

BAKKA Oulaya
GIRAUD Philémon
LTIFI Mohamed Ayoub
SIROT Emilie

Groupe 1

Projet d'intégration

Master CCI 2020-2021

Audit 1



I. Introduction



m2cci-oulaya-bakka



m2cci-emilie-sirot



m2cci-mohamed-ltifi



m2cci-philemon-giraud

Projet d'intégration
Master 2 CCI

Du 10 Mars 2021 au 06 Avril 2021

AUDIT N° 1



Plan de la présentation

- ☐ Introduction
- ☐ Contexte
- ☐ Organisation
- ☐ Etat d'avancement
- ☐ Avancement principal
- ☐ Prochaines Étapes

II. Contexte



L'entreprise MyTheatre souhaite bénéficier d'un site internet afin de vendre des billets pour des représentations de théâtre.

Notre équipe de développeurs doit donc concevoir et développer un site WEB utilisable pour ce théâtre.

Pour cela des scénarios nous ont été fournis par le client.

III. Organisation

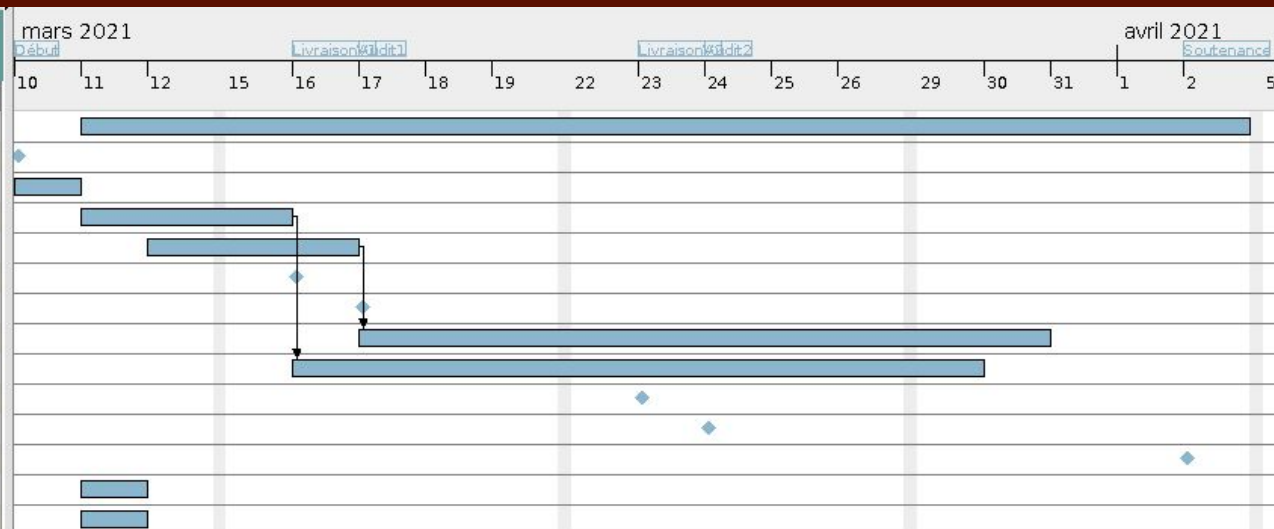
III. Organisation

Audit 1

Issue #6 : Planning de Gantt

Planning prévisionnel

GANTT project		
Nom	Date de début	Date de fin
• Suivi du projet	11/03/2021	02/04/2021
• Début	10/03/2021	10/03/2021
• Glossaire	10/03/2021	10/03/2021
• Base de données	11/03/2021	15/03/2021
• Conception IHM	12/03/2021	16/03/2021
• LivraisonV1	16/03/2021	16/03/2021
• Audit1	17/03/2021	17/03/2021
• Développement IHM	17/03/2021	30/03/2021
• Développement BackEnd	16/03/2021	29/03/2021
• LivraisonV2	23/03/2021	23/03/2021
• Audit2	24/03/2021	24/03/2021
• Soutenance	02/04/2021	02/04/2021
• #36 projet.reunion	11/03/2021	11/03/2021
• #35 projet.soutenance	11/03/2021	11/03/2021

