

Rapport de Conception

Projet Médiathèque

Sommaire

- I. Introduction
- II. Les langages de développement
- III. Schéma relationnel de la base de données
- IV. Documentation MySQL
- V. Diagramme de classe
- VI. Diagramme de séquence des opérations
- VII. Conclusion

I. Introduction:

La médiathèque Entreprise, afin de pouvoir mettre à disposition de ses utilisateurs un site web leur permettant de gérer en ligne leurs locations, à fait appel à nous pour mettre en place ce projet. Ce projet consiste à une interface mise en place pour les utilisateur pour leurs permettre de gérer en ligne la location et la réservation de leurs produits. Les produits peuvent êtres sous la forme de livres, d'album ou bien de vidéos.

Un utilisateur doit pouvoir grâce à cette interface s'authentifier et gérer ses informations. Il pourra faire un emprunt si un article est disponible ou bien faire une réservation d'un article. Cette action lui donnera une date de retrait à laquelle il devra se rendre sur place pour récupérer son emprunt. Il ne pourra cependant effectuer ces actions que s'il dispose d'une réserve suffisante de fonds, pouvant varier selon les articles choisi. Ces fonds agissent comme une caution en cas de perte, dégradation ou vol du produit. Afin de rajouter des fonds à son porte-monnaie l'utilisateur doit se rendre sur place.

Certains salariés de la médiathèque peuvent eux aussi s'authentifier sur le site web mais contrairement aux utilisateurs ne peuvent pas effectuer d'emprunts. Leur rôle est de valider les inscriptions des utilisateurs, confirmer les informations saisies en cas d'erreur. Ils doivent aussi gérer l'arrivée de nouveaux articles et modifier le nombre d'exemplaire disponibles. Afin d'éviter les fraudes de la part des gestionnaires, lors des actions critiques une double validation est faite avec un autre gestionnaire.

Le site web doit aussi servir d'interface aux nouveaux arrivants. Les visiteurs ont le droit de voir les articles mais aussi de faire une pré-inscription en ligne afin de vérifier des informations directement en ligne et simplifier l'inscription finale sur place.

Pour mettre à bien ce projet nous disposons de trois personnes et une durée limite de rendu du produit fonctionnel de 2 mois. Ce projet doit être réalisé en intégralité avec des ajouts de fonctionnalités optionnelles si le temps le permet et que ça n'impacte pas négativement les autres fonctionnalités.

Au vue de la charge de travail nécessaire et du temps imparti nous n'avons pas jugé nécessaire la mise en place d'un framework. Cela n'aurait fait que rendre le site plus lourd étant donné la taille du projet et l'apprentissage de l'outils aurait empiété sur le développement effectif des fonctionnalités.

Cependant la méthode MVC sera utilisé afin de faciliter la structuration et le codage tout en permettant une légère sécurité en empêchant aux utilisateurs de faire des actions non voulues sur le serveur et la base de données.

II. Les langages de développement

II.1 HTML (Hyper Text Markup Language)

Le HTML est un langage dit de « marquage » (de « structuration » ou de « balisage ») dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents. Le langage HTML permet notamment la lecture de documents

sur Internet à partir de machines différentes, grâce au protocole HTTP, permettant d'accéder via le réseau à des documents repérés par une adresse unique, appelée URL.

