

Rapport de projet

Auteurs:

Yanis BOUSSNENE

Nathan DE ALBA

Yann PILLON

Présentation Générale

Equipe :

Etudiant 1 : Yanis BOUSSNENE

Etudiant 2 : Nathan DE ALBA

Etudiant 3 : Yann PILLON

Client :

Lycée Jean Rostand

boulevard 93420, 243 Boulevard Robert Ballanger

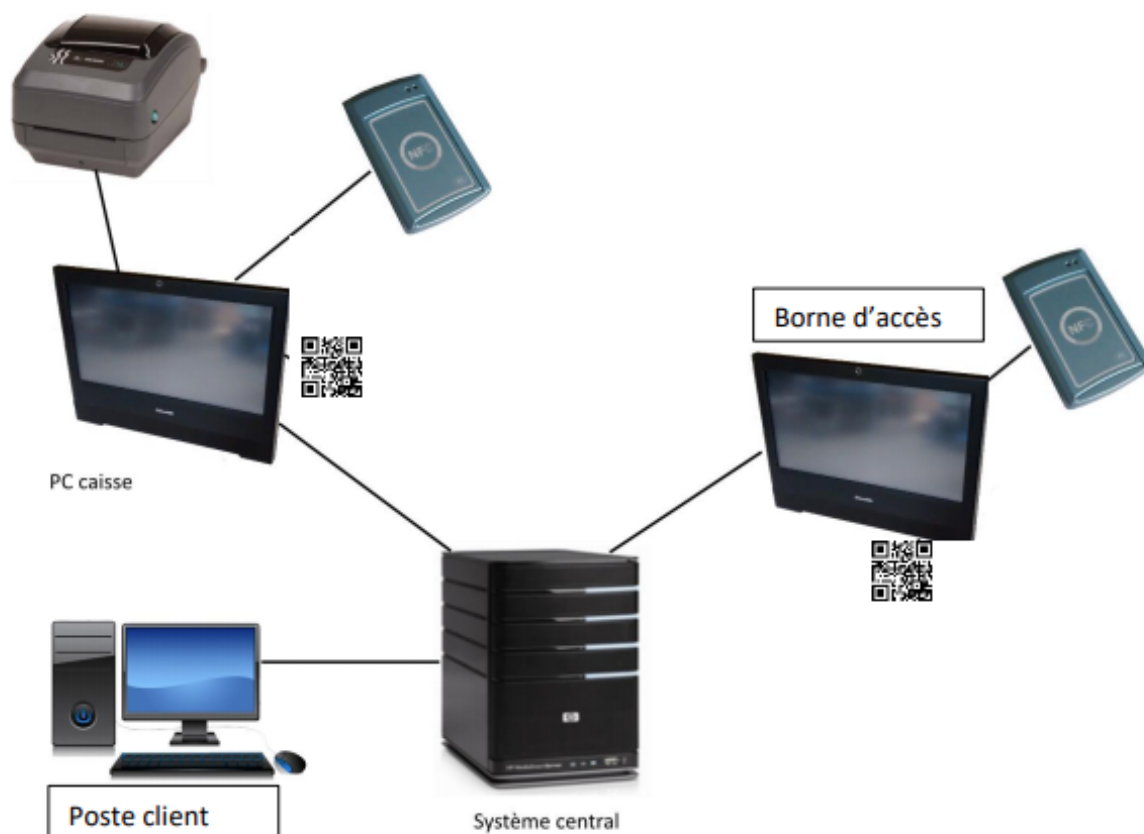
93420 Villepinte

Objectif du projet

Ce projet a pour but de pouvoir gérer les entrées au stade de football « SnirArea ». Il permettra la gestion des places pour des clients non-abonnés et des abonnés. Ceux-ci se verront remettre un badge RFID leur autorisant l'accès au stade. Pour les clients non abonnés un code QR-code leur sera fourni lors du paiement.

Cahier des charges – Expression du besoin

Synoptique de l'architecture :



Le client souhaite :

Un site internet permettant l'accès au paiement en ligne avec choix des places pour les clients abonnés ou non-abonnés.

Les clients abonnés bénéficieront d'une réduction de 30% après paiement d'un abonnement de 100€ pour tous les matchs de la saison. Ils auront une carte RFID pour accéder directement au match qu'ils auront payé.

Pour les clients non abonnés, ils recevront après paiement un QR-code leur permettant d'accéder au stade.

Répartitions des tâches

Etudiant 1 :

Il devra coder le site web permettant

- à un abonné de pouvoir :
 - Sélectionner et réserver sa place via un site web.
 - Payer sa place avec un mail de confirmation.
 - Accéder directement à la borne d'accès au stade via son badge RFID mis à jour.
 - A un non-abonné de pouvoir :
 - Sélectionner et réserver sa place via un site web.
 - Payer sa place avec un mail de confirmation + QR-code.
 - Accéder directement à la borne d'accès au stade via son QR-code.
- Il devra conceptualiser et modéliser la base de données. Fournir les requêtes SQL permettant le fonctionnement de la caisse, du site web et de la borne d'accès.

Etudiant 2 :

Il devra développer le logiciel de caisse qui permettra de :

- Vendre des places à des clients non-abonnés :
 - Sélectionner une place.
 - Procéder au paiement de la place.

Imprimer un ticket récapitulatif avec le montant de la transaction, le numéro de la place et un QR-code.

- Vendre des places à des clients abonnés.

Lire le badge client ou rechercher son nom etc...

Sélectionner la place

Procéder au paiement de la place et mettre à jour la base de données pour activer le passage.

Imprimer un ticket récapitulatif avec le montant de la transaction, le numéro de la place.

Etudiant 3 :

Il devra développer le logiciel de la borne d'accès qui permettra de :

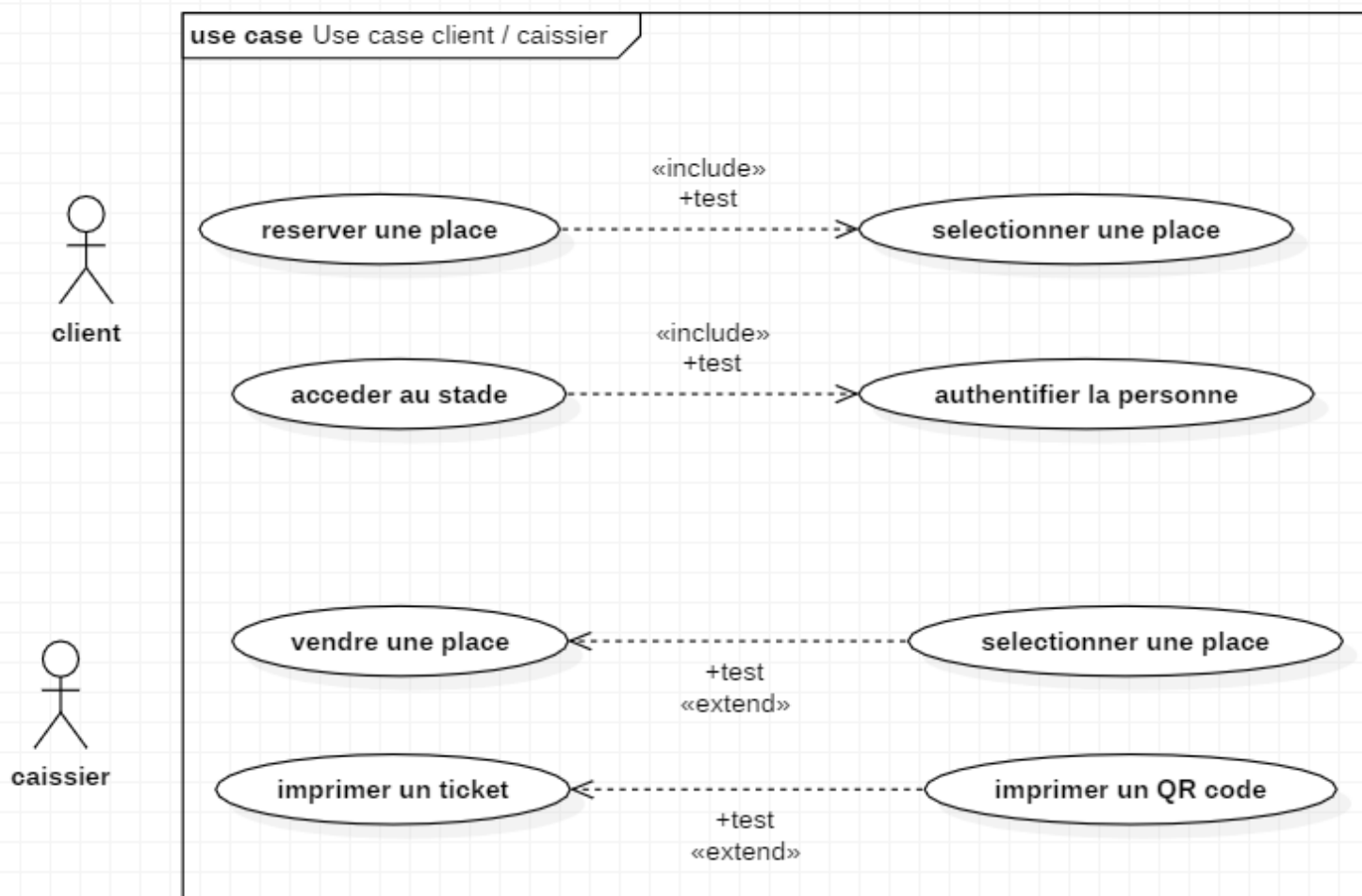
Lire les QR-codes des clients non-abonnés et signaler via une carte Velleman l'autorisation de passer.

