

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

polytech.univ-tours.fr

Projet Recherche & Développement

2017-2018

Développement d'une application WEB permettant de gérer les sorties culturelles des étudiants d'échange

Tuteur académique
Malika LABANE

Étudiant
Saoussen KHELIFI (DI5)

2 avril 2018



Liste des intervenants

Nom	Email	Qualité
Saoussen KHELIFI	saoussen.khelifi@etu.univ-tours.fr	Étudiant DI5
Malika LABANE	malika.labane@univ-tours.fr	Tuteur académique, Département Informatique



Avertissement

Ce document a été rédigé par Saoussen Khelifi susnommé l'auteur.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Malika Labane susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



Pour citer ce document

Saoussen Khelifi, *Développement d'une application WEB permettant de gérer les sorties culturelles des étudiants d'échange*, Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2017-2018.

```
@mastersthesis{
  author={Khelifi, Saoussen},
  title={Développement d'une application WEB permettant de gérer les sorties culturelles
    des étudiants d'échange},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2017-2018}
}
```

Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	iv
Remerciements	1
Remerciements	1
Introduction	2
1 Présentation	2
2 Sujet et objectifs du projet.....	2
3 Plan du rapport	3
1 Contexte du projet	4
1 Présentation	4
2 Bases méthodologiques	5
3 Organisation du travail	5
2 Analyse du sujet	7
1 Principe général de l'application existante	7
1.1 Modélisation de l'existant.....	8
1.2 Fonctionnalités de l'existant.....	9

2	Principe général de l'application cible.....	10
2.1	Environnement du projet	11
2.1.1	Qu'est ce que EXT-JS?	11
2.1.2	Qu'est ce que EclipseLink?	12
2.2	Caractéristiques des utilisateurs	12
2.3	Fonctionnalités du système	13
2.3.1	Fonctionnalité principale : gestion du workflow	14
2.3.2	Fonctionnalité développée : système d'authentification	16
2.3.3	Fonctionnalité principale : consultation	16
2.3.4	Fonctionnalité principale : modification / ajout / suppression d'une sortie	16
2.3.5	Fonctionnalité principale : modification / ajout / suppression du tarif	16
2.3.6	Fonctionnalité principale implémentée : service de paiement.....	17
2.3.7	Fonctionnalité principale : filtrage.....	17
2.3.8	Fonctionnalité secondaire : système de recherche	17
2.3.9	Fonctionnalité secondaire : exportation de liste des participants..	17
2.4	Interface Homme/Machine - IHM	18
3	Diagramme de classes	18
3	Développement de l'application	20
1	Plan de développement	20
1.1	Sprint 1.....	20
1.2	Sprint 2.....	20
1.3	Sprint 3.....	21
1.4	Sprint 4.....	21
1.5	Sprint 5.....	21
2	Configuration du serveur GlassFish	22
3	Classes développées.....	24
4	Schéma d'architecture	27
5	Développement des différentes interfaces.....	28
5.1	Interface utilisateur	28
5.2	Interface gestionnaire	30
5.3	Réception de billets par messagerie	31
6	Tests et validation	33
6.1	Tests d'intégration HTML	33
6.2	Tests non fonctionnels	33
6.3	Tests unitaires	33
6.4	Tests fonctionnels	35

Conclusion	37
Annexes	38
A Spécifications non fonctionnelles	39
1 Sécurité	39
2 Maintenance et évolution du système.....	39
B Conduite du projet	40
1 Planning réel.....	40
C Documents annexes	43
Comptes rendus hebdomadaires	44
Webographie	47

Table des figures

1 Contexte du projet

1	Trello	5
2	GIT - TICKETFAC	6

2 Analyse du sujet

1	Refonte de l'application.....	7
2	Application existante : TICKETFAC	8
3	Modélisation - Existant.....	9
4	Exemple de spectacles	10
5	Environnement du projet	11
6	IntelliJ	12
7	Caractéristiques des utilisateurs	13
8	Modélisation du système	14
9	Workflow du projet	15
10	Diagramme de classes	19

3 Développement de l'application

1	WampServer.....	20
2	Configuration GlassFish.....	22
3	Configuration GlassFish depuis IntelliJ.....	23
4	Configuration en ligne de commande.....	23
5	Configuration de la base de données	24
6	Bout de code	26
7	Page de paiement Paybox	30

8	Ticket de confirmation	32
9	Exemple de Test	34
10	Exemple de Test	35
11	Exemple de Test	36
 B Conduite du projet		
1	Liste des tâches	41
2	Diagramme de Gantt du planning du projet	42



Remerciements

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer ce rapport de mon projet de recherche et développement par des remerciements, à ceux qui m'ont beaucoup appris au cours de ces sept mois de recherche et développement, et même à ceux qui ont eu la gentillesse de faire de ce projet un moment intéressant et agréable.