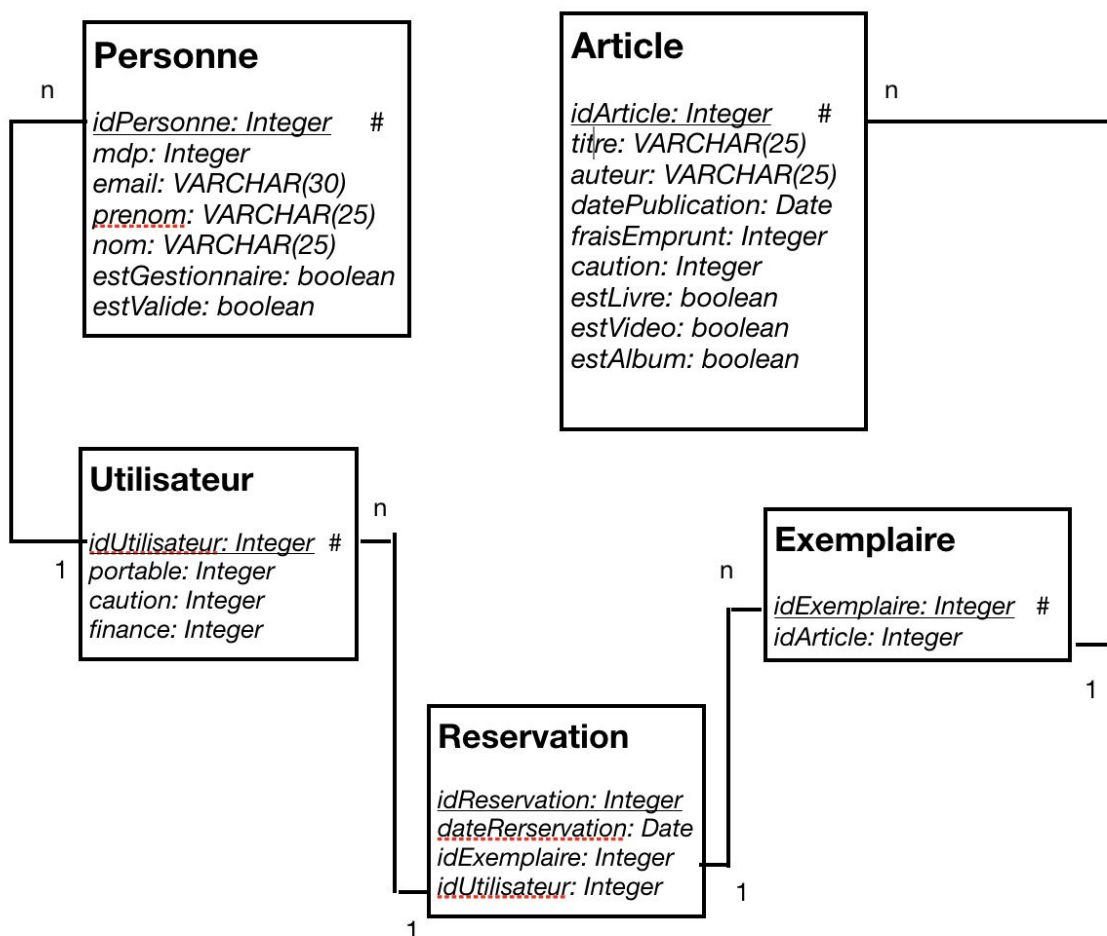
II.2 PHP :

PHP Générateur pour MySQL est un outil pour créer des applications Web orientée base de données visuellement. Il nous permet de générer des scripts PHP de haute qualité pour travailler avec des tables MySQL, vues et les requêtes à travers le web. Vous devez pas avoir de connaissances en programmation pour l'utiliser

II.3 Relation MYSQL/PHP

Le couple PHP/MySQL est très utilisé par les sites web et proposé par la majorité des hébergeurs Web. Plus de la moitié des sites Web fonctionnent sous Apache, qui est le plus souvent utilisé conjointement avec PHP et MySQL.

III. Schéma relationnel de la base de données



IV. Documentation MySQL

Types de données :

Integer : longueur de 4 octets, stocke les nombres de -2 147 483 648 à 2 147 483 647