Lire les badges RFID des clients abonnés et signaler via une carte Velleman l'autorisation de passer.

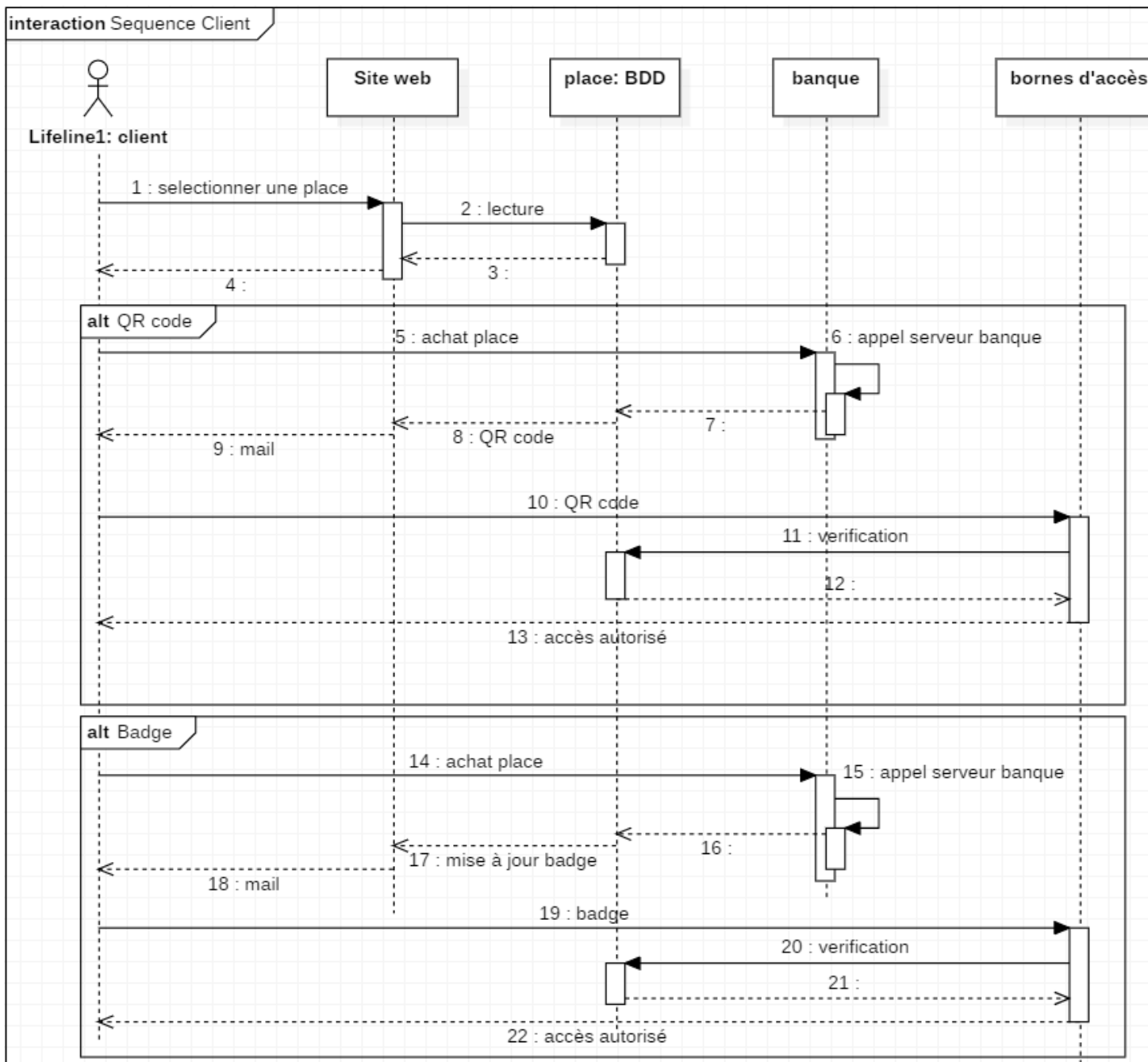
	Fonctions à développer et tâches à effectuer	
Étudiant 1 EC <input type="checkbox"/> IR <input checked="" type="checkbox"/>	Liste des fonctions assurées par l'étudiant Création du site Web en http Modélisation et conceptualisation de la BDD Tester et fournir les requêtes SQL	Installation : Serveur Web, OS , IDE, OS, SGBD MySQL, client MySQL Mise en œuvre : serveur web, MySQL, MySQLWorkbench. Configuration : serveur web Réalisation : Site web, gestion de la BDD par le site web en PHP, mail avec ticket QR code si non abonné ou mise à jour carte RFID abonné. : MCD MLD, requêtes SQL du système snirArea Documentation : Déploiement du serveur Web, déploiement du SGBD et configuration Utilisation client du site web
Étudiant 2 EC <input type="checkbox"/> IR <input checked="" type="checkbox"/>	Liste des fonctions assurées par l'étudiant Logiciel pour le poste caisse Lecture des cartes RFID Impression des tickets et des codes QRcodes Gestion d'accès à la base de données	Installation : lecteur RFID, OS et IDE Mise en œuvre : Interface graphique Configuration : Lecteur RFID Réalisation : Impression d'un ticket avec QRcode, lecture de la carte RFID, IHM du poste du caissier Documentation : Manuel d'utilisation du logiciel de caisse
Étudiant 3 EC <input type="checkbox"/> IR <input checked="" type="checkbox"/>	Liste des fonctions assurées par l'étudiant Logiciel pour la borne d'accès Gestion d'accès à la base de données Lecture des codes QRcode par caméra Lecture des RFID	Installation : OS et IDE et Installation camera QRcode Mise en œuvre : Interface graphique Configuration : lecteur QRcode (caméra) Réalisation : Module de gestion de la base de données en C++, Logiciel IHM pour la borne de réservation Documentation : Manuel d'utilisation du logiciel borne d'accès

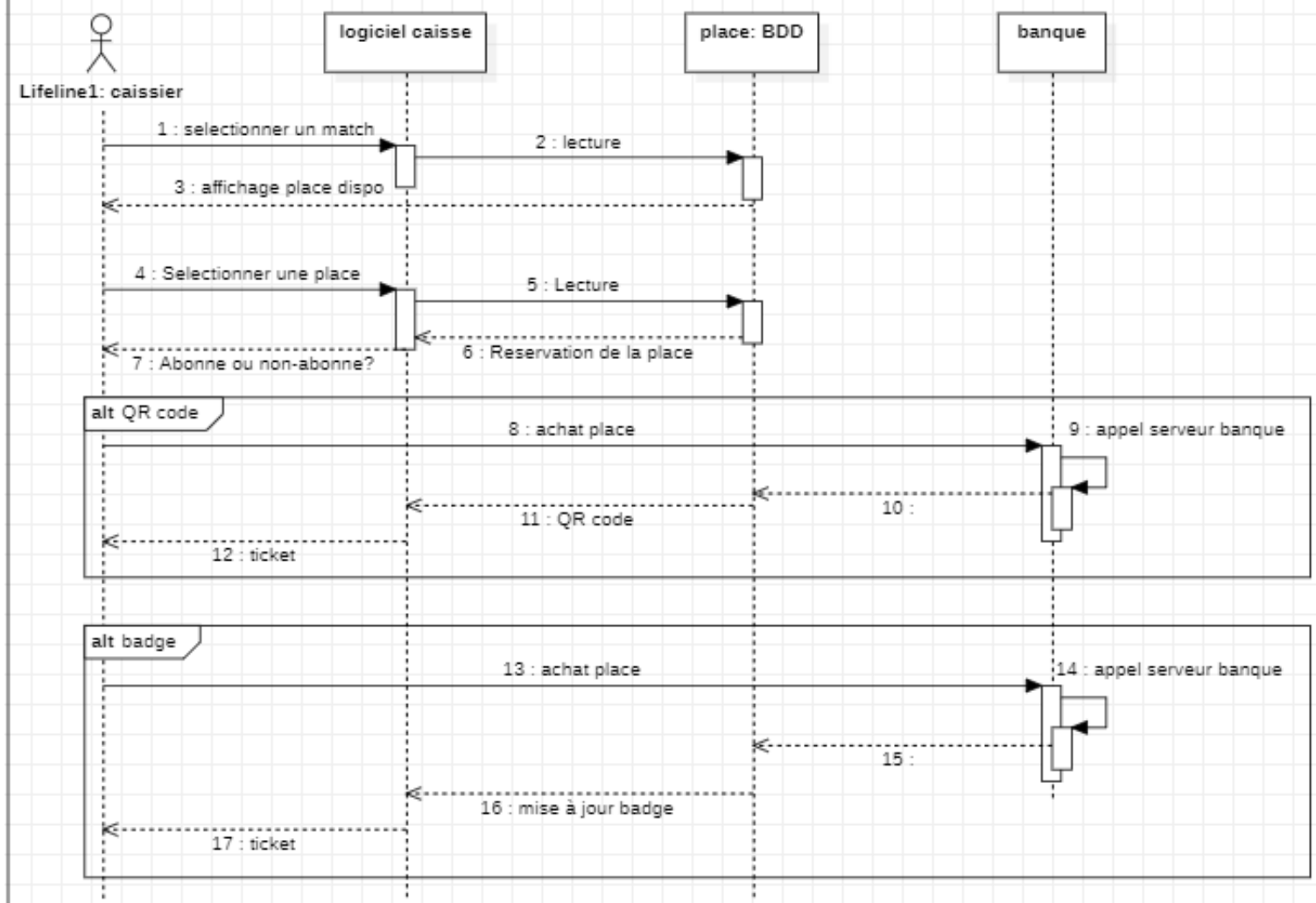
Diagrammes UML

Diagramme d'utilisation:



