->next())           //parcours de la réponse à la requête sql
{
    std::cout << "\t... MySQL replies: ";
    /* Access column data by alias or column name */
    //std::cout << ptr_res->getString("User") << std::endl;

    element = ptr_res->getString("field");
    std::cout << element.c_str() << std::endl;

    std::cout << "\t... MySQL says it again: ";
    /* Access column data by numeric offset, 1 is the first column */
    //std::cout << ptr_res->getString(1) << std::endl;
    element2 = ptr_res->getString(1);
    std::cout << element2.c_str() << std::endl;
}

/* Libération des éléments affectés */
delete ptr_res;
delete ptr_stmt;
delete ptr_con;

} //fin try

catch (sql::SQLException& e)
{
    std::cout << "# ERR: SQLException in " << __FILE__;
    std::cout << "(" << __FUNCTION__ << ")" on line " << __LINE__ << std::endl;
    std::cout << "# ERR: " << e.what();
    std::cout << " (MySQL error code: " << e.getErrorCode();
    std::cout << ", SQLState: " << e.getSQLState() << ")" << std::endl;
} //fin catch

std::cout << std::endl;
std::cout << "Appuyer sur une touche pour quitter...";
std::cin.get();           //attente de l'appui sur une touche

return EXIT_SUCCESS;
} //fin programme main()

```

Lundi 13 avril : Comme la partie "Base de donnée" est avancée, je décide de passer à la partie "IHM". Recherche sur les applications IHM autre que MFC.

Mardi 14 avril : suite des recherches sur l'IHM, retour à MFC. Comment enchaîner plusieurs menus dans la même fenêtre. Lecture de tutoriaux MFC dans YouTube de Pekka K.

Mercredi 15 avril : Création d'un premier test d'IHM avec 2 fenêtres (1 fenêtre et 1 calque) et de 1 bouton sur chaque fenêtre. L'objectif est que quand on appuie sur un bouton, on passe sur l'autre page.

Ligne de code modifiée :

```
Dans CIHMtest2Dlg::OnInitDialog()  
page1.Create(IDD_PAGE_1, this);  
page1.ShowWindow(SW_SHOW);
```

```
void CIHMtest2Dlg::OnBnClickedMainButton()  
{  
    // TODO: ajoutez ici le code de votre gestionnaire de notification de contrôle  
    page1.ShowWindow(SW_SHOW);  
    ShowWindow(SW_HIDE);  
}
```

Jeudi 16 avril : Création d'un deuxième test d'IHM MFC avec 1 fenêtre et 2 calques et de 1 bouton sur chaque fenêtre/calque. L'objectif est de pouvoir afficher plusieurs suites de boutons différentes dans la même fenêtre de dialogue, sans avoir à faire une fenêtre de dialogue par menu et par sous menus.

On teste que l'on affiche le premier calque sur la fenêtre principale qui est vide, et puis passage d'une page à l'autre si on appuie sur le bouton correspondant de la page. On reste sur un échange entre le premier calque et le deuxième.

Ligne de code modifiée :

```
Dans CIHMtest2Dlg::OnInitDialog()  
page1.Create(IDD_PAGE_1, this);  
page2.Create(IDD_PAGE_2, this);  
page1.ShowWindow(SW_SHOW);
```

```
Dans CPage1.cpp  
void CPage1::OnBnClickedButtonP2()  
{  
    page1.ShowWindow(SW_HIDE);  
    page2.ShowWindow(SW_SHOW);  
}
```