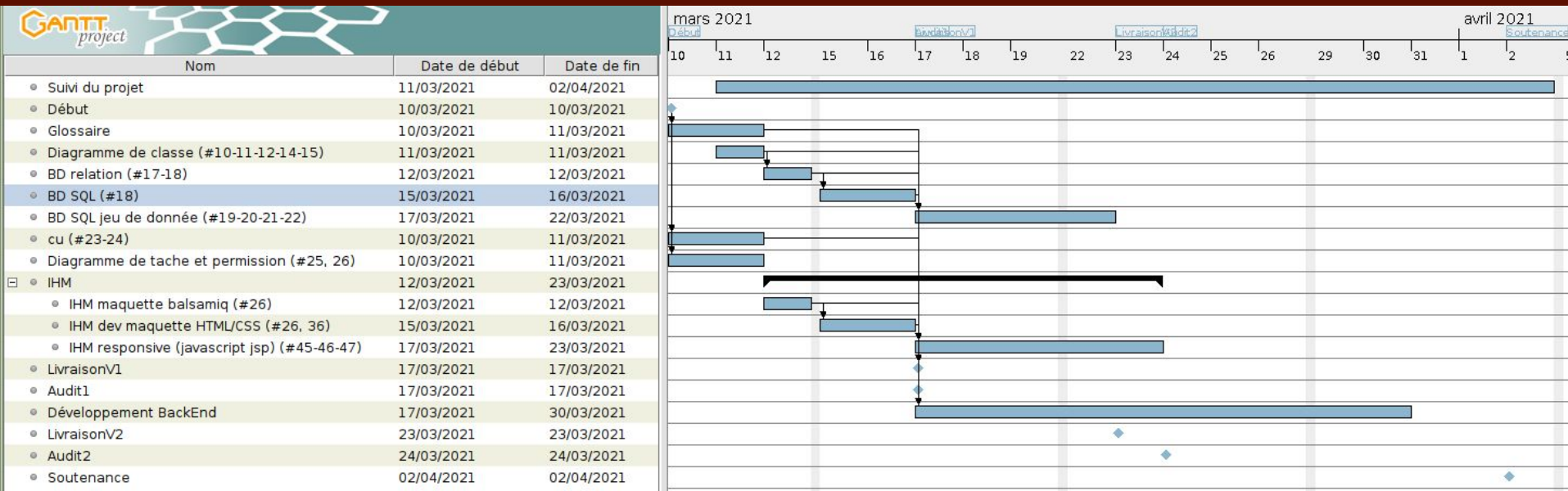


III. Organisation

Audit 1

Issue #6 : Planning de Gantt

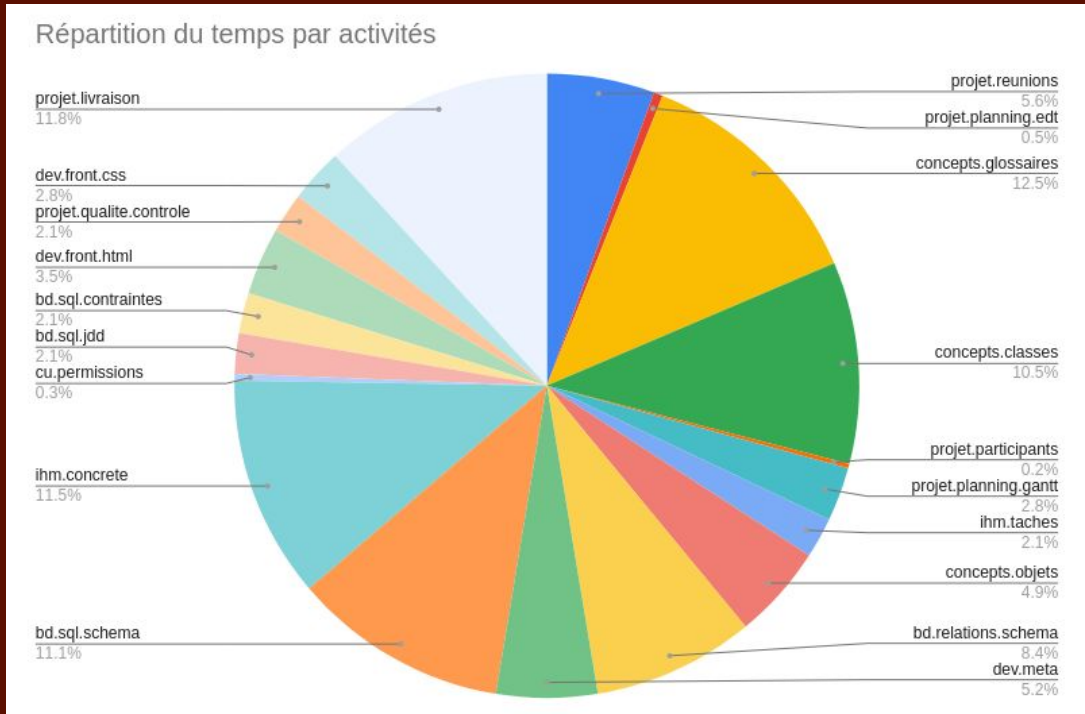
Planning effectif



III. Organisation

Audit 1

Issue #8 : Timetracking



IV. Etat d'avancement

V. Avancement principal

Issue #3, #4 et #5 : Glossaire, participants, suivi

Issues #10 et #11 : Diagramme d'objet, diagramme de classes

Issues #17 à #22 : Base de donnée

Issues #26 : IHM

Issue #46-47 : Implémentation IHM

IV. Etat d'avancement

Issue #3 : Glossaire terminé ✓

Issue #4 : Participants définis ✓

Issue #5 : projet.suivis mis à jour au fur et à mesure

```
Question Q1: Peut-on `acheter` de `billets` sans être `Utilisateurs` ?
```

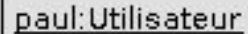
```
| Oui, une personne peut décider d'acheter des `billets` sans fournir d'informations  
| personnelles. Elle n'est alors pas un `Utilisateur` enregistré du site.  
| Pour faire une pré-réservation, il faut par contre être `Utilisateur`
```

```
decision D2: Les numeros des `sièges`
```

```
| Les numeros des `sièges` recommencent à 1 dans chage `rang`  
| donc un `siège` est défini par un numéro de `siège` et un numéro de `rang`
```

V. Avancement principal

Issues #10 et #11 : Diagramme d'objets, diagramme de classe

A UML object notation consisting of a rectangular box with a double border. Inside the box, the text 'paul:Utilisateur' is written, where 'paul' is the object name and 'Utilisateur' is the class name.

