|  |
| --- |
|  |
| PROJET PHP 2017-2018 Système de réservations pour un spectacle |
| Rapport : chapitre3 |
|  |
|  |

Nardella Marine

Verhaeghe Julien

|  |
| --- |
|  |

Contenu

[1 Conception objet détaillée 2](#_Toc509866095)

[1.1 Calendrier prévisionnel – état d’avancement 2](#_Toc509866096)

[1.2 Standards de développement 3](#_Toc509866097)

[1.2.1 Nom des variables, couleurs, procédures d’erreur 3](#_Toc509866098)

[3.2.2 Outils coopératifs utilisés 3](#_Toc509866099)

[1.3 Modèle de données 4](#_Toc509866100)

[3.3.1 Schéma physique de la BD 4](#_Toc509866101)

[3.3.2 Fichier de création bd (SQL) 5](#_Toc509866102)

[1.4 Architecture 11](#_Toc509866103)

[3.4.1 Schéma d’architecture physique 11](#_Toc509866104)

[3.4.2 Schéma d’architecture software 11](#_Toc509866105)

[3.4.3 Organisation des répertoires du site (MVC) 11](#_Toc509866106)

[1.5 Conception détaillée de chaque Use Case 12](#_Toc509866107)

[12](#_Toc509866108)

# Conception objet détaillée

## Calendrier prévisionnel – état d’avancement

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cas d’utilisation** | **Priorité** | **Nombre d’heures prévues** | **Nombre d’heures réelles** | **Programmeur** | **Date prévue** | **Date effective** |
| **Package administration** |  |  |  |  |  |  |
| Ajouter rang utilisateur | 1 | 2 | 2 | Marine | 14/03/2018 | 13/03/2018 |
| Modifier droit utilisateur | 1 | 4 | 10 | Julien  Marine | 14/03/2018 | 12/03/2018 |
| **Package identification** |  |  |  |  |  |  |
| S’identifier | 1 | 1 | 2 | Julien | 14/03/2018 | 10/03/2018 |
| **Package réservation** |  |  |  |  |  |  |
| Gérer réservations | 1 | 8 | 13 | Marine  Julien | 14/03/2018 | 26/03/2018 |
| Gérer spectateurs | 1 | 4 | 6 | Marine | 14/03/2018 | 24/03/2018 |
| Gérer spectacles | 1 | 4 | 6 | Julien | 14/03/2018 | 24/03/2018 |
| Gérer représentations | 1 | 3 | 2 | Marine | 20/03/2018 | 23/03/2018 |
| Gestion disposition salle | 1 | 15 | 10 | Julien | 20/03/2018 | 25/03/2018 |
| Consulter liste  spectateurs | 2 | 4 |  | Marine | 29/04/2018 |  |
| Etiquettes | 3 | 15 |  | Julien | 10/05/2018 |  |
| Toutes boîtes | 3 | 20 |  | Marine | 10/05/2018 |  |
| Liste des spectateurs | 2 | 10 |  | Marine | 15/04/2018 |  |
| Liste des réservations | 2 | 12 |  | Julien | 15/04/2018 |  |
| Envoyer mail | 3 | 2 |  | Julien | 29/04/2018 |  |