```
Dans CPage2.cpp  
void CPage2::OnBnClickedExitP1()  
{  
    // TODO: ajoutez ici le code de votre gestionnaire de notification de contrôle  
    page2.ShowWindow(SW_HIDE);  
    page1.ShowWindow(SW_SHOW);  
}
```

Vendredi 17 avril : Travail sur le support de présentation de la revue de projet. Intégration des remarques et des corrections de monsieur Dudreuil sur la première partie. Modification des diagrammes généraux aussi. Répétitions sur la première partie.

18 et 19 avril : Suite de la préparation de ma revue de projet : écriture du support pour la seconde partie, celle de mon travail. Animations, réalisation des diagrammes, texte. 4 répétitions et modifications du contenu pour tenir dans le temps donné.

Lundi 20 avril : répétition pour une revue de projet blanc à 11h avec monsieur Dudreuil. Remarques de monsieur Dudreuil sur les diagrammes, et sur le rôle du bouton poussoir maître, et sur la définition des séquences. Aussi une remarque très importante sur la notion d'un temps maximum d'éclairage d'un bouton poussoir esclave que je n'avais pas compris. Cela remet en question mon étude et le fonctionnement que j'avais imaginé. Remarques sur l'organisation de ma base de données : les tables ne sont pas claires et contiennent des champs trop complexes. Modification du support de présentation pour tenir compte des remarques. Et répétitions de la revue de projet entière.

Mardi 21 avril : Répétition de la revue de projet avant mon passage final à 8h25. Questions posées par les profs : pourquoi attendre la fin d'une séquence pour que le bouton maître renvoie tous les temps ? Pourquoi le bouton maître ne renvoie pas au fur et à mesure chaque temps d'appui d'un bouton poussoir esclave ? Pourquoi ne pas avoir mis un diagramme de gantt à la place de mes diagrammes à bulles successives à la fin de ma présentation ? Modification de mes doc à remettre en fonction des remarques d'hier sur le bouton maître principal.

Du 22 au 24 avril : Après la revue de projet, reprise de la modélisation des pages IHM. Après avoir trouvé comment fonctionne le passage entre plusieurs pages, je me mets à la recherche de comment colorier le fond d'une page ou comment mettre une image de fond à la page.

Dimanche 26 avril : Méthode pour colorier le fond d'une page par une couleur unie, techniques dans ces pages :

<https://docs.microsoft.com/fr-fr/cpp/mfc/setting-the-dialog-boxs-background-color?view=vs-2019>

<https://docs.microsoft.com/fr-fr/cpp/mfc/reference/mfc-class-wizard?view=vs-2019>

<http://codexpert.ro/blog/2013/03/13/painting-the-dialog-background/>

Deuxième approche : charger une image en fond de Dialogbox, pour avoir d'un côté les boutons et les infos d'aide sur cette image de fond. Et ne plus avoir à gérer qui est au premier plan ou derrière, le problème rencontré si on utilise une PictureBox avec un bouton dessus.

<https://stackoverflow.com/questions/2091203/load-a-cbitmap-dynamically>

Ai trouvé comment figer la même taille pour la dialog box et les page1 et page2. Dans le fichier .rc