V. Avancement principal

Issues #10 et #11 : Diagramme d'objets, diagramme de classe

paul:Utilisateur

leMaladeImaginaire:SpectacleHumoristique

V. Avancement principal

Issues #10 et #11 : Diagramme d'objets, diagramme de classe

paul:Utilisateur

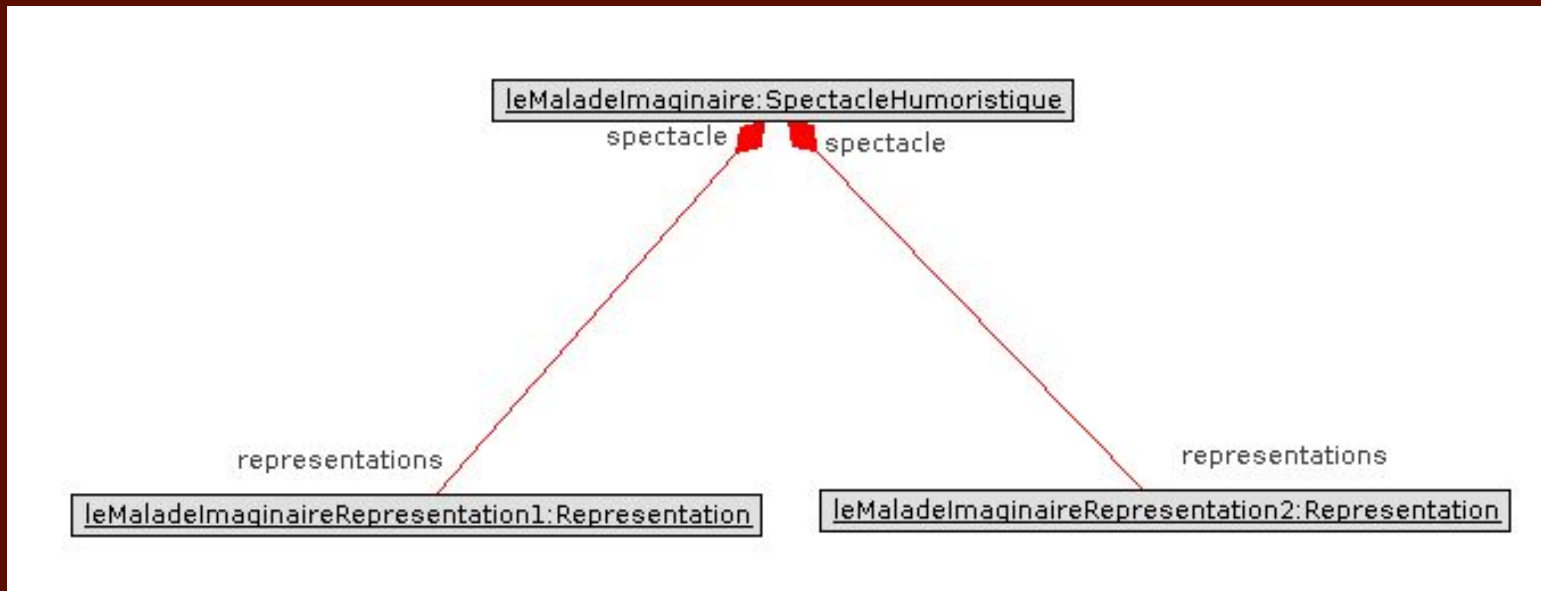


leMaladeImaginaire:SpectacleHumoristique

V. Avancement principal

Audit 1

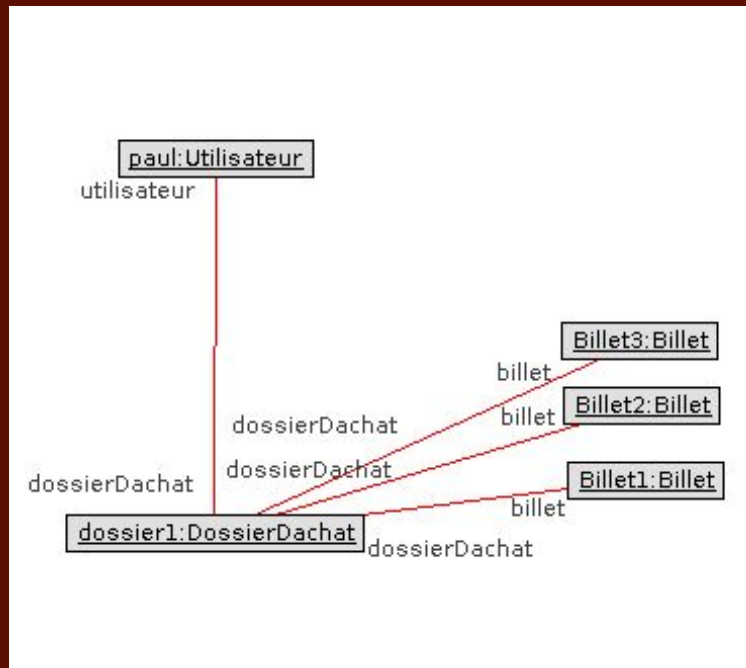
Issues #10 et #11 : Diagramme d'objets, diagramme de classe



V. Avancement principal

Audit 1

Issues #10 et #11 : Diagramme d'objets, diagramme de classe



V. Avancement principal

Audit 1

Issues #10 et #11 : Diagramme d'objets, diagramme de classe

siege1:Siege

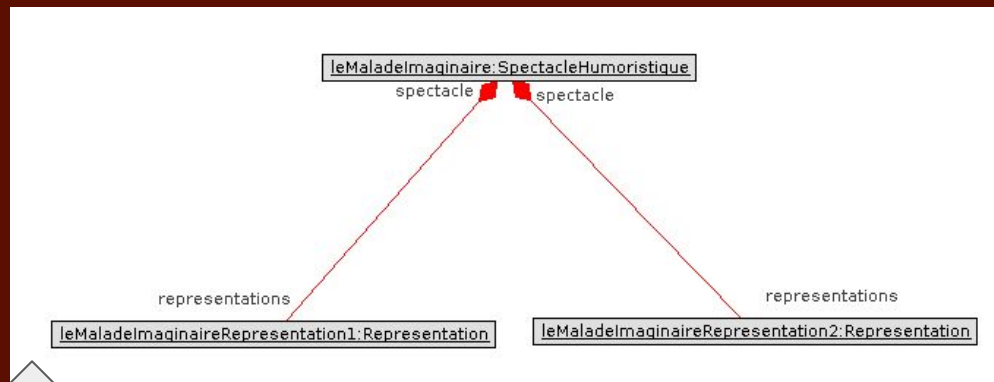
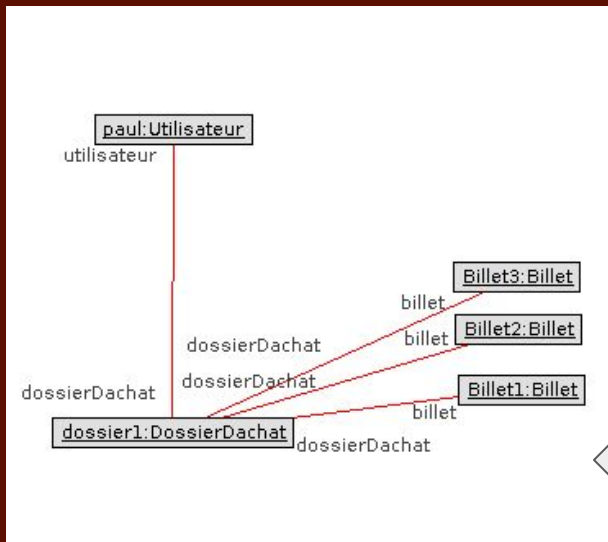
siege2:Siege