## Standards de développement

### Nom des variables, couleurs, procédures d’erreur

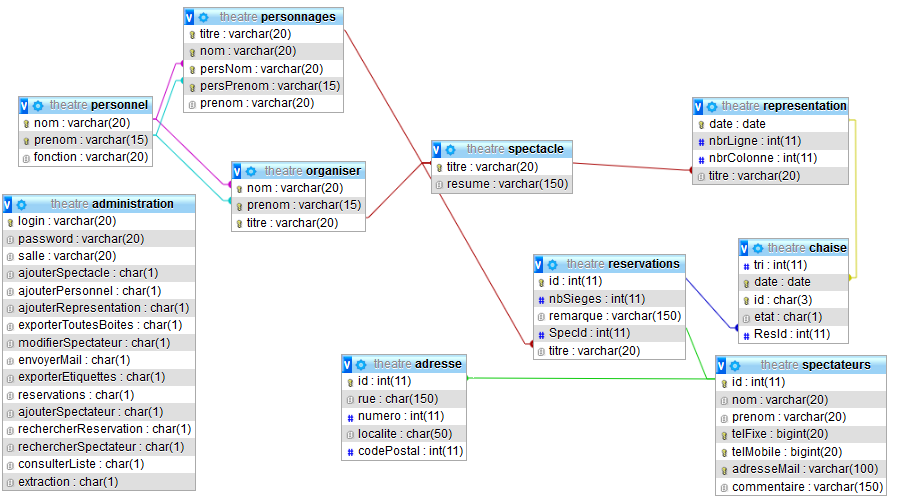
* Le nom de chaque méthode et de chaque variable sera en lower camel case.
* Les méthodes accédant à la base de données s’appelleront :
* « get… » si elles effectuent une sélection dans la base de données
* « ajouter… » si elles effectuent une insertion de données
* « modifier… » si elles modifient une valeur existante.

### 3.2.2 Outils coopératifs utilisés

* Git avec l’interface « SourceTree » : partager les sources de l’application à développer
* Dropbox : partager les fichiers autres que les sources, par exemple les fichiers d’analyse ou le rapport.

## Modèle de données

### 3.3.1 Schéma physique de la BD



### 

### 3.3.2 Fichier de création bd (SQL)

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- \* SQL MySQL generation

-- \*--------------------------------------------

-- \* DB-MAIN version: 10.0.3

-- \* Generator date: Aug 17 2017

-- \* Generation date: Thu Mar 22 11:33:26 2018

-- \* LUN file: C:\Users\Julien\Dropbox\Partage\Projet PHP\ACP\THEATRE.lun

-- \* Schema: logique/5

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Database Section

-- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

drop database if exists theatre;

create database if not exists theatre;

use theatre;

-- Tables Section

-- \_\_\_\_\_\_\_\_\_\_\_\_\_

create table ADMINISTRATION (

login varchar(20) not null,

password varchar(20) not null,

salle varchar(20) not null DEFAULT 'Z',

ajouterSpectacle char(1) not null DEFAULT 'Z',

ajouterPersonnel char(1) not null DEFAULT 'Z',

ajouterRepresentation char(1) not null DEFAULT 'Z',

exporterToutesBoites char(1) not null DEFAULT 'Z',

modifierSpectateur char(1) not null DEFAULT 'Z',

envoyerMail char(1) not null DEFAULT 'Z',

exporterEtiquettes char(1) not null DEFAULT 'Z',

reservations char(1) not null DEFAULT 'Z',

ajouterSpectateur char(1) not null DEFAULT 'Z',

rechercherReservation char(1) not null DEFAULT 'Z',

rechercherSpectateur char(1) not null DEFAULT 'Z',

consulterListe char(1) not null DEFAULT 'Z',

extraction char(1) not null DEFAULT 'Z',

constraint ID\_ADMINISTRATION primary key (login)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Déchargement des données de la table `administration`

--

INSERT INTO `administration` (`login`, `password`, `salle`, `ajouterSpectacle`, `ajouterPersonnel`, `ajouterRepresentation`, `exporterToutesBoites`, `modifierSpectateur`, `envoyerMail`, `exporterEtiquettes`, `reservations`, `ajouterSpectateur`, `rechercherReservation`, `rechercherSpectateur`, `consulterListe`, `extraction`) VALUES