Diagrammes de séquence:





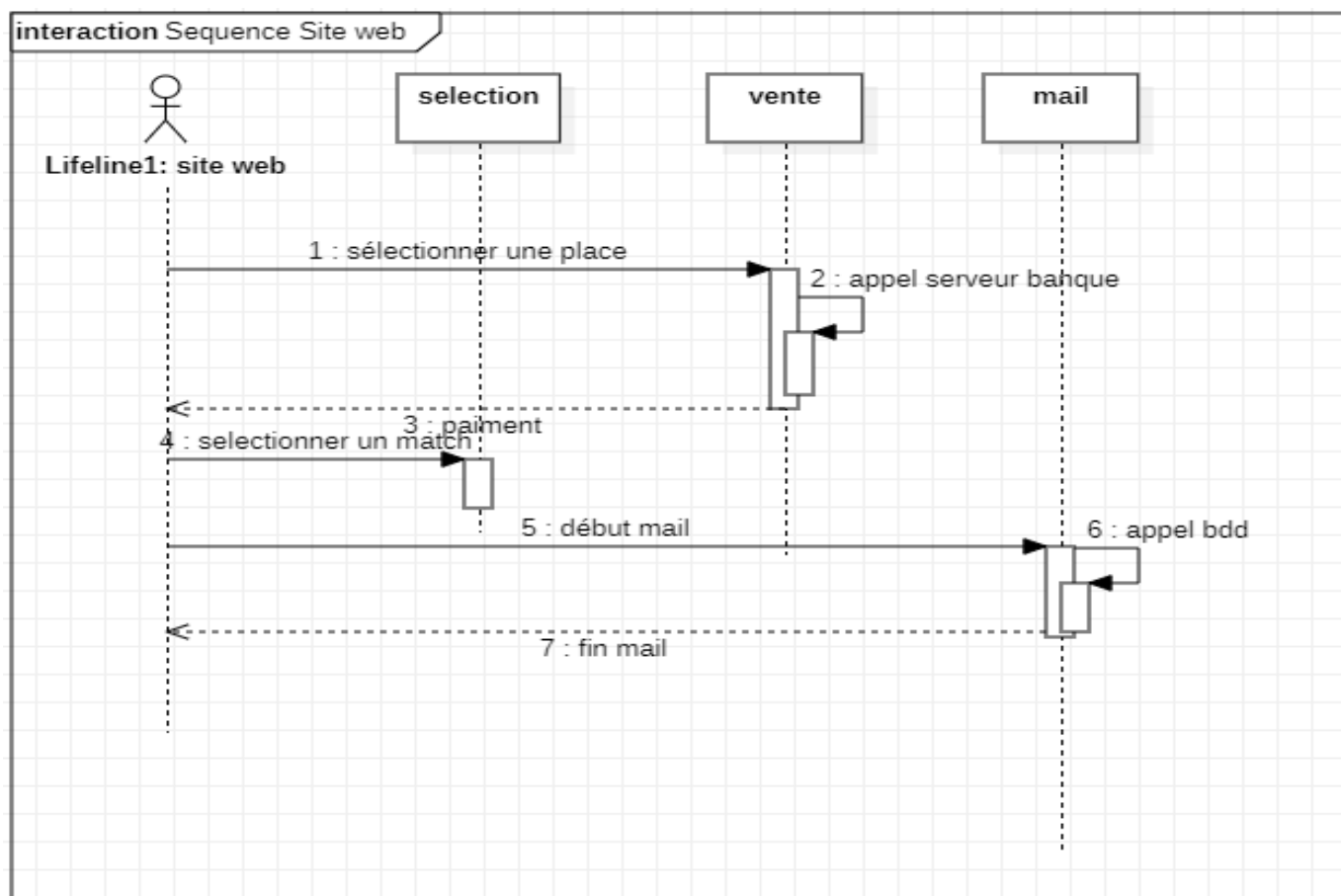
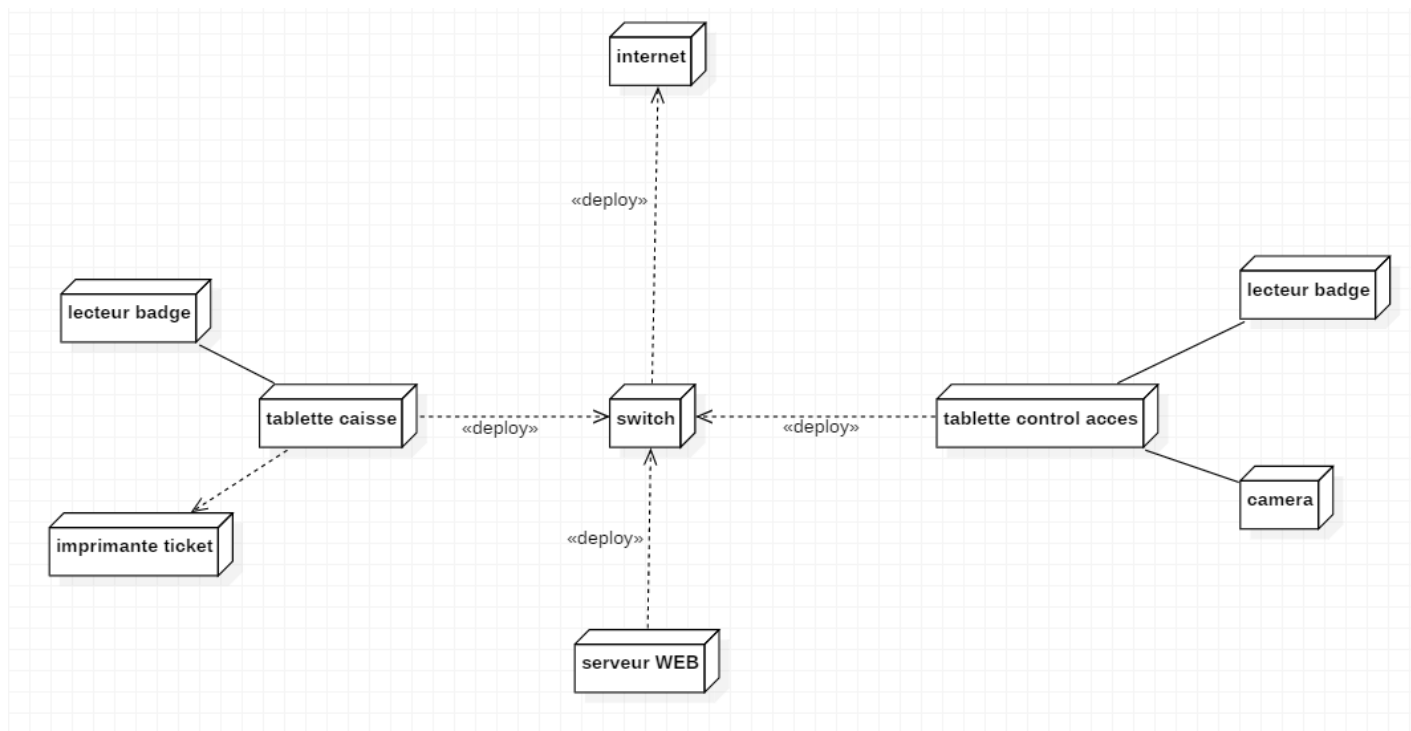


Diagramme de déploiement:



PARTIE CANDIDAT 1

Yanis BOUSSNENE

1 - Introduction

Base de données et liaison avec elle

Le premier candidat devra réaliser la conception de la base de données ainsi que le site web.