```
IDD_IHM_MAIN_DIALOG DIALOGEX 0, 0, 400, 300
```

```
IDD_PAGE_2 DIALOGEX 0, 0, 400, 300
```

```
IDD_PAGE_1 DIALOGEX 0, 0, 400, 300
```

Lundi 27 avril : Application de la méthode pour charger une image en fond de dialogbox.

Mardi 28 avril : Après une définition schématique de mes pages principales, réalisation de ces pages en créant une autre version du projet "test3" qui inclus une page de fond, une page principale, une page gestion patient, une page définition séquence, une page paramétrage, une page exécution séance, avec des boutons et un fond de page pour chaque page. Je trouve aussi une méthode pour mettre plus facilement une image en fond de dialog : `SetBackgroundImage()`;



29 et 30 avril : programmation des boutons de "retour page précédente". Recherche sur la méthode pour charger une image de fond qui puisse s'étendre si on change la taille de la fenêtre de dialogue pendant son utilisation. N'ayant rien trouvé de concret, je décide de fixer les tailles de mes pages de dialogue et d'enlever la variable "WS\_MAXIMIZEBOX" qui permet de pouvoir agrandir ma boîte de dialogue pendant son utilisation.

Samedi 2 Mai : Reprise sur la recherche sur la méthode pour charger une image de fond qui puisse s'étendre si on change la taille de la fenêtre de dialogue pendant son utilisation. Pour ne pas trop modifier le code de mon projet en faisant mes recherches et application, je décide de créer un petit projet à part qui permet de faire mes recherches sur ce point sans que cela ne puisse compromettre le précédent code.

Lundi 4 Mai : Reprise sur la recherche sur la méthode de charger une image de fond qui puisse s'étendre si on change la taille de la fenêtre de dialogue pendant son utilisation.

Mardi 5 Mai : Recherche sur comment colorier le fond d'un bouton. Cette page :

<http://www.functionx.com/visualc/controls/mfcbtn.htm>

M'a permis de trouver la méthode `SetFaceColor()`, qui fonctionne en mettant : `EnableWindowsTheming(FALSE)`; Ajustement de certains boutons dans plusieurs pages (tailles, distances) et suppression des boutons "définition séquence" et "stockage temps" qui étaient créés en trop le 28 avril.

Codage du bouton "Fin Programme". De cette page :

<http://www.functionx.com/visualc/controls/mfcbtn.htm>

J'ai extrait la fonction suivante : `PostQuitMessage(0)`;

Mercredi 6 Mai : Sur la page "gérer patient" remplacement de 2 boutons par des listes pour améliorer la sélection d'un tel ou un tel... Mise en place de texte statique et recherche sur comment mettre une écriture coloriée pour les textes statiques. Méthode trouvée et appliquée grâce à ces 2 pages :

<https://stackoverflow.com/questions/1636590/mfc-change-text-color-of-a-cstatic-text-control>

<https://vcpptips.wordpress.com/tag/set-color-for-static-text/>

Début de la création de la page "créer le patient".

Jeudi 7 Mai : Fin de la création de la page "créer le patient". Cette page contient 2 boutons coloriés, 3 textes statiques coloriés, 3 cases de saisie, le tout placé et redimensionné.

Samedi 9 Mai : Reprise sur la recherche sur la méthode de charger une image de fond qui puisse s'étendre si on change la taille de la fenêtre de dialogue pendant son utilisation. Grâce à un débogage, j'ai trouvé qu'une variable locale connaît les dimensions (longueur, largeur) de mon image lorsque on l'a chargée dans le dialogue box (après le `SetBackgroundImage()`; ). J'en déduis ces lignes de code qui me permettent d'afficher le dialogue SANS connaître les dimensions de l'image au préalable :

```
this->SetBackgroundImage(IDB_BITMAP_FPRINCIPAL);  
this->SetWindowPos(NULL, 0, 0, this->m_sizeBkgrBitmap.cx, this->m_sizeBkgrBitmap.cy, NULL);
```

Mardi 12 Mai : Création de la page "Effacer patient" et "Création base client". Début de la création de la page : "définition séquence".

Samedi 16 Mai : Reprise sur la création de la page : "Définition Séquence" en ajoutant les boutons "Définition d'une Séquence", "Modification Séquence", "Effacement Séquence". Création de la page (vide) : "Définition d'une Séquence"



Dimanche 17 Mai : Reprise sur la création de la page "Définition d'une Séquence". Ajout de 5 boutons à cocher pour le temps d'affichage des boutons poussoirs et d'une saisit de texte pour mettre un temps personnalisé. Paramétrage de la saisit du texte pour qu'elle s'affiche seulement lorsque l'on sélectionne l'option "Autres".

Exemple :

Bouton 1 :

```
if (BoutonTempsTest1.GetCheck() == 1)
{
    BoutonTempsEdit.ShowWindow(SW_HIDE);
}
else
{
    BoutonTempsEdit.ShowWindow(SW_HIDE);
}
```

Bouton Autres :

```
if (BoutonTempsTest5.GetCheck() == 1)
{
    BoutonTempsEdit.ShowWindow(SW_SHOW);
}
else
{
    BoutonTempsEdit.ShowWindow(SW_HIDE);
}
```

Travail sur un bouton pour qu'il affiche une image lorsqu'il est sélectionné et une autre image lorsqu'il est non sélectionné.

Dans OnInitDialog :

```
Bmp1.GetBitmap(&aBmp);
imgWidthInPixels = aBmp.bmWidth;
imgHeightInPixels = aBmp.bmHeight;

BoutonTest.ModifyStyle(0, BS_BITMAP);
BoutonTest.SetBitmap((HBITMAP)Bmp1.GetSafeHandle());
BoutonTest.SetWindowPos(NULL, 102, 84, imgWidthInPixels, imgHeightInPixels, NULL);
```

Dans la méthode lorsque l'on clique sur le bouton :