('admin', '1234', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M');

create table adresse (

id int not null,

rue char(150) not null,

numero int not null,

localite char(50) not null,

codePostal int not null,

constraint FKSPE\_adr\_ID primary key (id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table CHAISE (

date date not null,

numLigne int not null,

numColonne int not null,

etat char(1) not null,

Res\_id int,

constraint ID\_CHAISE primary key (date, numLigne, numColonne)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table organiser (

nom varchar(20) not null,

prenom varchar(15) not null,

titre varchar(20) not null,

constraint ID\_organiser primary key (nom, prenom, titre)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table personnages (

titre varchar(20) not null,

nom varchar(20) not null,

Pers\_nom varchar(20) not null,

Pers\_prenom varchar(15) not null,

prenom char(1) not null,

constraint ID\_personnages primary key (titre, nom),

constraint FKjouer\_par\_ID unique (Pers\_nom, Pers\_prenom)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table PERSONNEL (

nom varchar(20) not null,

prenom varchar(15) not null,

fonction varchar(20) not null,

constraint ID\_PERSONNEL primary key (nom, prenom)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table REPRESENTATION (

date date not null,

nbrLigne int not null DEFAULT 10,

nbrColonne int not null DEFAULT 14,

titre varchar(20) not null,

constraint ID\_REPRESENTATION primary key (date)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table RESERVATIONS (

id int not null AUTO\_INCREMENT,

nbSieges int not null,

remarque varchar(150) not null,

Spec\_id int not null,

titre varchar(20) not null,

constraint ID\_RESERVATIONS primary key (id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table SPECTACLE (

titre varchar(20) not null,

resume varchar(150) not null,

constraint ID\_SPECTACLE primary key (titre)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

create table SPECTATEURS (

id int not null AUTO\_INCREMENT,

nom varchar(20) not null,

prenom varchar(20) not null,

tel\_fixe bigint,

tel\_mobile bigint,

adresse\_mail varchar(100),

commentaire varchar(150),

constraint ID\_SPECTATEURS primary key (id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Constraints Section

-- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

alter table administration

add constraint CHK\_admin\_salle

check (salle = 'Z' OR salle = 'C' OR salle = 'M');

alter table administration

add constraint CHK\_admin\_spectacle

check (ajouterSpectacle = 'Z' OR ajouterSpectacle = 'C' OR ajouterSpectacle = 'M');

alter table administration

add constraint CHK\_admin\_personnel

check (ajouterPersonnel = 'Z' OR ajouterPersonnel = 'C' OR ajouterPersonnel = 'M');

alter table administration

add constraint CHK\_admin\_representation

check (ajouterRepresentation = 'Z' OR ajouterRepresentation = 'C' OR ajouterRepresentation = 'M');

alter table administration

add constraint CHK\_admin\_ttBoite

check (exporterToutesBoites = 'Z' OR exporterToutesBoites = 'C' OR exporterToutesBoites = 'M');

alter table administration

add constraint CHK\_admin\_spectateur

check (modifierSpectateur = 'Z' OR modifierSpectateur = 'C' OR modifierSpectateur = 'M');

alter table administration

add constraint CHK\_admin\_mail

check (envoyerMail = 'Z' OR envoyerMail = 'C' OR envoyerMail = 'M');

alter table administration

add constraint CHK\_admin\_etiquettes

check (exporterEtiquettes = 'Z' OR exporterEtiquettes = 'C' OR exporterEtiquettes = 'M');

alter table administration

add constraint CHK\_admin\_reservations

check (reservations = 'Z' OR reservations = 'C' OR reservations = 'M');

alter table administration

add constraint CHK\_admin\_aSpectateur

check (ajouterSpectateur = 'Z' OR ajouterSpectateur = 'C' OR ajouterSpectateur = 'M');

alter table administration

add constraint CHK\_admin\_rechercheRes

check (rechercherReservation = 'Z' OR rechercherReservation = 'C' OR rechercherReservation = 'M');

alter table administration

add constraint CHK\_admin\_rechercheSpec

check (rechercherSpectateur = 'Z' OR rechercherSpectateur = 'C' OR rechercherSpectateur = 'M');

alter table administration

add constraint CHK\_admin\_consulterListe

check (consulterListe = 'Z' OR consulterListe = 'C' OR consulterListe = 'M');

alter table administration

add constraint CHK\_admin\_extraction

check (extraction = 'Z' OR extraction = 'C' OR extraction = 'M');