siege3:Siege

V. Avancement principal

Audit 1

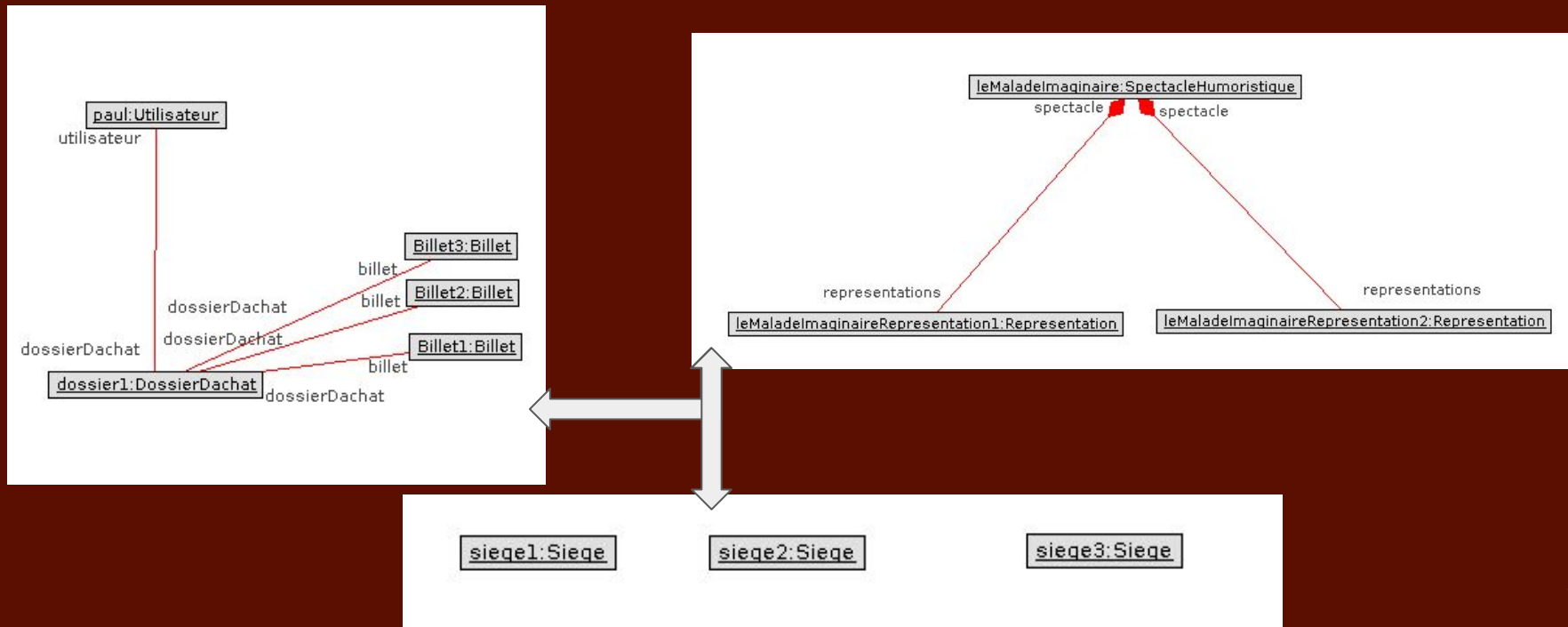
Issues #10 et #11 : Diagramme d'objets, diagramme de classe



V. Avancement principal

Audit 1

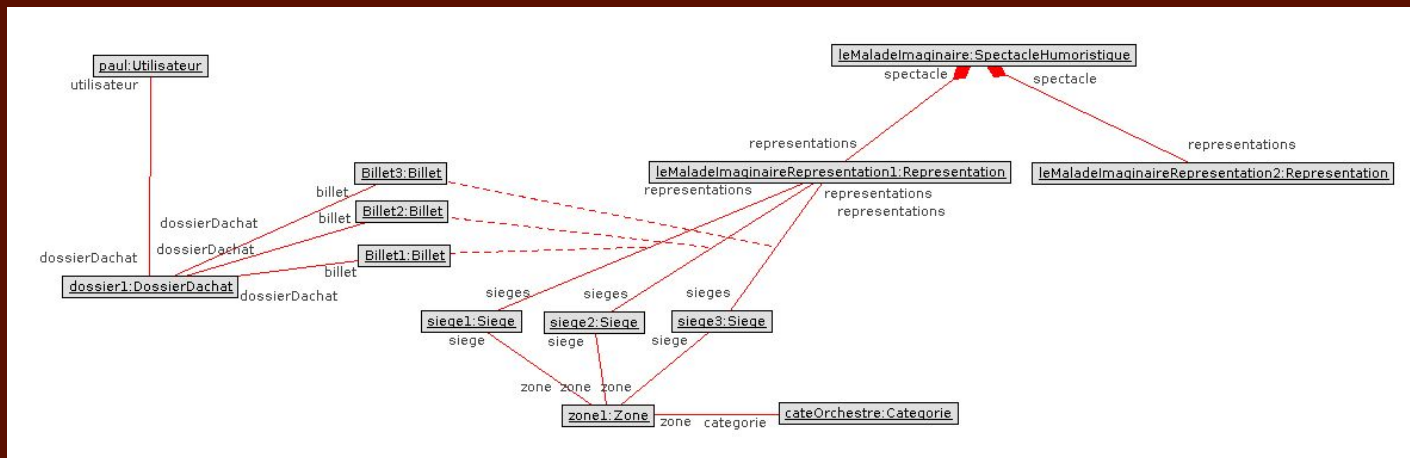
Issues #10 et #11 : Diagramme d'objets, diagramme de classe