1.1 Base de données

La base de données a été modélisée grâce au logiciel StarUML puis Workbench et devra contenir les informations suivantes :

Utilisateurs

Les différents utilisateurs seront les suivant :

- Client Abonné (carte RFID)
- Client Non - Abonné (code QR)
- Administrateur

On devra pouvoir stocker et récupérer les informations suivantes pour tous ces utilisateurs :

- Nom
- Prénom
- ID numéro de compte
- Abonné ou Non - Abonné
- Mot de passe

Etablissement

Nous avons choisi de prendre le Parc des Princes comme stade pour notre projet. On devra pouvoir stocker et récupérer les informations suivantes pour ce stade :

- Adresse
- Nombre de place
- Lieu sur google maps

Places

Les places disponibles devront être enregistrées afin de savoir combien restera-t-il de place disponible dans le stade. Ainsi que la position et l'emplacement de la place dans le stade afin de varier le prix de celle-ci.

- Position
- Tarif
- Stade
- Numéro de place

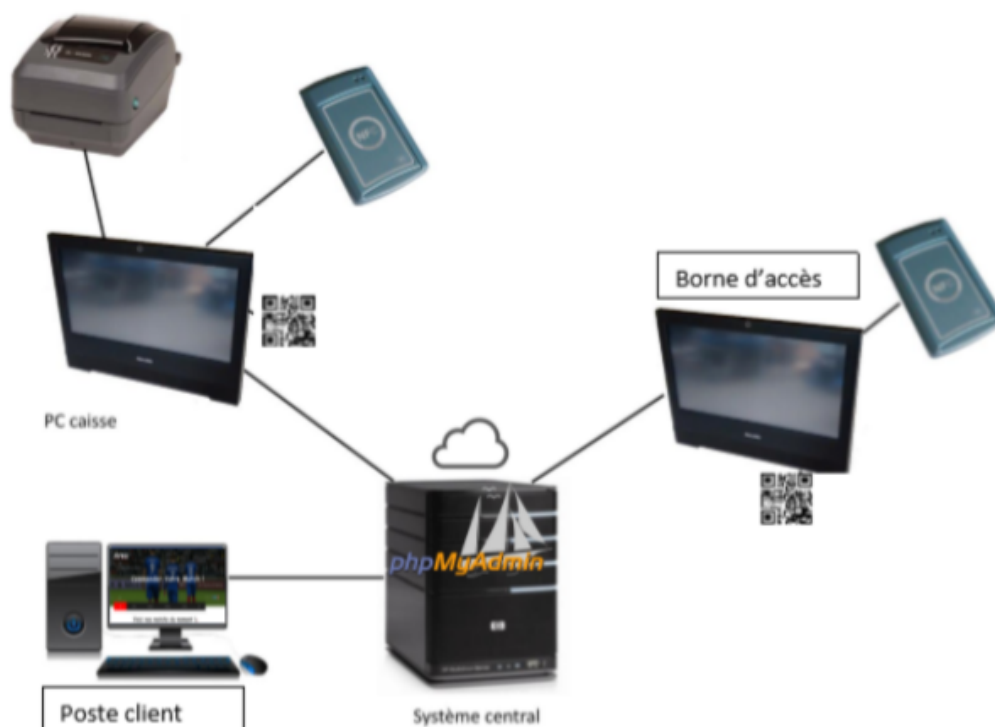
QR code / carte RFID

Elle devra contenir les informations du client si il est abonné ou non-abonné

- ID carte RFID (abonné)
- ID code QR (non-abonné)

1.2 Liaison avec le PC caisse

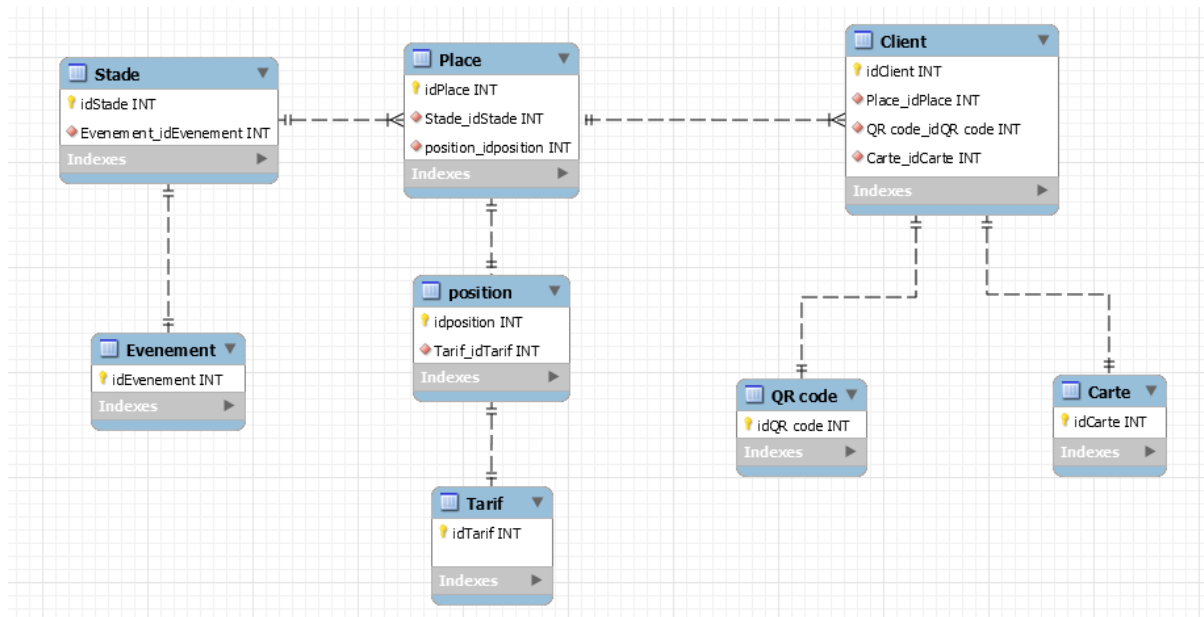
Le PC caisse sera relié avec quasiment les mêmes requêtes que le site web. En effet, ils iront chercher les mêmes informations dans la base de données. Le PC caisse est disponible sur place et il faut qu'il puisse savoir par exemple si la place n°7 est réservée ou non. Ensuite si le client est abonné sa carte RFID sera mise à jour, si il n'est pas abonné un code QR sera imprimé pour que la borne d'accès puisse scanner le code QR et ainsi permettre l'accès au stade.



2 - Structure de la base de données

La base de données est composée de 8 tables reliées par des associations qui permettent d'empêcher la redondance.

Toutes les tables ont une clé primaire qui permet l'unicité d'un enregistrement, certaines d'entre possèdent des clés étrangères qui permettent la liaison entre les tables.




```
--  
-- Base de données : `bddarea`  
--  
  
-----  
  
--  
-- Structure de la table `badge`  
--  
  
DROP TABLE IF EXISTS `badge`;  
CREATE TABLE IF NOT EXISTS `badge` (  
  `idBadge` int(11) NOT NULL AUTO_INCREMENT,  
  PRIMARY KEY (`idBadge`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;  
  
--  
-- Déchargement des données de la table `badge`  
--  
  
INSERT INTO `badge` (`idBadge`) VALUES  
(1),  
(2),  
(4);
```

```
-- Structure de la table `client`
--

DROP TABLE IF EXISTS `client`;
) CREATE TABLE IF NOT EXISTS `client` (
  `idClient` int(11) NOT NULL AUTO_INCREMENT,
  `Badge_idBadge` int(11) NOT NULL,
  PRIMARY KEY (`idClient`)
~ ) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

--
-- Déchargement des données de la table `client`
--

INSERT INTO `client` (`idClient`, `Badge_idBadge`) VALUES
(1, 1),
(2, 2),
(4, 4);

-----

--
-- Structure de la table `membres`
--

DROP TABLE IF EXISTS `membres`;
) CREATE TABLE IF NOT EXISTS `membres` (
  `id` int(11) NOT NULL,
  `pseudo` varchar(255) NOT NULL,
  `pass` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `date_inscription` date NOT NULL,
  PRIMARY KEY (`id`)
~ ) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-----
```

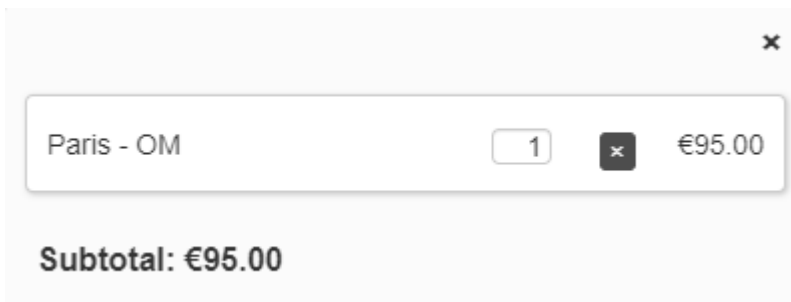
3 - Site Web et serveur PHP

Pour pouvoir réserver une place, j'ai imaginé ce site web qui est simple et ludique d'utilisation :