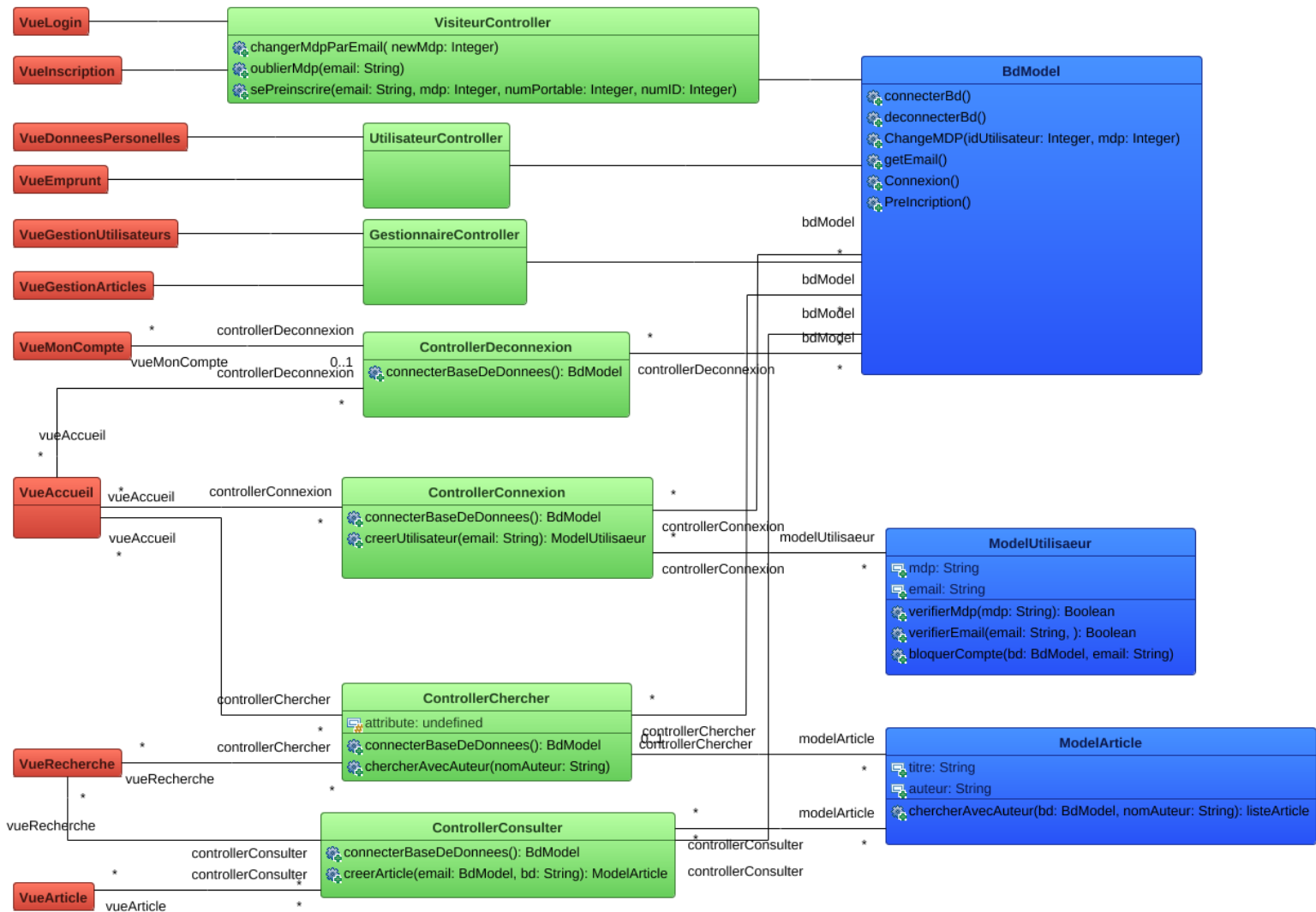
Date : sert à stocker une date. Taille de 3 octets, type de chaîne, format de AAAA-MM-JJ

MySQL supporte des DATE allant de '1001-01-01' à '9999-12-31'

VARCHAR : permet de stocker un texte relativement court (moins de 255 caractères), ce s'utilise avec un paramètre qui précise la taille que peut prendre