V. Avancement principal

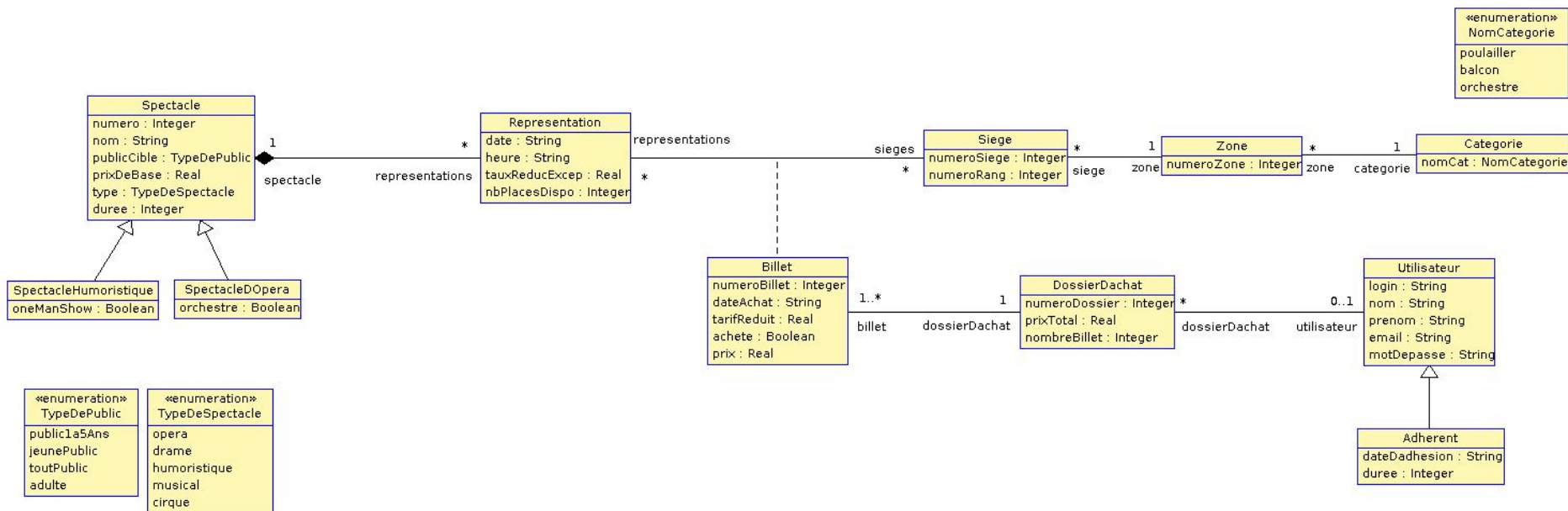
Audit 1

Issues #10 et #11 : Diagramme d'objets, diagramme de classe



V. Avancement principal

Issues #10 et #11 : Diagramme d'objets, diagramme de classe



V. Avancement principal

Audit 1

Issues #17 à #22 : Base de donnée

Issue #17 : bd.relations.schema ✓

Issue #18 : bd.sql.schema ✓

Issue #19 : bd.sql.jdd ✓

Issue #20: bd.sql.jdd.contraintes ✓

Issue #19 : bd.sql.jdd.négatifs ✓

Issue #22: bd.sql.jdd.requêtes

V. Avancement principal

Audit 1

Issues #17 à #22 : Base de donnée**Issue #17 : bd.relations.schema**


















```
Relation Spectacle(numeroSpect_, nomSpect, publicCibleSpect, prixDeBaseSpect, typeSpect, dureeSpect)
intention
    (nu, no, pc, pb, t, d) dans <=>
    | Le spectacle de numero <nu>, du nom <no> et de type <t>, a pour public cible <pc>
    | Son prix de base est <pb>. Il dure une période <d> du temps
examples
    (3000, 'Le Malade imaginaire de Molière', 'Adulte', 20.0, 'humouristique', '02:30')
constraints
    dom(numeroSpect) = Integer
    dom(prixDeBaseSpect) = Real
    dom(nomSpect, publicCibleSpect, dureeSpect) = String
    dom(typeSpect) = TypeDeSpectacle
    key numeroSpect
```

V. Avancement principal

Audit 1

Issues #17 à #22 : Base de donnée

Issue #18 : bd.sql.schema

Name	Type	Schema
▼  Tables (10)		
>  Adherents		CREATE TABLE Adherents(lo
>  Billets_base		CREATE TABLE Billets_base(
>  DossierDachats_base		CREATE TABLE DossierDacha
>  Representations_Base		CREATE TABLE Representatio
>  Sieges		CREATE TABLE Sieges(nume
>  Spectacles		CREATE TABLE Spectacles(n
>  SpectaclesDOperas		CREATE TABLE SpectaclesDC
>  SpectaclesHumoristiques		CREATE TABLE SpectaclesHu
>  UtiPossedeDoss		CREATE TABLE UtiPossedeDc
>  Utilisateurs		CREATE TABLE Utilisateurs(l
 Indices (0)		
▼  Views (3)		
 Billets		CREATE VIEW Billets(dateRep
 DossierDachats		CREATE VIEW DossierDachat
>  Representations		CREATE VIEW Representatio
 Triggers (0)		

V. Avancement principal

Audit 1

Issues #17 à #22 : Base de donnée

Issue #19 : bd.sql.jdd

Table: Spectacles

numeroSpect	nomSpect	publicCibleSpect	prixDeBaseSpect	dureeSpect	typeSpect
Filter	Filter	Filter	Filter	Filter	Filter
1	1 Le malade imaginaire de Molière	adulte	19	2H:30	humoristique
2	2 L'Avare de Molière	adulte	19	2H	humoristique
3	3 La Somnanbule	toutPublic	20	1H:45	opera
4	4 Tosca	toutPublic	25	2H:30	opera
5	5 Là-bas, de l'autre côté de l'eau	jeunePublic	20	3H	drame
6	6 Le Rock de la sorcière	public1a5Ans	15	2H	musical
7	7 Grattouillis	public1a5Ans	15	1H:30	musical
8	8 L'amour est une fumée faite de la ...	jeunePublic	25	2H30	drame
9	9 Cirque phénix	toutPublic	20	2H	cirque