alter table chaise

add constraint CHK\_chaise

check (etat = 'd' OR etat = 'n' OR etat = 'o' OR etat = 'r');

alter table spectateurs

add constraint UC\_spec unique (tel\_fixe, tel\_mobile, adresse\_mail);

alter table representation

add constraint UC\_repr unique (date);

alter table adresse add constraint FKSPE\_adr\_FK

foreign key (id)

references SPECTATEURS (id);

alter table CHAISE add constraint FKdisposer

foreign key (date)

references REPRESENTATION (date);

alter table CHAISE add constraint FKconcerner

foreign key (Res\_id)

references RESERVATIONS (id);

alter table organiser add constraint FKorg\_SPE

foreign key (titre)

references SPECTACLE (titre);

alter table organiser add constraint FKorg\_PER

foreign key (nom, prenom)

references PERSONNEL (nom, prenom);

alter table personnages add constraint FKSPE\_per

foreign key (titre)

references SPECTACLE (titre);

alter table personnages add constraint FKjouer\_par\_FK

foreign key (Pers\_nom, Pers\_prenom)

references PERSONNEL (nom, prenom);

alter table REPRESENTATION add constraint FKpresenter

foreign key (titre)

references SPECTACLE (titre);

alter table RESERVATIONS add constraint FKreserver

foreign key (Spec\_id)

references SPECTATEURS (id);

alter table RESERVATIONS add constraint FKporter\_sur

foreign key (titre)

references SPECTACLE (titre);

-- Index Section

-- \_\_\_\_\_\_\_\_\_\_\_\_\_

## Architecture

### 3.4.1 Schéma d’architecture physique

**Outils nécessaires sur le client**

Un navigateur internet qui supporte le HTML5, le CSS3, ainsi que le JavaScript8

**Outils nécessaires sur le serveur**

Le serveur doit être un serveur PHP7 avec un moteur MariaDB 10.2.13

### 3.4.2 Schéma d’architecture software

**Langages**

* PHP version 7 : interaction avec le serveur.
* HTML version 5 : structurer le site.
* CSS version 3 : design du site + responsive design.
* JavaScript version 8 : Interactivité avec l’utilisateur.
* SQL version 2011 : traitement des données du site.

**SGBD**

* MariaDB version 10 .2.13

**Librairies**

* Jquery version 3.1.1 : librairie JavaScript visant à simplifier son utilisation.

**Logiciels**

* JasperSoft Studio version 6.5.1 : génération de rapport en utilisant les données de la base de données.

### 3.4.3 Organisation des répertoires du site (MVC)

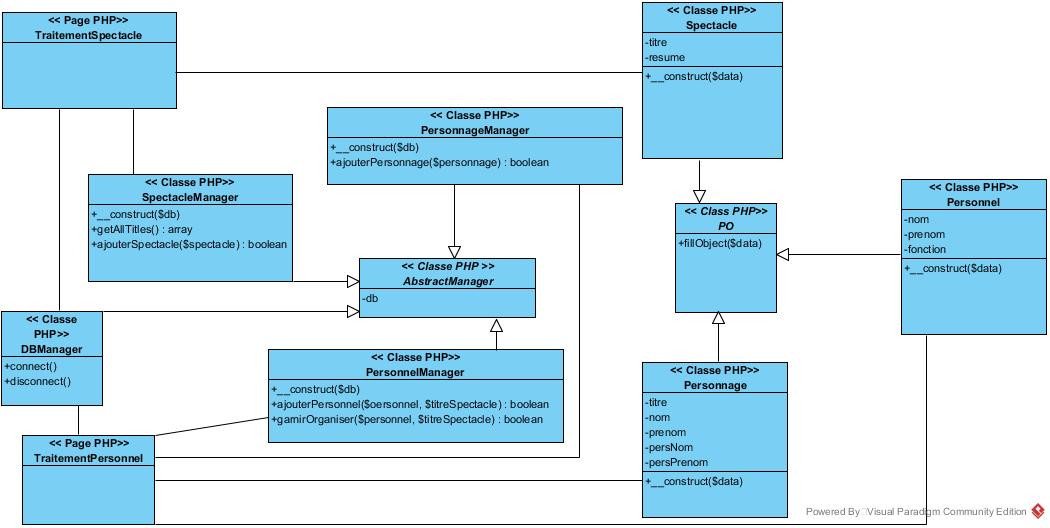
|  |  |  |  |
| --- | --- | --- | --- |
| Theatre | css |  | Les feuilles de styles |
| image |  | Image logo |
| js |  | Les librairies ou code personnel |
| manager | Couche persistance | Classes accédant à la base de données |
| model | Couche métier | Classes qui contiennent les données |
| traitements | Couche logique | Fichiers PHP qui effectuent différents traitements |
| utils | Couche logique | Fichiers contenant des fonctions utiles |
| index.php | Couche présentation | Page principale et présentation des données |