votre texte (entre 1 et 255). VARCHAR(x) stockera jusqu'à x caractères (entre 0 et x), et stockera en plus en mémoire la taille du texte stocké.

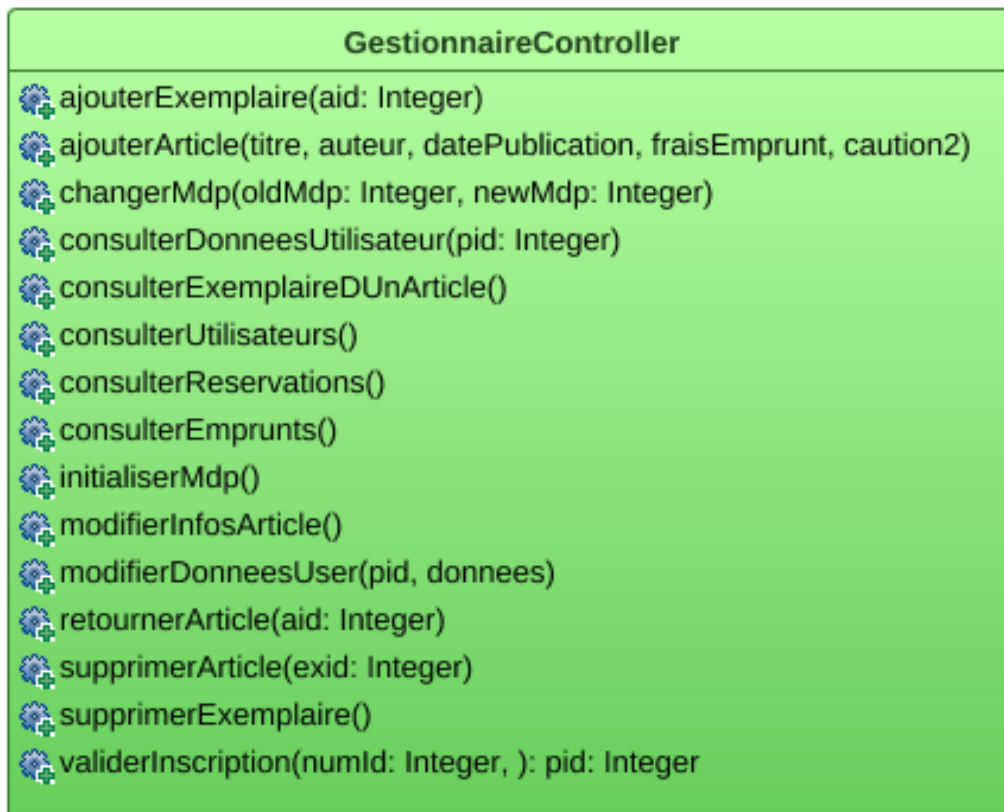
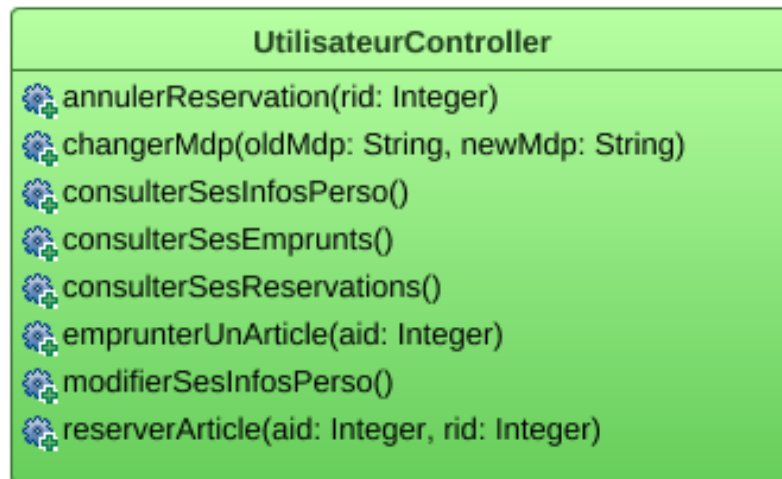
Boolean : These types are synonyms for TINYINT(1). A value of zero is considered false, Nonzero values are considered true