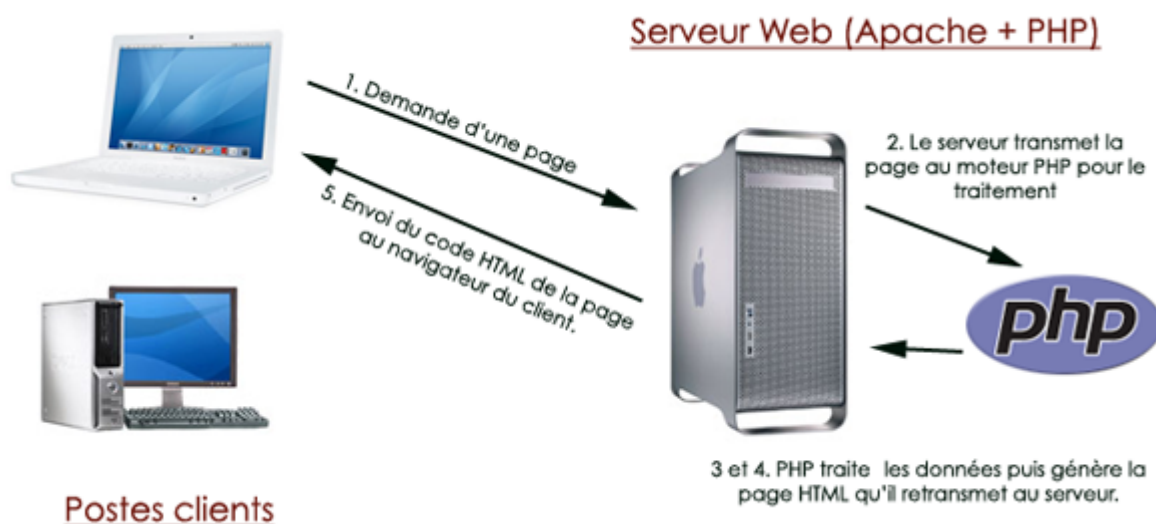
Voici nos matchs du moment ! 🇫🇷





J'ai pu modéliser ce site à l'aide du logiciel Bootstrap et aussi des connaissances que j'ai acquises au fur et à mesure de ma formation en BTS. Les images ci-dessus sont un aperçu du système e-commerce mis en place mais pas encore terminé.

Il manque plus que relier le tout à la base de données qui est actuellement en cours grâce au PHP.



```
<!DOCTYPE html>
<html>
  <head>
    <title>Test connexion</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>
  <body>
    <h1>Bases de données MySQL</h1>
    <?php
      $servername = 'localhost';
      $username = 'root';
      $password = '';

      //On établit la connexion
      $conn = new PDO("mysql:host=$servername;dbname=bddarea", $username, $password);
    ?>
  </body>
</html>
```

Le code ci-dessus permet de relier notre site web afin d'établir une connexion avec la base de données "bddarea".

Choisir votre place !

Merci de choisir le nombre de place :

Place n°1 ▼

Choisir votre position !

Merci de choisir la position de la place :

Bas ▼

Ajouter Au Panier



Le code ci-dessous est un aperçu du code HTML du site qui sera par la suite relier avec du PHP à la base de données afin de fixer un prix en fonction de la position et du nombre de places sélectionnées.

```
114 <div class="features">
115   <div class="container">
116     <div class="heading-section-w3ls">
117       <h2 class="title-w3layouts">Voici nos matchs du moment !<i class="fa fa-bell-o" aria-hidden="true"></i><i class="fa fa-bell" aria-hidden="true"></i></h2>
118     </div>
119     <div class="col-md-4 servies-agileinfo">
120       <p class="pink-w3ls">Choisissez votre place !</p>
121       <p class="para-w3agileits">Merci de choisir le nombre de place :</p>
122       <form>
123         <SELECT name="nom"size="1">
124           <OPTION> Place n°1
125           <OPTION> Place n°2
126           <OPTION> Place n°3
127         </SELECT>
128       </form>
129       <p class="pink-w3ls">Choisissez votre position !</p>
130       <p class="para-w3agileits">Merci de choisir la position de la place :</p>
131       <form>
132         <SELECT name="nom"size="1">
133           <OPTION> Bas
134           <OPTION> Milieu
135           <OPTION> Hauteur
136         </SELECT>
137       </form>
138       <a href="#" class="botton-agileits">Ajouter au panier</a>
139     </div>
140     <div class="clearfix"></div>
141   </div>
142 </div>
143 </div>
144 </div>
```

4 - Conclusion

Actuellement, ma partie concernant la création et la gestion de la base de données répond au besoin du projet et est terminée dans le délai qui était imposé. En effet, seule la partie sur le paiement de la place et la liaison avec la base de données reste encore inachevée. Quelques détails restent encore à régler, concernant notamment le système du panier du site, la création du code QR.

Je pense avoir répondu au mieux aux attentes et aux besoins du client, et j'espère satisfaire le client de mon travail.

Ce projet a été très intéressant malgré certaines contraintes. Grâce à ce projet j'ai un aperçu de mon travail dans l'avenir.

Etudiant 2 : De Alba Nathan

1. Présentation

Le premier étudiant aura la responsabilité de réaliser le logiciel de caisse du stade.

Il devra réaliser le logiciel de caisse, comprenant :

- a. Une connexion du caissier
- b. Une sélection de l'événement
- c. Une sélection des places
- d. La lecture d'une carte RFID
- e. L'impression d'un ticket, contenant un QR-Code pour accéder à la salle

2. Cahier des charges

- a. Vendre des places à des clients non-abonnés :
 1. Le client peut sélectionner une place.
 2. Procéder au paiement de la place.
 3. Imprimer un ticket récapitulatif avec le montant de la transaction, le numéro de la place et un QR-code.
- b. Vendre des places à des clients abonnés.
 1. Lire le badge client
 2. Le client peut sélectionner une place.
 3. Procéder au paiement de la place et mettre à jour la base de données pour activer le passage.
 4. Imprimer un ticket récapitulatif avec le montant de la transaction, le numéro de la place.

3. Outils utilisé

Qt Creator



Qt Creator va me permettre de coder l'application de caisse

PhpMyAdmin

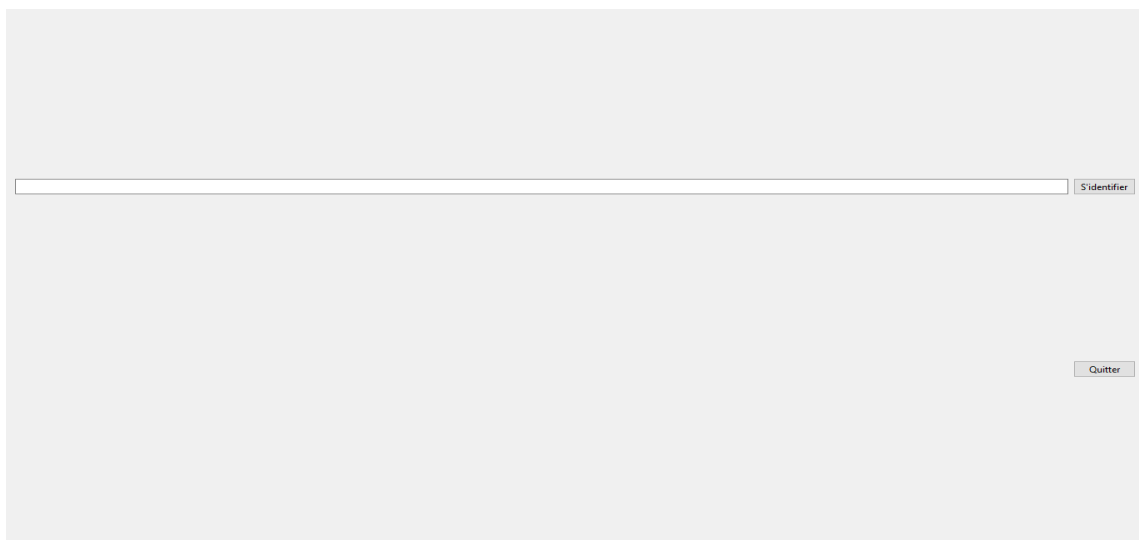


phpMyAdmin va me permettre de gérer la bdd.

4. Application

a. Connexion du caissier

Le caissier rentre son identifiant afin de s'identifier.

A simple login form with a light gray background. It features a single text input field for the username. To the right of the input field is a button labeled 'S'identifier'. Below the input field, on the right side, is a button labeled 'Quitter'.