V. Avancement principal

Issues #17 à #22 : Base de donnée

Issue #20 : bd.sql.jdd.contraintes

Contraintes PK: Les contraintes qui définissent la clé de la table

Exemple:

```
CONSTRAINT PK  
    PRIMARY KEY (dateRep, heureRep),
```

Contraintes FK: Foreign Key

Exemple:

```
CONSTRAINT FK_Spectacles  
    FOREIGN KEY (numeroSpect) REFERENCES Spectacles(numeroSpect),
```

Contraintes CK: Check

Exemple:

```
CONSTRAINT CK_Representation CHECK (tauxReducExcepRep >= 0 AND  
    <= 1);
```

V. Avancement principal

Issues #17 à #22 : Base de donnée

Issue #21 : bd.sql.jdd.négatifs

Exemples de jeu de données négatif selon les contraintes:

Contraintes ck_mon_1 et ck_mon_2:

```
INSERT INTO Spectacles VALUES (1, 'Le malade imaginaire de  
Molière', 'Adultes', 19, '2H:30', 'humoristique');
```

Les mots “Adultes” et “humoristique” ne respectent pas ces contraintes car ils n’appartiennent pas aux énumérations ‘typeSpect’ et ‘publicCibleSpect’.

V. Avancement principal

Issues #17 à #22 : Base de donnée

Issue #21 : bd.sql.jdd.négatifs

Exemples de jeu de données négatif selon les contraintes:
Contraintes FK_spectacles:

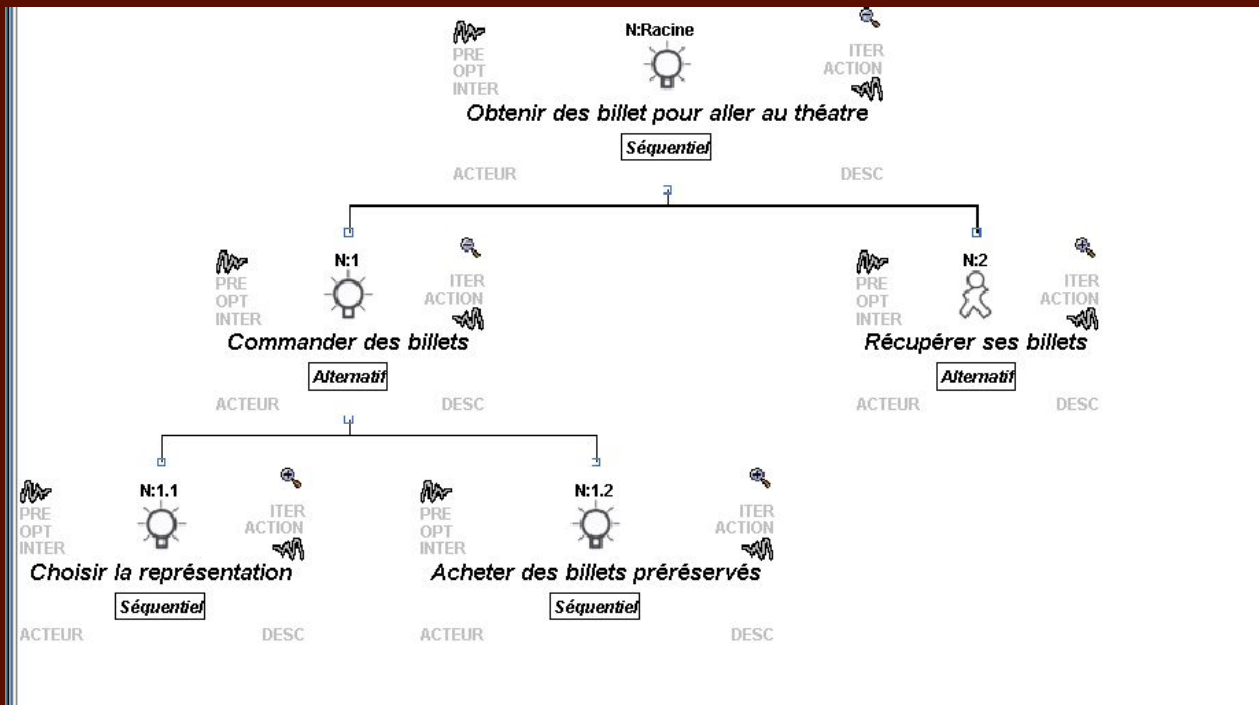
```
INSERT INTO Representations_Base VALUES ('08-03-2021', '18:00:00', 150 , 75 , 0.25 )  
;
```

Le numéro de Spectacle “numéroSpect” n’est pas inséré à l’avance dans la table spectacles