V. Diagramme de classe



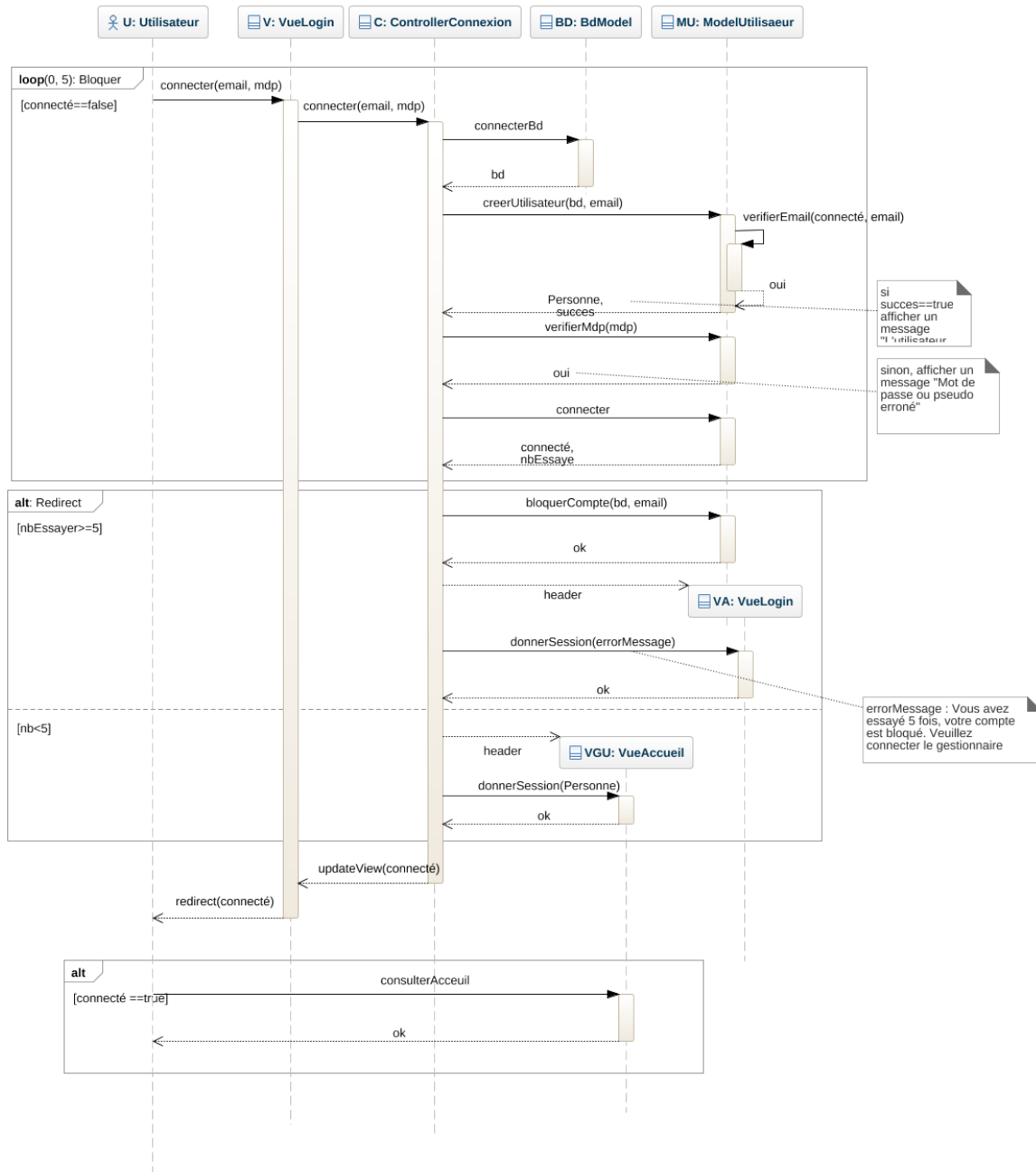
***Afin de rendre le diagramme plus lisible, les classes avec beaucoup d'attributs ou d'opérations ont été réécrite indépendamment, voir ci dessous.**