Code de la page caissier

```
#include "caisse.h"
#include "ui_caisse.h"
#include <QPushButton>
#include "match.h"
#include <QGridLayout>
#include <QLineEdit>
Caisse::Caisse(QWidget *parent)
    : QWidget(parent)
    , ui(new Ui::Caisse)
{
    ui->setupUi(this);
    Identifiant = new QLineEdit();
    PageMatch = new QPushButton("S'identifier", this);
    Quitter = new QPushButton("Quitter", this);
    QObject::connect(PageMatch, SIGNAL(clicked()), this, SLOT(AfficherPageMatch()));

    QObject::connect(Quitter, SIGNAL(clicked()), this, SLOT(close()));

    QGridLayout *mainLayout = new QGridLayout;
    mainLayout->addWidget(Identifiant,0,1);
    mainLayout->addWidget(PageMatch,0,2);
    mainLayout->addWidget(Quitter,80,2);

    setLayout(mainLayout);
}

void Caisse::AfficherPageMatch()
{
    Match * Fenetre = new Match ;
    Fenetre->showFullScreen();
    close();
}
```

J'utilise des QObject afin de lier mes fermetures ou ouvertures de pages a la pression du bouton.

```
QGridLayout *mainLayout = new QGridLayout;
mainLayout->addWidget(Identifiant,0,1);
mainLayout->addWidget(PageMatch,0,2);
mainLayout->addWidget(Quitter,80,2);

setLayout(mainLayout);
```

Création d'un layout afin de placer les éléments sur notre page

b. Sélection des événements

Page de sélection

foot	2021-10-04T15:00:00.000	<input type="checkbox"/> événement 1
concert	2021-05-10T13:00:00.000	
marathon	2022-07-25T18:00:00.000	

Selectionnez match

Quitter

```
QSqlDatabase db = QSqlDatabase::addDatabase("QODBC");
db.setHostName("localhost");
db.setPort(3306);
db.setDatabaseName("Driver={MySQL ODBC 5.3 ANSI Driver};DATABASE=bddarea;");
db.setUserName("root");
db.setPassword("");
if(db.open())
{
    qDebug() << "Vous êtes maintenant connecté à " << db.hostName() ;

    QSqlQuery query;

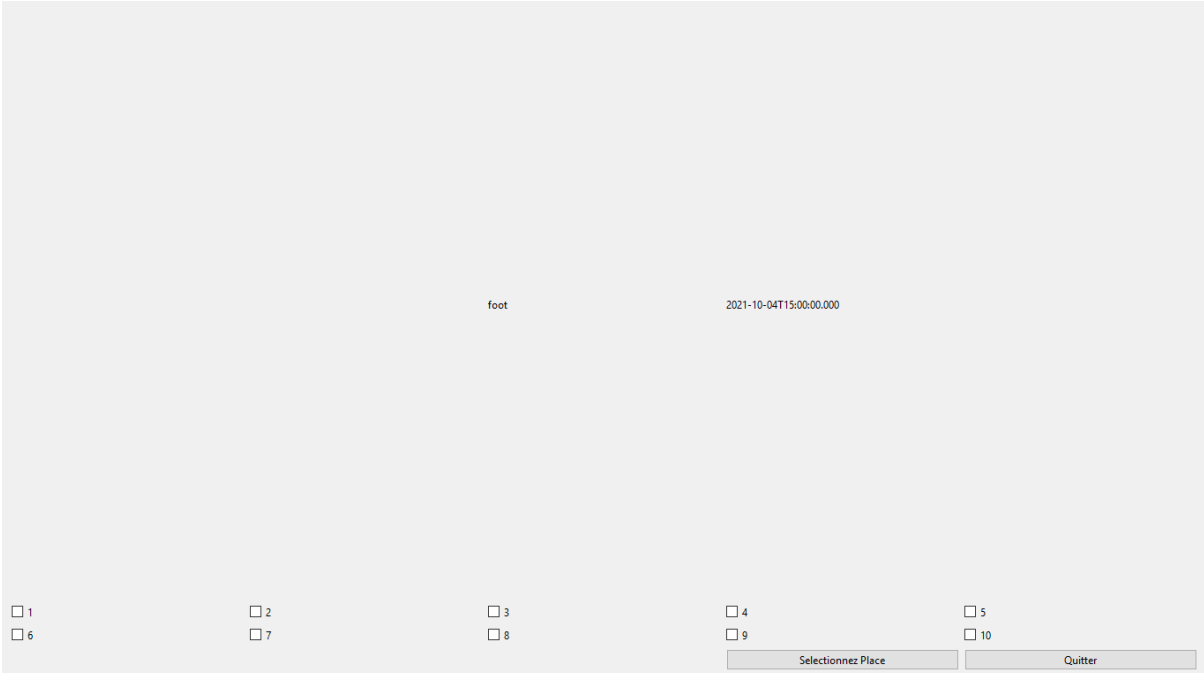
    query.exec("SELECT * FROM `evenement` ");
    query.next();
    {
        QString evenement = query.value(1).toString();
        QString date = query.value(2).toString();

        type->setText(evenement);
        jour->setText(date);
    }
}
```

Connexion à la bdd et récupérations des informations sur les événement

c. Sélection des places.

Page d'affichage des places



The screenshot shows a Qt application window with the title bar 'foot'. The window has a light gray background. In the center, there is a large empty space. At the bottom of the window, there is a grid of 10 checkboxes arranged in two rows of five. The checkboxes are labeled 1 through 10. Below the grid, there are two buttons: 'Selectionnez Place' and 'Quitter'. The window also displays the text '2021-10-04T15:00:00.000' in the top right corner.

Lecture de la bdd afin de vérifier les places disponibles

```
query.exec("SELECT * FROM `places` ");
query.next();
{

    QString id = query.value(0).toString();
    QString status = query.value(3).toString();

    if(status=='1' || status=='2')
    {
        place1->setChecked(1);
        place1->setEnabled(0);
    }
    else
    {
        place1->setChecked(0);
    }
}
```

Les places réservées sont cocher et non cliquable afin d'éviter les confusions

Mise à jour de la bdd afin de pré-réserver la place avant paiement

```
query.exec("SELECT * FROM `places`");
query.next();
{
    QString reserver = query.value(3).toString();
    if(place1->isChecked() && reserver == '0')
    {
        query.exec("UPDATE `places` SET `Occupation` = 2 WHERE `idPlaces` = 1");
        qDebug() << "ca marche";
    }
}
```

d. Vérification de la sélection et affichage du prix

foot 2021-10-04T15:00:00.000

La place 1 = 10eu

La place 9 = 10eu

La place 3 = 20eu

La place 7 = 10eu

50

Commande Verifiez

Quitter

Page d'affichage du prix

```
query.exec("SELECT * FROM bddarea.places");
query.next();
{
    QString position = query.value(1).toString();
    QString reserver = query.value(3).toString();

    if(reserver == "2")
    {
        if(position == "1")
        {
            placel->setText("La place 1 = 10eu");
            prix = prix + 10;
        }
        else if (position == "2")
        {
            placel->setText("La place 1 = 15eu");
            prix = prix + 15;
        }
        else
        {
            placel->setText("La place 1 = 20eu");
            prix = prix + 20;
        }
    }
}
```

Code vérifiant si la place a été sélectionné ainsi que son prix en fonction de l'événement

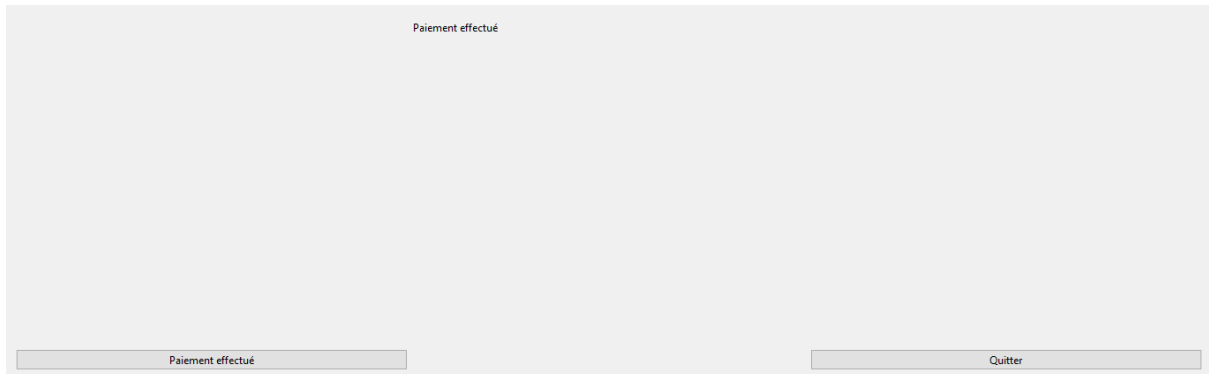
```
QSqlQuery query;

query.exec("SELECT * FROM `places`");
query.next();
{
    query.exec("UPDATE `places` SET `Occupation` = 0 WHERE `Occupation` = 2");
    qDebug() << "ca marche";
}
}
```

A cette étape si le client décide de changer de places, en quittant la page, les places pré-sélectionnées sont remises à leur état d'origine.

e. Paiement des places

Le paiement est simulé et les places sont réservées définitivement.