V. Avancement principal

Issue #25 : ihm.tâches

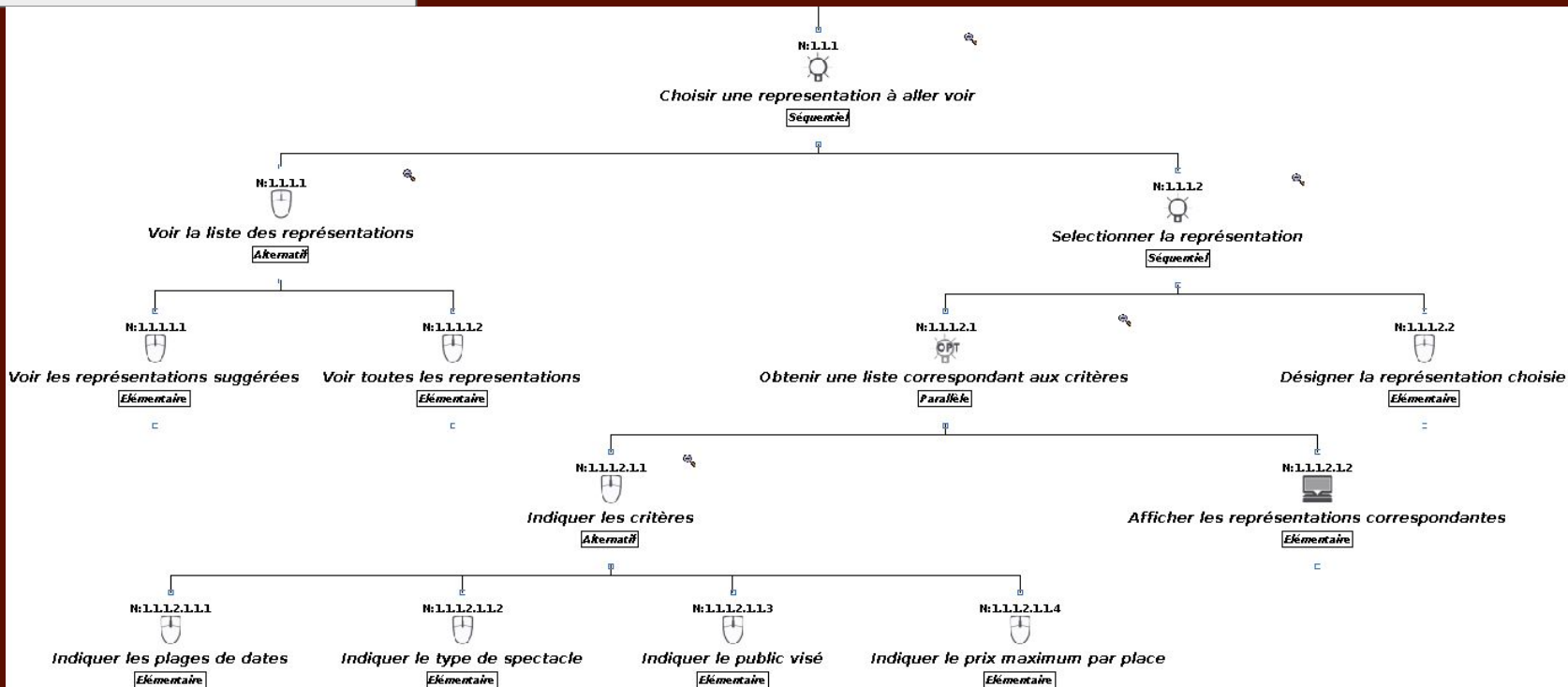
Comment obtenir des billets pour aller au théâtre ?



V. Avancement principal

Issue #25 : ihm.tâches

Comment obtenir des billets pour aller au théâtre ?



V. Avancement principal



Audit 1

Issue #26 : ihm.concrete

Charte graphique

Couleurs :

- rouge
- noir
- jaune (doré)