Je remercie beaucoup Mme. Malika Labane mon encadrante, pour son encadrement efficace, ses précieux conseils et de m'avoir suivi dans mon travail.

Introduction

1 Présentation

Ce rapport vise à aborder l'ensemble des spécifications détaillé du projet « gestion des sorties culturelles des étudiants d'échange ». Il explique les fonctionnalités de l'application développée, et présente sa conception.

Cet outil devra permettre uniquement aux étudiants RI (relations internationales) de consulter les programmes disponibles par semestre, choisir et payer une ou plusieurs activité(s) proposée(s).

Le projet détaillé par ce document est commandé par l'université de Tours, encadré par Mme. Malika LABANE et rédigé par KHELIFI Saoussen.

2 Sujet et objectifs du projet

La tâche qui m'a été confiée est d'intervenir en tant que concepteur et développeur sur le sujet Gestion des sorties culturelles des étudiants RI (Cf. *Analyse du sujet*).

Le but de ce projet est de proposer et développer un outil permettant de gérer les sorties culturelles des étudiants d'échange tout en améliorant l'application TICKETFAC qui nécessite elle-même une refonte et une évolution.

Pour répondre donc aux besoins des RI, il faut :

- Adapter et faire évoluer l'application TICKETFAC,
- L'outil à développer sera accessible à partir de l'application TICKETFAC,
- Concevoir et développer l'outil afin de proposer et gérer les sorties culturelles, Le service culturel doit pouvoir continuer à utiliser l'application existante TICKETFAC pour tout le monde, l'étudiant d'échange peut continuer à acheter des entrées à des spectacles s'il le souhaite.
- Filtrer les étudiants sur le critère d'étudiant d'échange sortant.

La première partie de ce projet est de rédiger le cahier des charges fonctionnel pour bien identifier le besoin du client.

Une fois le CdCf est validé, la prochaine étape consiste à rédiger le cahier de spécification qui explique l'implémentation des fonctionnalités définies dans le cahier des charges et présente sa conception.

Ensuite, il faut faire les maquettes pour l'étude et la visualisation des fonctionnalités de l'application.

La dernière partie décrit le développement de l'application, et explique l'implémentation des différentes fonctionnalités et la communication de la BdD avec le serveur utilisé.

3 Plan du rapport

Ce rapport présente une synthèse de mes travaux effectués durant les sept mois du projet.

Dans un premier temps, je commencerai par décrire le contexte du projet et présenter les bases méthodologiques (chapitre 1).

Dans un second temps, j'expliquerai le principe général du projet, modélisation et fonctionnalité de l'application, etc. (chapitre 2).

Finalement, je présenterai le plan de développement du projet, les tâches effectuées et les résultats obtenus (chapitre 3).

1

Contexte du projet

1 Présentation

L'application existante TICKETFAC permet aux étudiants, personnels ou enseignants de l'Université de Tours d'acheter en ligne des billets pour assister aux spectacles proposés par le service culturel de l'université de Tours.

L'utilisateur peut donc choisir et acheter le spectacle qu'il veut, en effectuant le paiement en ligne soit avec sa CB visa soit avec sa CB masterCard. Les billets sont à retirer soit au service culturel de l'Université pendant les heures d'ouverture, soit directement à la billetterie 15 mn avant le début du spectacle.

L'application comprend une notion de groupe tarifaire.

Le tarif est lié au type de spectacle, il est possible de créer autant de types de spectacles qu'il y a de tarifs différents proposés.

Le service culturel de l'université gère également les sorties qui sont organisées chaque W.E, les étudiants sont informés de toute activité/sortie par liste de diffusion.

Actuellement, il n'existe pas un outil qui permet de gérer automatiquement les sorties culturelles des étudiants RI. Ces étudiants quelque soit leurs nationalités, arrivent par semestre, soit le 1er semestre soit le 2e semestre ce qui explique les périodes de démarrage des sorties qui sont le mois de Septembre (1er semestre) ou le mois de Janvier (2e semestre).

Problématique : « Comment identifier et distinguer uniquement les étudiants RI afin de gérer les sorties culturelles? ».