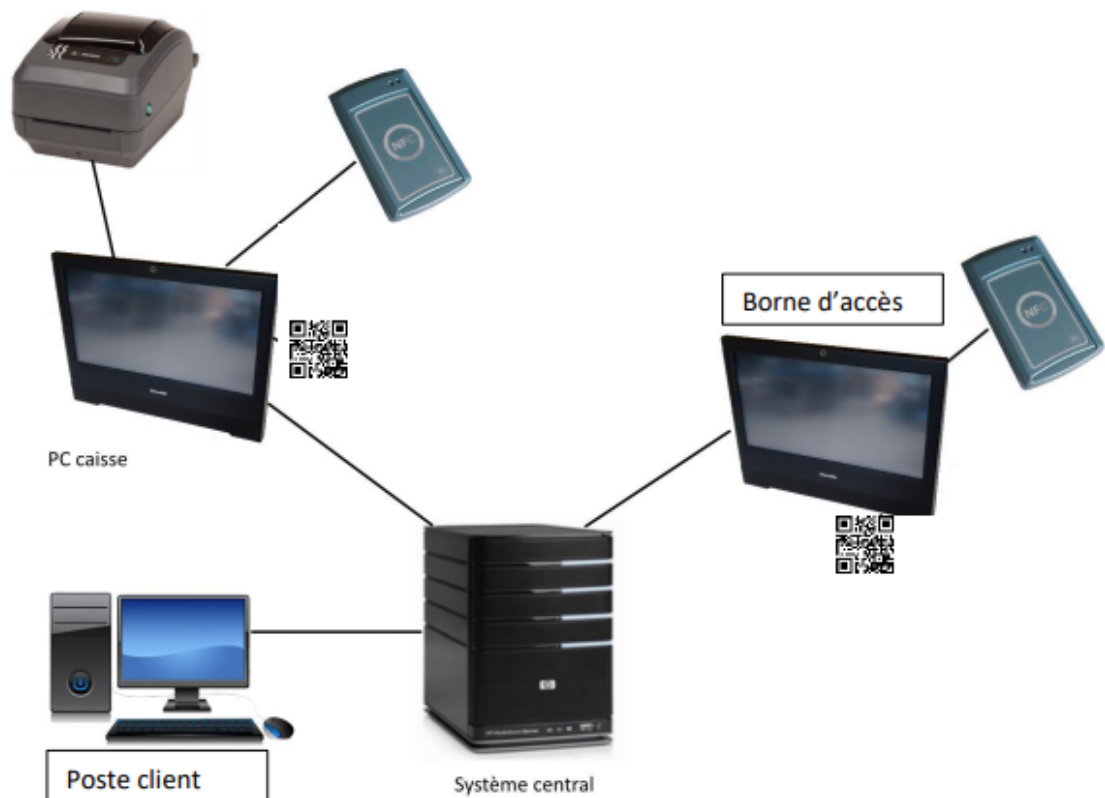
```
QSqlQuery query;  
  
query.exec("SELECT * FROM `places`");  
query.next();  
{  
query.exec("UPDATE `places` SET `Occupation`= 1 WHERE `Occupation` = 2");  
qDebug() << "ca marche";  
}
```

Etudiant 3 : PILLON YANN

Présentation du projet

L'objectif de ce projet c'est de pouvoir gérer les entrées au stade de football « SnirArea ». Pour ceux qui ont un badge RFID, l'accès au stade est autorisé. Pour les clients non abonnés un code QR-code leur sera fourni lors du paiement.

Synoptique de l'architecture :



Outils nécessaires



Code de la page

```
#include "projet.h"
#include "ui_projet.h"
#include <QLabel>
#include <QPushButton>
#include <QTextEdit>
#include <QHBoxLayout>
#include <QApplication>
#include <QLineEdit>
#include <QFormLayout>

Projet::Projet(QWidget *parent)
    : QWidget(parent)
    , ui(new Ui::Projet)
{
    ui->setupUi(this);
    QPushButton * BoutonFermer = new QPushButton(" fermer ",this) ;

    QPushButton * BoutonID = new QPushButton(" s'identifier ",this) ;

    QLabel * TexteNumAbo = new QLabel(" Numéro d'abonner: ",this) ;
    QLabel * TexteNomUtilisateur = new QLabel(" Nom d'utilisateur: ",this) ;
    QLabel * TexteMDP = new QLabel(" Mot de passe: ",this) ;
    QLineEdit* lineEdit = new QLineEdit(this);
    layout()->addWidget(lineEdit);
    QGridLayout *layout = new QGridLayout;

    layout->addWidget(TexteNumAbo,0,3);
    layout->addWidget(TexteNomUtilisateur,1,3);
    layout->addWidget(TexteMDP,2,3);
    layout->addWidget( BoutonID,4,0);
    layout->addWidget(BoutonFermer,4,3);
    this->setLayout(layout);

    // Connexion du clic du bouton
    QObject::connect(BoutonFermer, SIGNAL(clicked()), this, SLOT(close()));
}

Projet::~Projet()
{
    delete ui;
}
```


Manuel d'utilisation

```
QPushButton * BoutonFermer = new QPushButton(" fermer ",this) ;  
QPushButton * BoutonID = new QPushButton(" s'identifier ",this) ;
```

Cette partie sert à créer des boutons.

```
QLabel * TexteNumAbo = new QLabel(" Numéro d'abonner: ",this) ;  
QLabel * TexteNomUtilisateur = new QLabel(" Nom d'utilisateur: ",this) ;  
QLabel * TexteMDP = new QLabel(" Mot de passe: ",this) ;
```

Celle-ci permet au client de s'identifier.

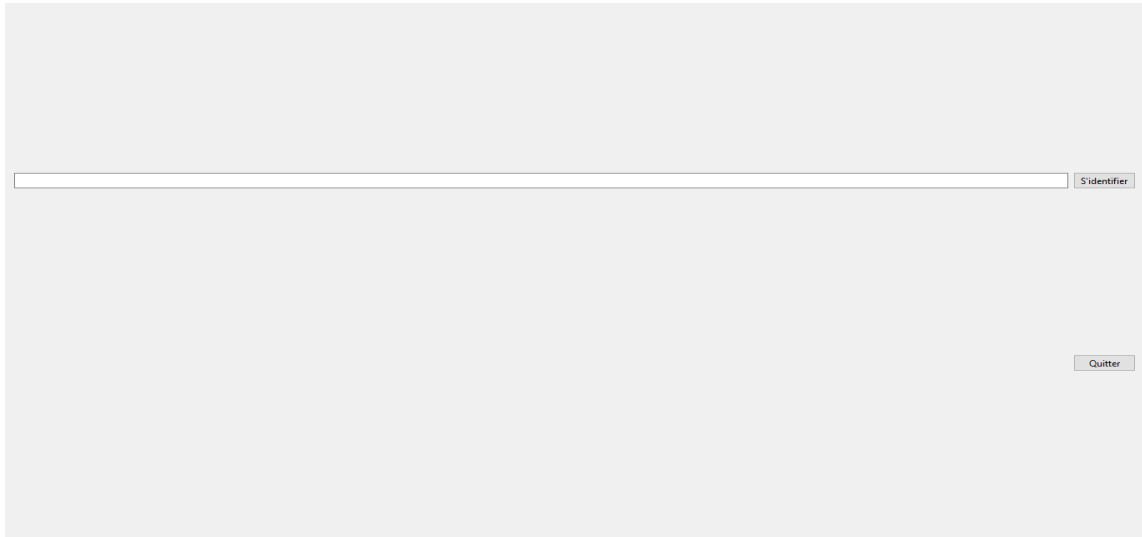
```
QLineEdit* lineEdit = new QLineEdit(this);  
layout()->addWidget(lineEdit);  
QGridLayout *layout = new QGridLayout;  
  
layout->addWidget(TexteNumAbo,0,3);  
layout->addWidget(TexteNomUtilisateur,1,3);  
layout->addWidget(TexteMDP,2,3);  
layout->addWidget( BoutonID,4,0);  
layout->addWidget(BoutonFermer,4,3);  
this->setLayout(layout);
```

Ici on positionne les différentes parties dans la fenêtre (gauche, droite, etc).

```
// Connexion du clic du bouton  
QObject::connect(BoutonFermer, SIGNAL(clicked()), this, SLOT(close()));
```