## Conception détaillée de chaque Use Case

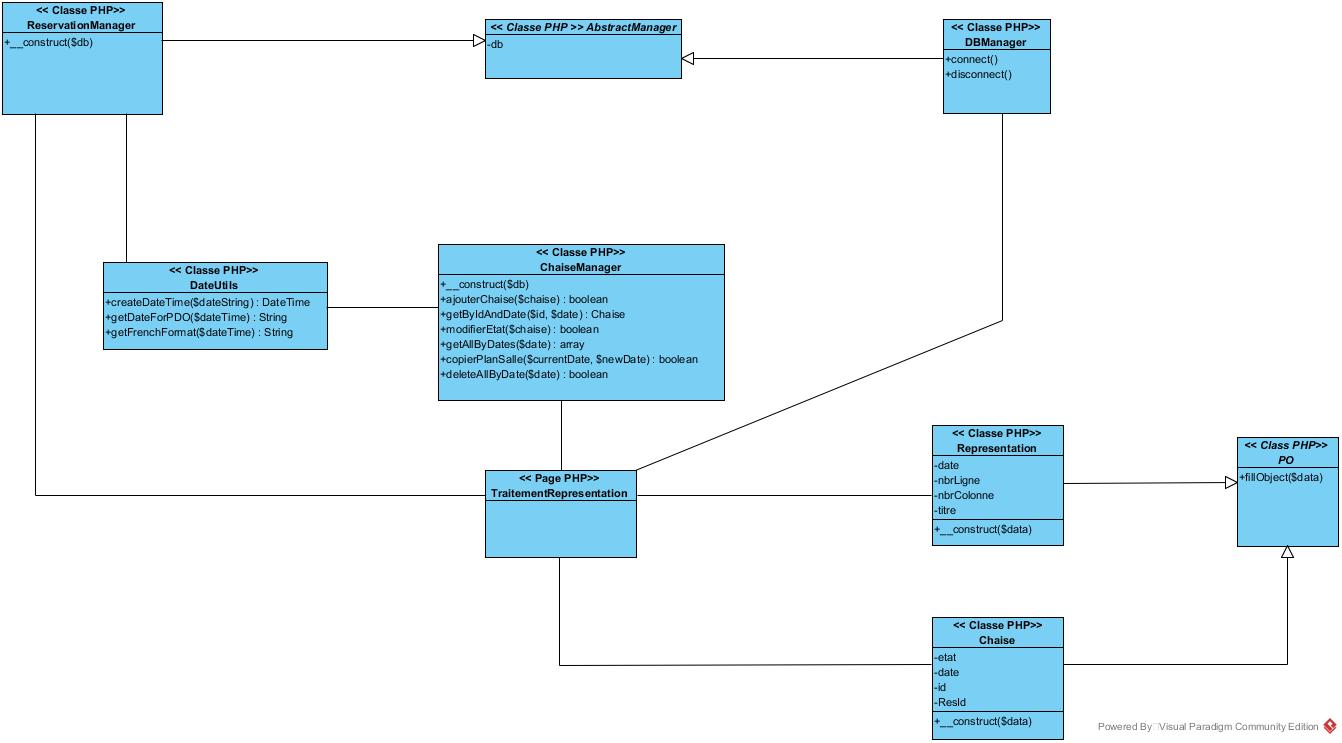
## 

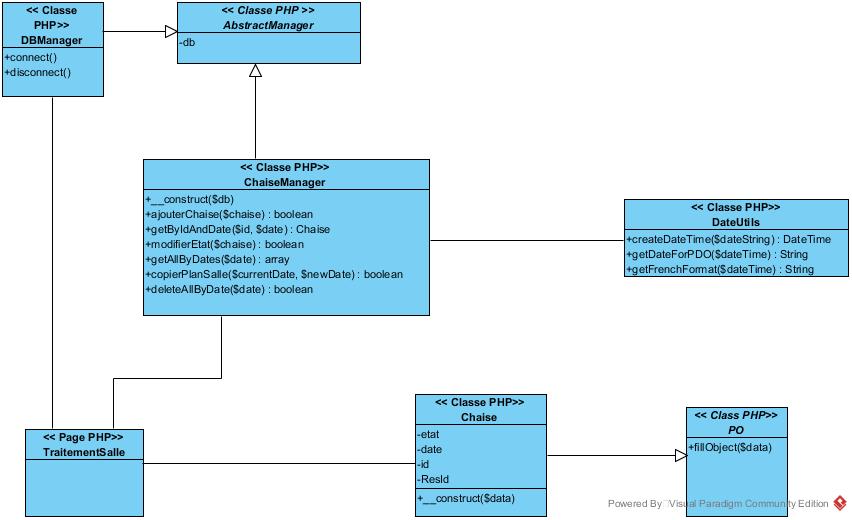