→ Couleurs associées au théâtre



Logo créé

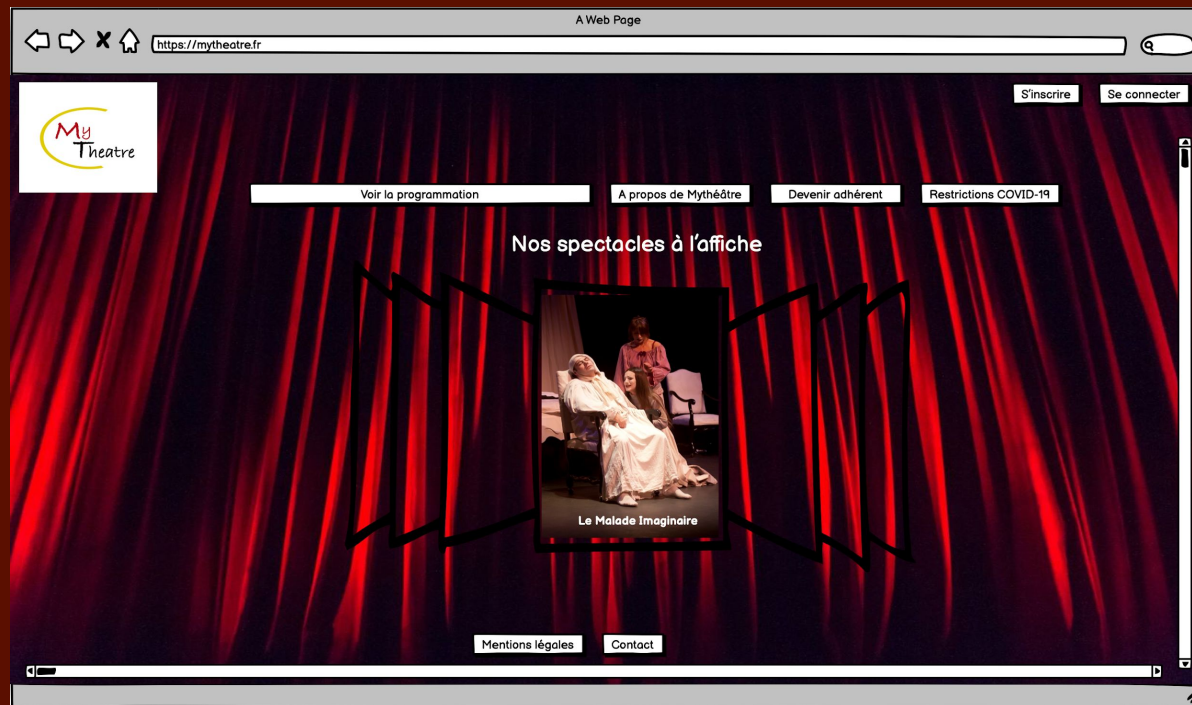


V. Avancement principal

Issue #26 : ihm.concrete

Maquettes

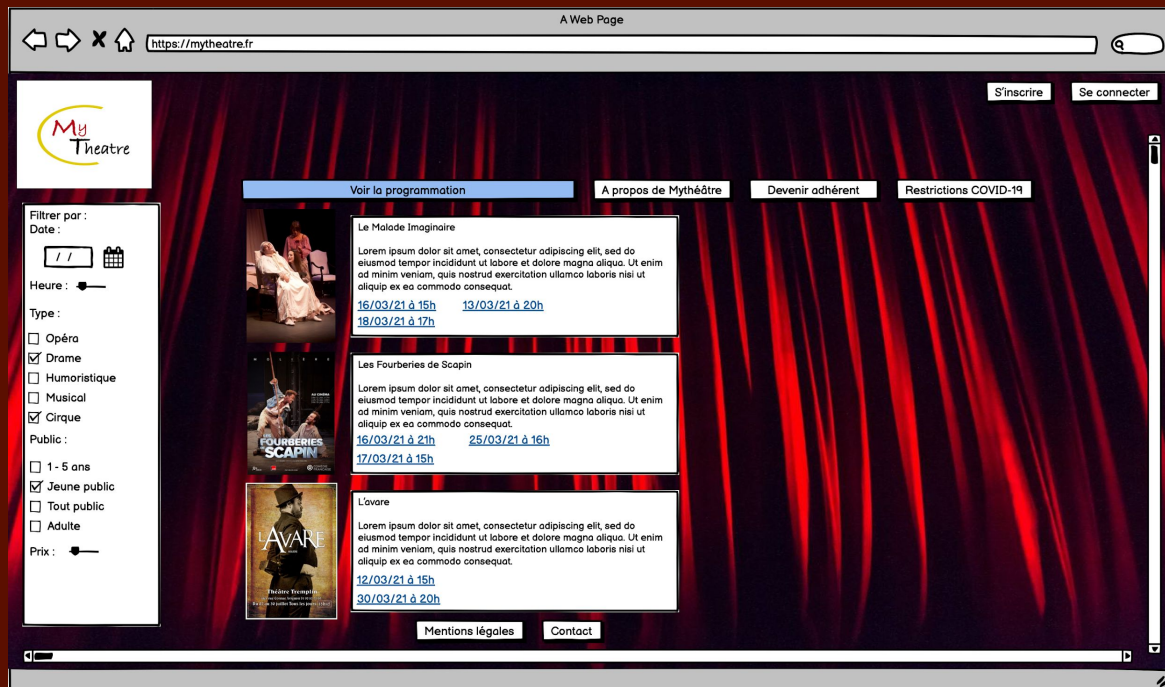
Page d'accueil



Issue #26 : ihm.concrete

Maquettes

Page de la programmation



V. Avancement principal

Issue #26 : ihm.concrete

Heuristiques de Nilsen

- 1 – Visibilité de l'état du système
- 2 – Correspondance du système avec le monde réel
- 3 – Liberté, contrôle de l'utilisateur
- 4 – Cohérence et standards
- 5 – Prévention des erreurs
- 6 – Reconnaître plutôt que se souvenir
- 7 – Flexibilité dans l'utilisation
- 8 – Esthétique et design minimaliste
- 9 – Faciliter l'identification, le diagnostic et la « récupération » des erreurs par l'utilisateur
- 10 – Aide et documentation

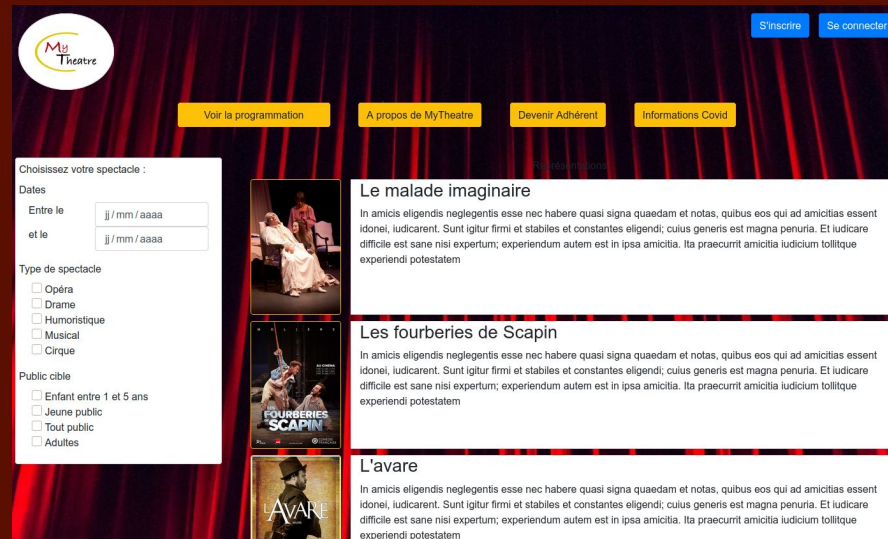
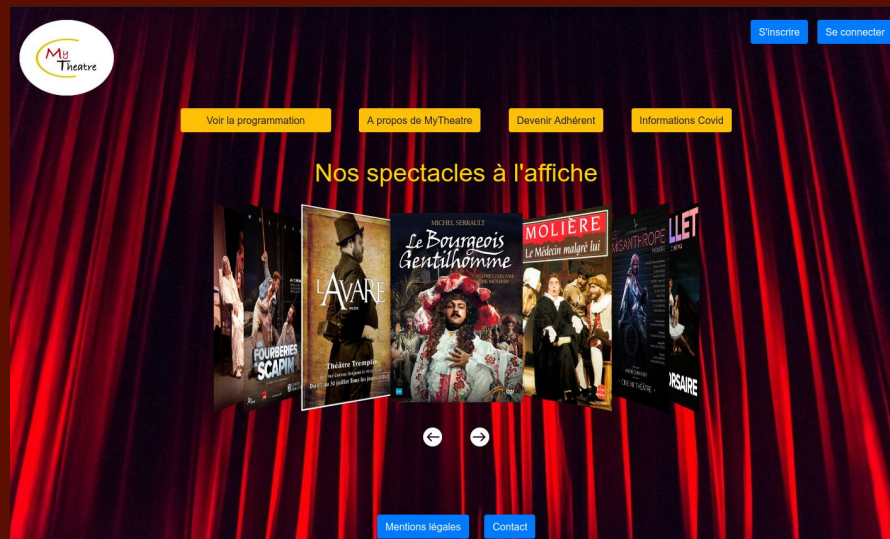
V. Avancement principal

Audit 1

Issue #46 : dev.front.html

Issue #47 : dev.front.css

Prototype HTML/CSS



VI. Prochaines Étapes

VI. Prochaines Étapes

Développement

Site web sur serveur local Tomcat



1. Modèle : implémentation JAVA du diagramme de classe
2. Contrôleur : communication avec la base de donnée
3. Vue : Affichage dynamique