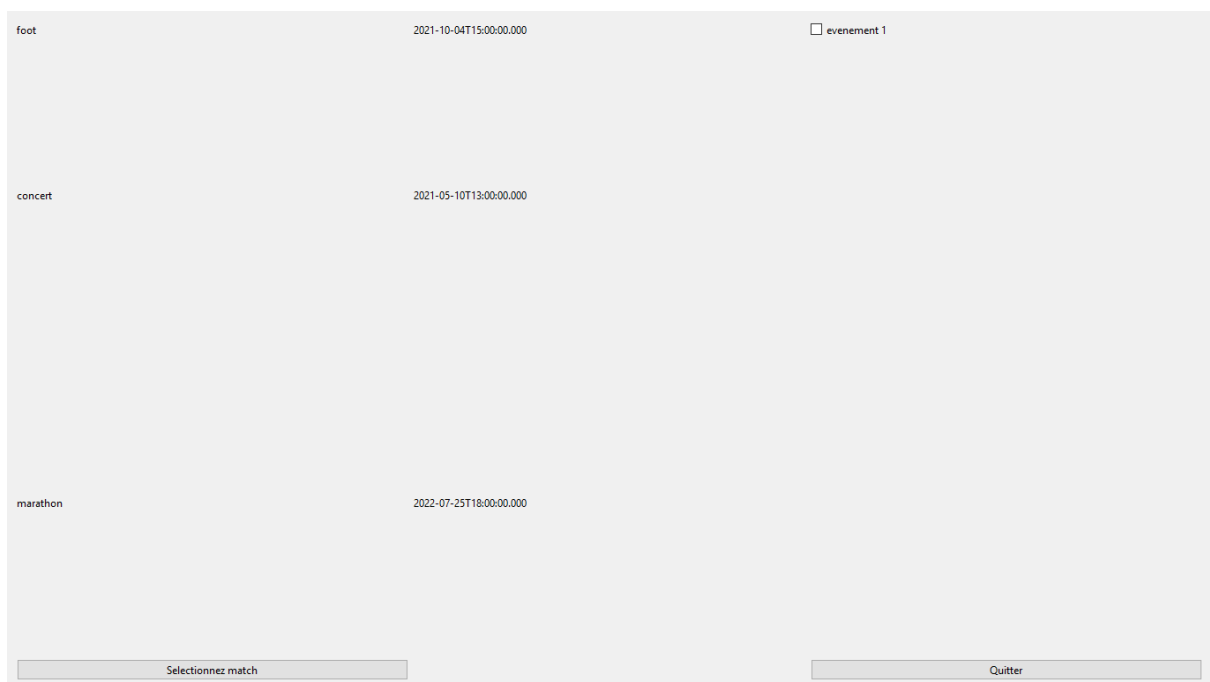
Enfin, celle-ci sert à fermer la fenêtre lorsque l'on clique sur le bouton « fermer ».

Manuel D'utilisation du logiciel caisse



A screenshot of a login interface. It features a large, empty rectangular input field for a username or password. To the right of this field is a button labeled "S'identifier". In the bottom right corner of the interface, there is a button labeled "Quitter".

Ceci est la première page du logiciel, le caissier entre son identifiant et accède à l'application en appuyant sur le bouton s'identifier.



A screenshot of a main menu or event selection screen. The background is light gray. At the top, there is a header bar with three sections: "foot" on the left, a date and time "2021-10-04T15:00:00.000" in the center, and a checkbox labeled "evenement 1" on the right. Below this, the screen is divided into three rows, each representing an event: "concert" with the date "2021-05-10T13:00:00.000", and "marathon" with the date "2022-07-25T18:00:00.000". At the bottom of the screen, there are two buttons: "Selectionnez match" on the left and "Quitter" on the right.

Page de sélection d'évènement, checkbox sur l'évènement 1

En cliquant sur le bouton sélectionnez match, l'on accède à la sélection des places du stade.

foot 2021-10-04T15:00:00.000

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Sélectionnez Place Quitter

Les différentes checkbox permettent de sélectionner une ou plusieurs places et d'ainsi les pré-réserver.

Le bouton "Sélectionner Place" permet de passer à la page de vérification de commande.

foot 2021-10-04T15:00:00.000

La place 1 = 10eu

La place 9 = 10eu

La place 3 = 20eu

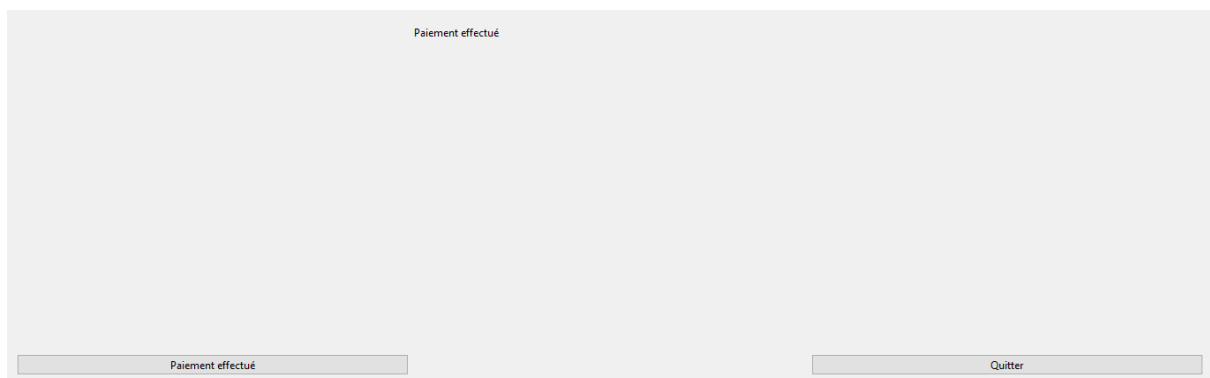
La place 7 = 10eu

50

Commande Vérifiez Quitter

La page de vérification de commande affiche la ou les places sélectionnées ainsi que leur prix. Le prix total est lui aussi affiché.

Il est possible d'effectuer une saisie afin d'appliquer une réduction si la personne est abonnée.



Cette page simule le paiement et réserve définitivement les places sélectionnées.

Manuel utilisation du site web

Yanis BOUSSNENE

1 - Installation

Afin de pouvoir utiliser notre site web, il faudra au préalable installer différents logiciels :



Wampserver :

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.

Voici un lien de téléchargement : <https://www.wampserver.com/>

Il s'agit là d'une plateforme afin d'accéder au script qui sont codés en PHP. Sans interface un navigateur google chrome ou bien internet explorer ne pourra pas traiter ces données.

2 - Utilisation du site web

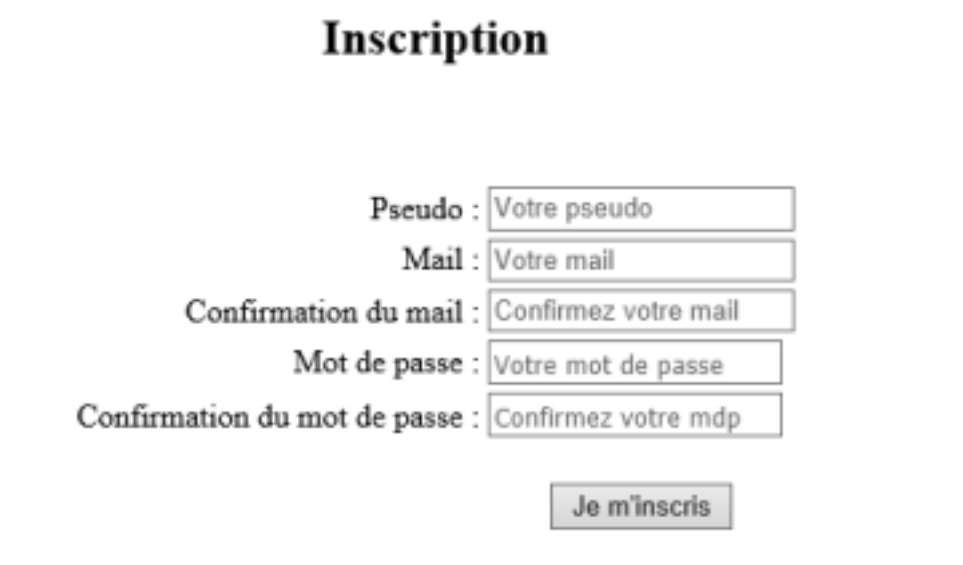
► Page d'accueil (index.html)



Tout d'abord vous tombez sur cette première page. Ensuite vous pouvez visiter le site avec les différents onglets.

► Inscription (inscription.php)

Vous pouvez également créer votre compte dans la catégorie “Inscription” comme ci-dessous :



Inscription

Pseudo :

Mail :

Confirmation du mail :

Mot de passe :

Confirmation du mot de passe :

► Connexion (connexion.php)

Vous pouvez à présent vous connecter et accéder à votre compte :



Connexion

► [Profil](#) (profil.php)

Une fois connecté voici à quoi cela ressemble :

Profil de Yanis

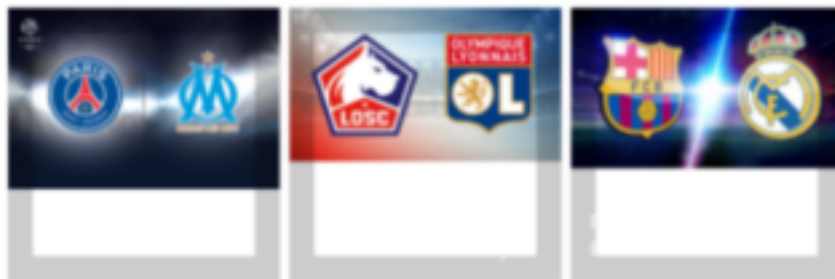
Pseudo = Yanis
Mail = yanis.boussnene@gmail.com

[Se déconnecter](#)
[Editer mon profil](#)
[Retourner sur le site !](#)

► [Boutique](#) (buynow.html)

Ici vous pouvez commander et réserver votre place de match :

Sélectionner votre match



A propos

Adresse

Dernières nouvelles

► Page de paiement (codes.html)

Une fois votre place sélectionné voici un récapitulatif de la commande :