```
static int l_var = 0;

if (l_var == 0)
{
    l_var = 1;
    BoutonTest.ModifyStyle(0, BS_BITMAP);
    BoutonTest.SetBitmap((HBITMAP)Bmp2.GetSafeHandle());
    BoutonTest.SetWindowPos(NULL, 30, 200, imgWidthInPixels, imgHeightInPixels, NULL);
}
else
{
    l_var = 0;
    BoutonTest.ModifyStyle(0, BS_BITMAP);
    BoutonTest.SetBitmap((HBITMAP)Bmp1.GetSafeHandle());
    BoutonTest.SetWindowPos(NULL, 102, 84, imgWidthInPixels, imgHeightInPixels, NULL);
}
```

Lundi 18 Mai : Après la suite du cours MFC d'aujourd'hui, légère optimisation du code. Suppression de cette partie du code :

```
if (nResponse == IDOK)
{
    // TODO: placez ici le code définissant le comportement lorsque la boîte de dialogue
est fermée avec OK
}
else if (nResponse == IDCANCEL)
{
    // TODO: placez ici le code définissant le comportement lorsque la boîte de dialogue
est fermée avec Annuler
}
```

Cela correspond au comportement que doit avoir la boîte de dialogue si on appuie sur les boutons "OK" et "annuler" de la boîte de dialogue générée par défaut. Or je n'utilise pas ces boutons et je les ai supprimés. Donc ce bout de code n'a plus d'intérêt.

Mardi 19 Mai : Recherche sur comment colorier le fond d'un bouton radio et changer la couleur du texte d'un group box pour la page "Définition d'une Séquence".

Du 20 au 22 : Méthode trouvée pour colorier le fond d'un bouton radio et changer la couleur du texte d'un "group box". Intégration d'un code qui me permet d'afficher un nombre de bouton esclave sur la page. Exemple 10 bouton esclave. Re-design de la page "Définition d'une Séquence" en y ajoutant plus de case de radio bouton.

Dimanche 24 Mai : Code qui me permet de personnaliser le texte d'une "static". Le code fait que j'affiche le numéro de l'étape après clic sur bouton étape suivante.

Lundi 1 Juin : Début de création de la page "Historique Résultat" et "Paramétrer Matériels". Ajustement design page "Effacer patient", "Effacer séquence" et "Création Base Patients".

Mardi 2 Juin : Début de création de la page "Historique Résultat"

Mercredi 3 Juin : Début de création de la page "Historique Résultat"

Vendredi 5 juin : ai trouvé une méthode pour que l'image de fond s'élargisse lorsqu'on élargit la fenêtre de l'application. Néanmoins cette méthode ne sera pas incluse car cela rend pixelisé le fond et gâche l'application. Correction au menu principal du nom d'un bouton qui s'appelle maintenant "Gestion séquence". Correction avec le titre d'une page : "Menu gestion patients".