UtilisateurController et GestionnaireController :



VI. Diagramme de séquence des opérations

VI.1 Se connecter



Description

Un visiteur possédant un compte utilisateur souhaite se connecter au site web, pour cela il entre son adresse mail et son mot de passe, les deux informations sont correctes et l'utilisateur est connecté au site web.

personne.seConnecter(email: String, mdp: Integer)

Pre-conditions

$\text{email} \neq \text{null} \wedge \text{mdp} \neq \text{null}$

Post-conditions

$\exists p \in \text{bd.personne}, \text{email} = p.\text{email} \wedge \text{mdp} = p.\text{mdp}$

$\wedge \text{bd.personne.estGestionnaire} = \text{true} \Rightarrow \text{result} = \text{VueGestionnaire}$

$\exists p \in \text{bd.personne}, \text{email} = p.\text{email} \wedge \text{mdp} = p.\text{mdp}$

$\wedge \text{bd.personne.estGestionnaire} = \text{false} \Rightarrow \text{result} = \text{VueUtilisateur}$

$\forall p \in \text{bd.personne}, \text{email} \neq p.\text{email} \wedge \text{mdp} \neq p.\text{mdp} \Rightarrow \text{result} = \text{VueLogin avec message d'erreur}$

Arguments

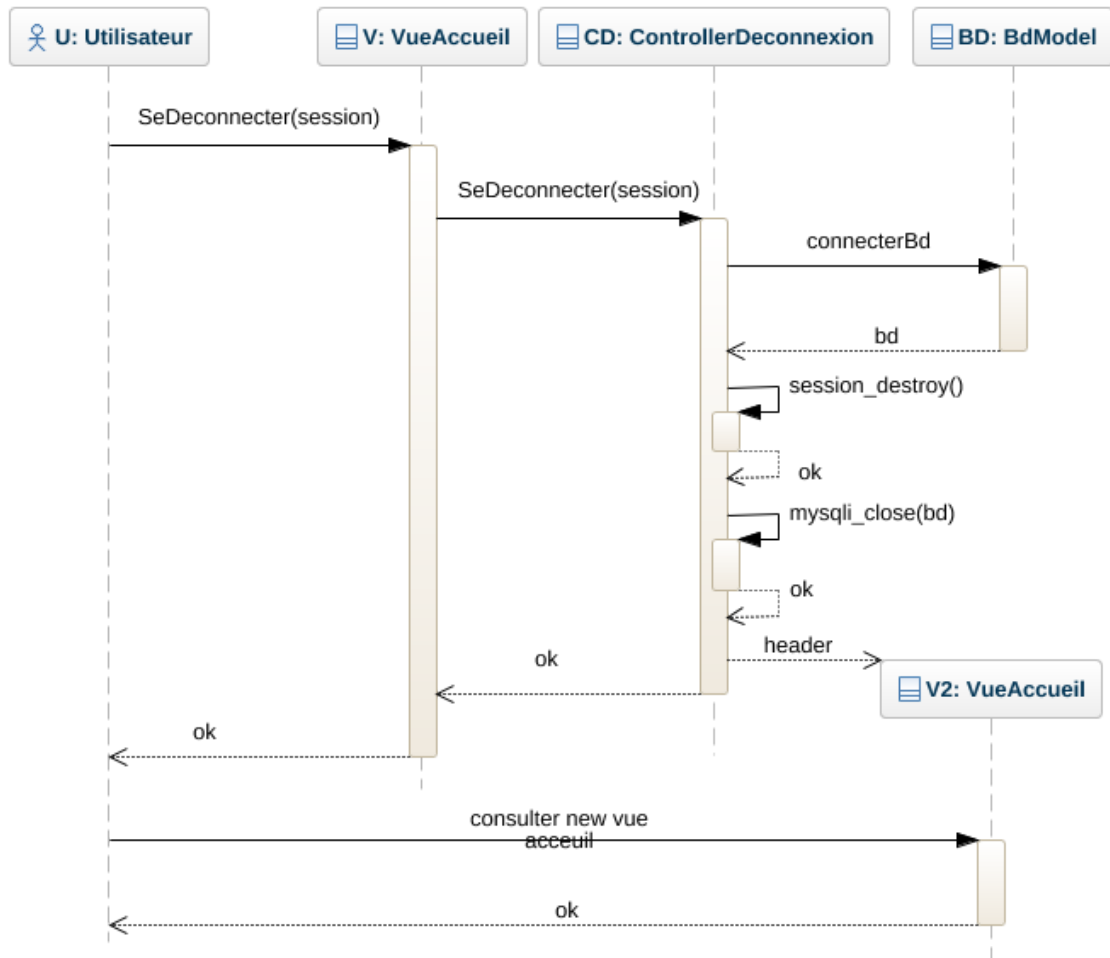
email, motDePasse

Type des arguments et de retour

email : String

mdp : Integer

VI.2 Se déconnecter



Description

Un utilisateur étant connecté au site web souhaite se déconnecter, pour cela il peut cliquer sur le bouton spécifique. Cette action le déconnecte automatiquement du site web.