**S’identifier**

**Gérer spectacle**

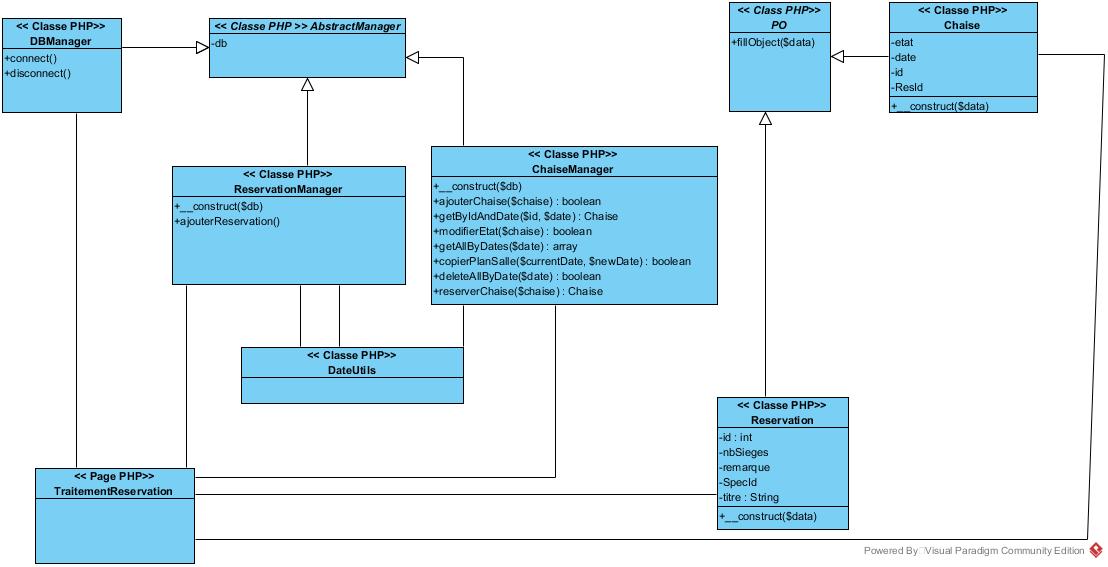
****

**Gérer représentation**

****

**Gérer salle**

**Gérer réservation**

****