Du 8 au 15 juin : Fin de création des pages "Paramétrer Matériels", "Modification Séquence". Création des pages "Tester Matériels" et "Sauvegarde base patients". Modification de la page "Historique séance". Optimisation de tous les .h des classes de pages pour qu'il respecte la convention C++. Optimisation de code en remplaçant toutes les couleurs "RGB()" par les noms des couleurs déclarées par macros dans un .h nommée "Couleurs.h".

Mardi 16 juin : Fin de création de la page "historique séance" en y ajoutant la zone de texte des résultats historiques manquants. Fin de création de page "Modification séquence" en y ajoutant 2 listes, 2 zones de saisit et 2 boutons et en remaniant le positionnement dans l'espace de la page. Dans la page "Paramétrer Matériel", ajout d'une zone de saisit pour l'adressage IP du bouton maître. Application d'une nouvelle fonction qui me permet de mettre un titre à mes zones de saisit et à mes listes déroulantes. (Fonction : SetCueBanner(LPCTSTR lpszText))

Mercredi 17 juin : Résultat et application d'une recherche sur comment ajouter un retour à la ligne à un texte. (Méthode : ajouter "\r\n" dans le texte). Passage à la base de donnée. Re-modélisation des tables de la base de donnée. Ajout d'une table "config\_rms". Ligne de code pour la console MYSQL :

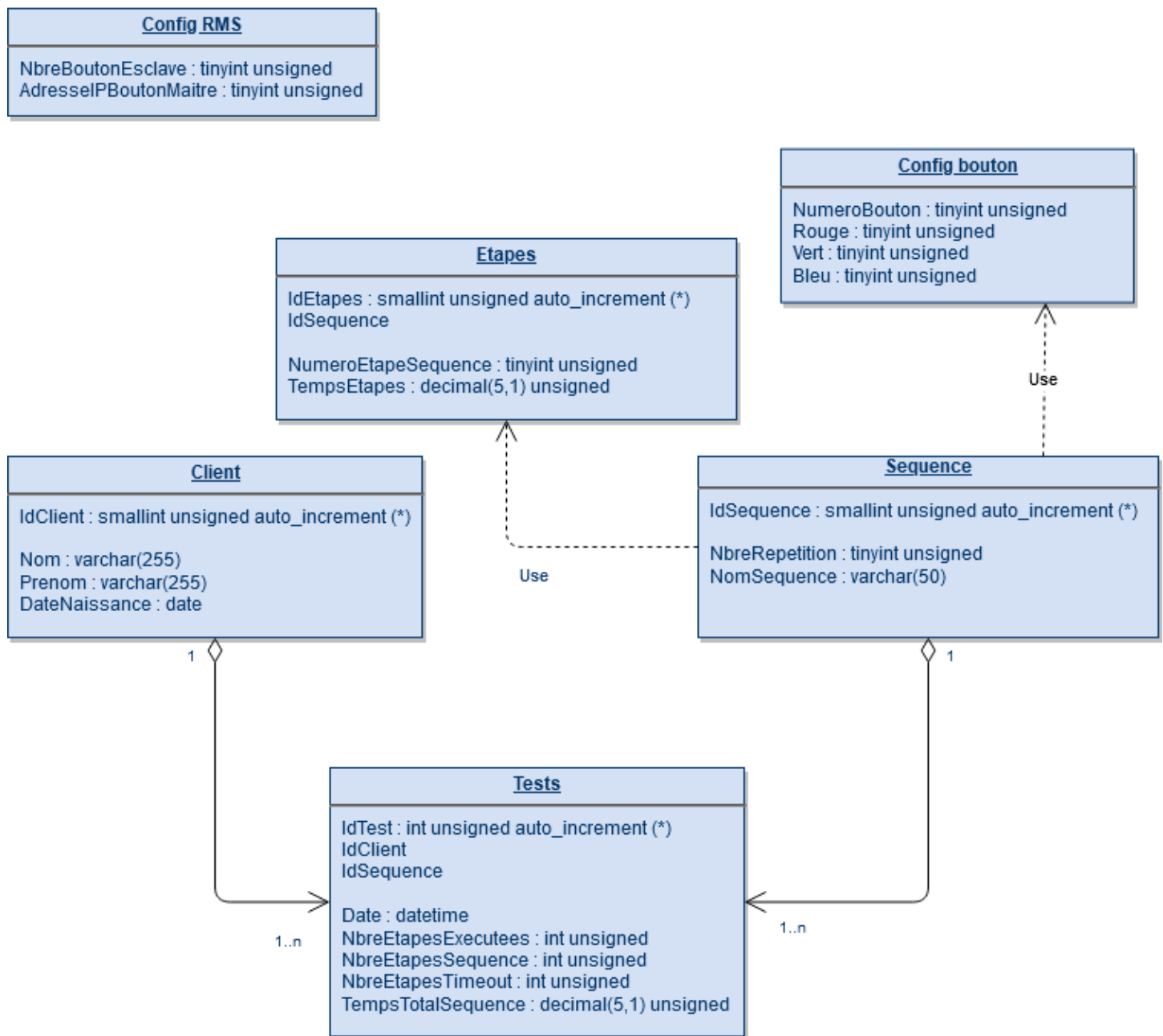
```
CREATE DATABASE rms CHARACTER set 'utf8';
USE rms;
CREATE TABLE client ( IdClient smallint unsigned auto_increment, Nom varchar(255) not null, Prenom
varchar(255) not null, DateNaissance date not null, PRIMARY KEY (IdClient) ) ENGINE=INNODB;
CREATE TABLE config_bouton ( NumeroBouton tinyint unsigned, Rouge tinyint unsigned, Vert tinyint
unsigned, Bleu tinyint unsigned ) ENGINE=INNODB;
CREATE TABLE sequence ( IdSequence smallint unsigned auto_increment, NbreRepetition tinyint unsigned,
NomSequence varchar(50) not null, PRIMARY KEY (IdSequence) ) ENGINE=INNODB;
CREATE TABLE tests ( IdTest int unsigned auto_increment, IdClient smallint unsigned, IdSequence smallint
unsigned, NbreEtapesSequence int unsigned, NbreEtapesTimeout int unsigned, NbreEtapesExecutees int
unsigned, TempsTotalSequence decimal(5,1) unsigned, Date datetime not null, PRIMARY KEY (IdTest) )
ENGINE=INNODB;
CREATE TABLE etapes ( IdEtapes smallint unsigned auto_increment, IdSequence smallint unsigned,
NumeroEtapeSequence tinyint unsigned, TempsEtapes decimal(5,1) unsigned, PRIMARY KEY (IdEtapes) )
ENGINE=INNODB;
CREATE TABLE config_rms ( NbreBoutonEsclave tinyint unsigned, AdresseIPBoutonMaitre tinyint unsigned )
ENGINE=INNODB;