Pre-conditions

utilisateur s'est déjà connecté

Post-condition

result = vueAccueil

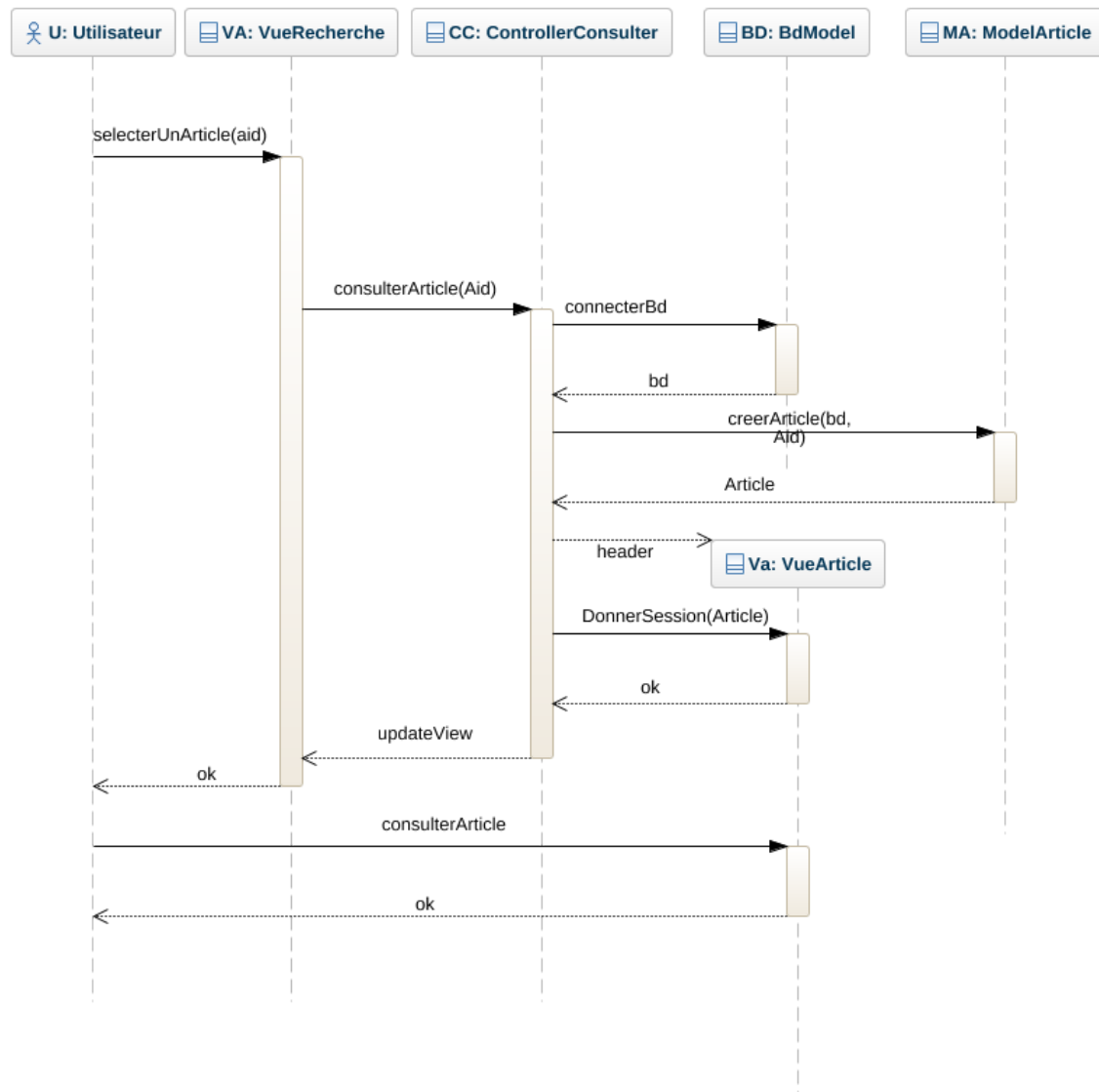
Arguments

session : url

Type des arguments et de retour

url : String

VI.3 Consulter un article



Description

Un visiteur souhaite consulter les détails d'un article, pour cela il clique sur le lien (représenté par une image et une brève description de l'article), les informations sont donc affichées.

`selecterUnArticle(aid) : String`

Pre-conditions

$\exists a \in \text{bd.article}, \text{aid} = a.\text{idArticle}$

Post-condition

$a.\text{estLivre} = \text{true} \Rightarrow \text{resultat} = \text{"informations : " } \vee a.\text{titre} \vee a.\text{auteur} \vee a.\text{datePublication} \vee a.\text{fraisEmprunt} \vee a.\text{caution} \vee \text{"type : livre"}$

a.estVideo = true => resultat = "informations : " ∨ a.titre ∨ a. auteur ∨
a.datePublication ∨ a. fraisEmprunt ∨ a. caution ∨ "type : vidéo"

a.estAlbum = true => resultat = "informations : " ∨ a.titre ∨ a. auteur ∨
a.datePublication ∨ a. fraisEmprunt ∨ a. caution ∨ "type : album"

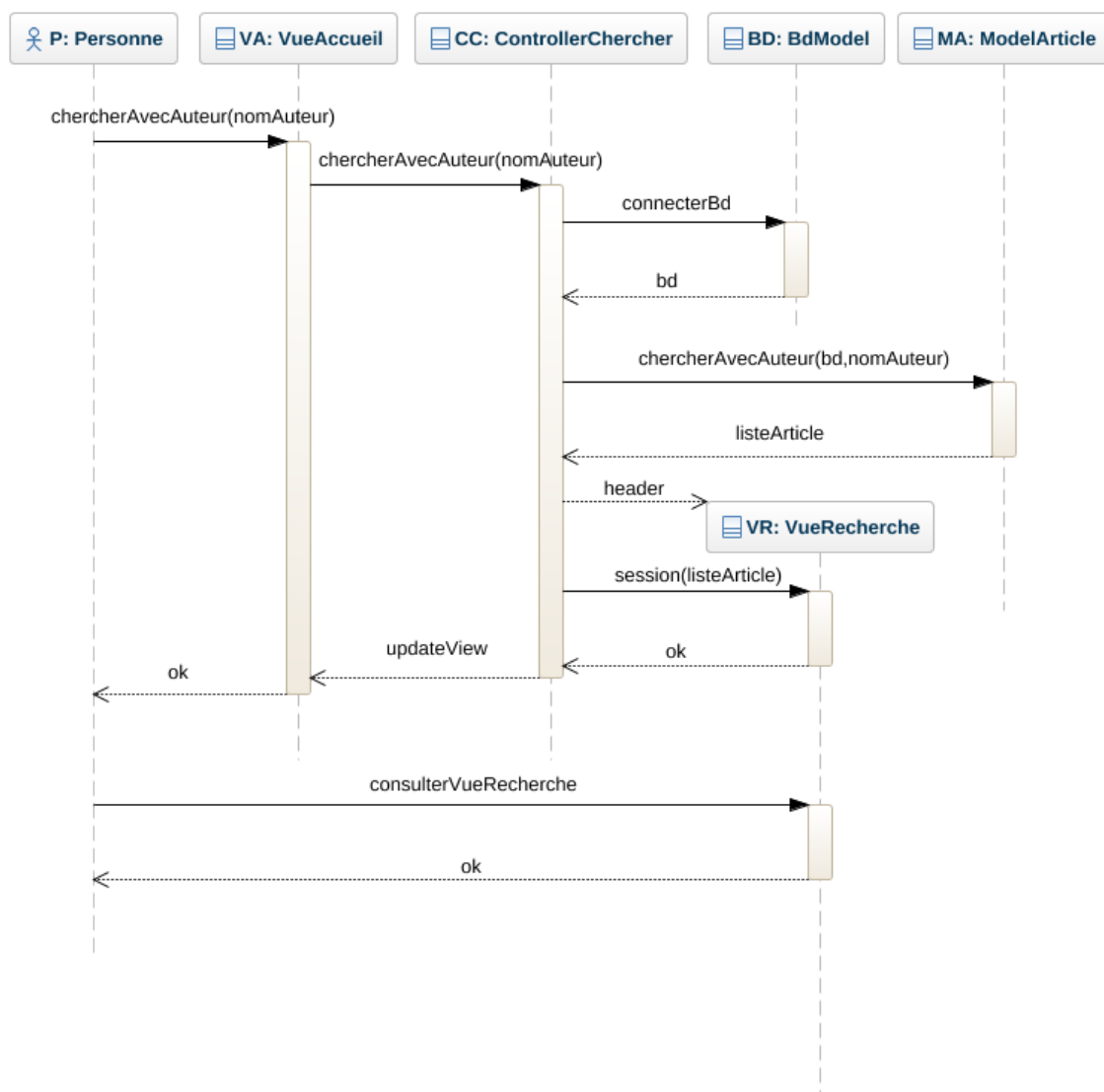
Arguments

aid

Type des arguments et de retour

aid : Integer

VI.4 Chercher avec le nom d'auteur



Description

Un visiteur souhaite chercher un article sur le site web, pour cela il entre un critère (ici par exemple le nom d'auteur), le système renvoie à l'utilisateur une sélection d'articles contenant le critère dans leur description.

chercherAvecAuteur(auteur) : String

Pre-conditions

auteur \neq null

Post-condition

$\exists a \in \text{bd.article}, \text{auteur} = a.\text{auteur}, a.\text{estLivre} = \text{true} \Rightarrow \text{resultat} = \text{"informations : " } \vee a.\text{titre} \vee a.\text{auteur} \vee a.\text{datePublication} \vee a.\text{fraisEmprunt} \vee a.\text{caution} \vee \text{"type : livre"}$

$\exists a \in \text{bd.article}, \text{auteur} = a.\text{auteur}, a.\text{estVideo} = \text{true} \Rightarrow \text{resultat} = \text{"informations : " } \vee a.\text{titre} \vee a.\text{auteur} \vee a.\text{datePublication} \vee a.\text{fraisEmprunt} \vee a.\text{caution} \vee \text{"type : vidéo"}$

$\exists a \in \text{bd.article}, \text{auteur} = a.\text{auteur}, a.\text{estAlbum} = \text{true} \Rightarrow \text{resultat} = \text{"informations : " } \vee a.\text{titre} \vee a.\text{auteur} \vee a.\text{datePublication} \vee a.\text{fraisEmprunt} \vee a.\text{caution} \vee \text{"type : album"}$

Arguments

auteur

Type des arguments et de retour

auteur : String

VII. Conclusion

Avec les informations décrites ci dessus nous avons tous les outils nécessaire afin de faire un développement rapide. En effet un squelette du code peut être généré à partir de ces informations et seule la complétion de ce code devrait être nécessaire. L'architecture MVC devrait permettre une séparation bien définie des rôles dans le site web et permettre aussi aux développeurs de pour travailler séparément de leur côté sans avoir de problème de compatibilité entre les différents code. De plus une supervision générale du projet sera minimale car chaque tranches de travail sont claires et séparées.

Il faut noter que même si nous allons utiliser le squelette décrits ci dessus il peut être soumis à des changements minimes au cours du développement dues à des imprévues, des échecs lors des tests ou bien l'ajout de nouvelles fonctionnalités si le temps le permet.

récrire le diagramme de classe (association, héritage) correspondant à la language qu'on veut utiliser (framework) -> Ajouter/Modifier des clés uniques pour chaque classes

codage (unicode / utf-8) (contient pas de caractere commentaire)

Enlever les calculs avec des types flottant (les remplacer par des integers)

MCD -> la conception de la base de données (Juste un diagramme de classe représentant la base de données est suffisant

pour chaque fonctionnalité, le diagramme de séquence + pré/post conditions

Dessins/Schémas des vues

Sécurité: Empêcher un utilisateur de récupérer/modifier des données critiques en faisant des requêtes SQL.