ALTER TABLE tests ADD CONSTRAINT relationIdClient FOREIGN KEY (IdClient) REFERENCES
client(IdClient);
ALTER TABLE tests ADD CONSTRAINT relationIdSeqTests FOREIGN KEY (IdSequence) REFERENCES
sequence(IdSequence);
ALTER TABLE etapes ADD CONSTRAINT relationIdSeqEtapes FOREIGN KEY (IdSequence) REFERENCES
sequence(IdSequence);
```

J'ai changé le schéma de ma base de données : création d'une table contenant toutes les définitions de toutes les séquences dans une table. Il suffira d'extraire les étapes d'une séquence par une recherche du genre "me trouver toutes les étapes de telle séquence".

# Suivi Projet

## RMS (rééducation membres supérieurs)



Jeudi 18 et Samedi 20 juin : Recherche de la méthode qui me permet d'afficher une fenêtre pop-up lorsque j'appuie sur un bouton. Application de la méthode sur le bouton "fin programme" et sur le "X" de mon application. Code :

```
void CRMSDIg::OnClose()
{
    // TODO: ajoutez ici le code de votre gestionnaire de messages et/ou les paramètres par défaut des appels
    int MsgBoxError = MessageBoxW(L"Etes vous sur de vouloir quitter l'application?", L"Reeducation Membres Superieurs", MB_YESNO | MB_ICONQUESTION);

    if (MsgBoxError == IDYES)
        CDialogEx::OnClose();
}
```

```
void CMenuPrincipal::OnBnClickedButtonFinProgramme()
{
    // TODO: ajoutez ici le code de votre gestionnaire de notification de contrôle
    int MsgBoxError = MessageBoxW(L"Etes vous sur de vouloir quitter l'application?", L"Reeducation Membres Superieurs", MB_YESNO | MB_ICONQUESTION);

    if (MsgBoxError == IDYES)
        PostQuitMessage(0);
}
```

Déplacement des lignes de macro de définition des couleurs "RGB()" utilisées par les noms des couleurs déclarées dans un .h déjà existant ce qui me permet d'économiser un fichier dans mon programme.

Début d'implémentation de mon projet de connexion et création de base de donnée MYSQL dans mon projet d'IHM, en transférant les .cpp et les .h d'un projet à l'autre et en y faisant les liens aux bibliothèques nécessaires dans les paramètres du projet. Réussite de l'implémentation de la partie connexion à la base de donnée dans mon projet IHM, tout en affichant un message pop d'erreur si une erreur à la connexion d'une base se produit.

Finalement toute la journée passée à ce problème de pop-up de connexion à la base de données au lancement du programme, car problème de conversion LPCTSTR.