Pour répondre à cette problématique, la solution la plus fiable est de développer un outil qui ne sera accessible que par ces derniers. Il est donc réalisé uniquement pour les étudiants d'échange (RI) qui représentent 10% des étudiants étrangers.

L'application TICKETFAC propose des billets pour assister aux spectacles alors que la nouvelle application propose des sorties et des activités culturelles pour les étudiants d'échange elle est accessible à partir de l'application TICKETFAC.

2 Bases méthodologiques

Pour la modélisation du système, le choix se portera sur l'utilisation du langage UML (Unified Modeling Language), cela permet d'avoir une modélisation formelle et compréhensible par l'ensemble des intervenants sur le projet.

Pour la partie développement, les langages de programmation utilisés sont : HTML5/CSS3/JS, Java.

Les sources du logiciel seront versionnées grâce à l'outil GIT qui permet de conserver les différentes versions.

La méthode agile est utilisée dans ce projet, grâce à elle le demandeur obtient une meilleure visibilité de la gestion des travaux qu'avec une méthode classique.

L'implication du client dans le processus permet d'obtenir un feedback régulier afin d'appliquer directement les changements nécessaires.

3 Organisation du travail

Lors de mon premier jour de projet, j'ai créé un Trello pour noter les étapes de mon travail, les tâches à faire ainsi que les tests et les résultats obtenus.

Dans le Trello, j'ai créé des cartes : « À faire » où je mets les tâches à réaliser, une carte « En cours » où je mets les tâches que je suis entrain de les faire, une autre carte « Terminé » où je mets les tâches effectuées.

Pour partager mon code avec mon encadrante, Mme. Malika Labane m'a envoyé le code existant disponible dans GIT « TICKETFAC » pour pouvoir apporter des modifications et pour qu'elle puisse suivre mon avancement. Le travail à effectuer sera disponible sur la branche « dev » que j'avais créée.

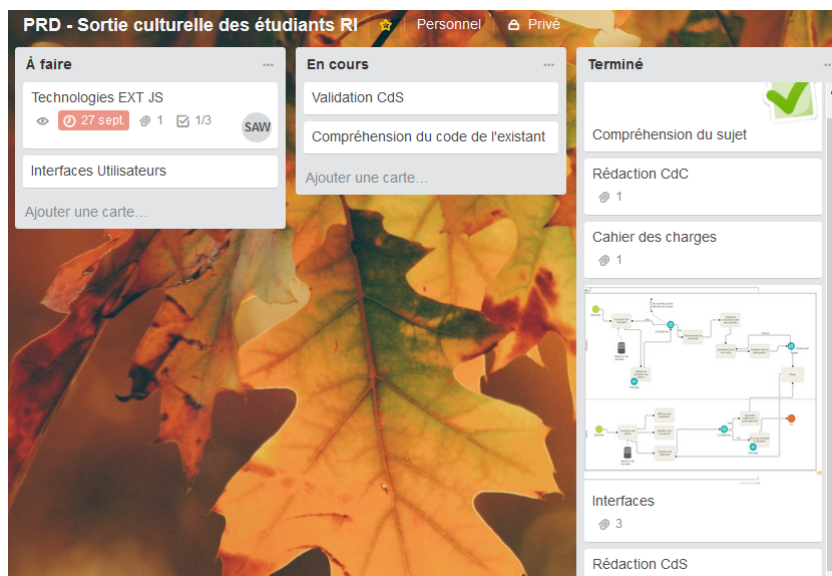



Figure 1 – Trello

Malika Labane > ticketfac > Repository

dev ticketfac / + History Find file

 Paybox : mise en conformité pour utiliser une clé HMAC - correction retour paiement
Malika Labane committed a month ago 7e73d1df

Name	Last commit	Last Update
JTadmin	Module invitations : sauf affichage de la liste des ré...	9 months ago
src	Paybox : mise en conformité pour utiliser une clé H...	a month ago
101105_Sourd_Malentendant.jpg	modifications finales	3 years ago
README.md	initial commit	4 years ago
log4j.properties	correction bugs 2 emails + exports spectacles	3 years ago
pom.xml	changement version : 1.0.0 -> 1.1.0	8 months ago

Figure 2 – GIT - TICKETFAC

2

Analyse du sujet

1 Principe général de l'application existante

L'application TICKETFAC permet à tout utilisateur de visualiser les spectacles à venir en précisant toute information relative à ce spectacle telle que la date, une présentation des artistes, le tarif de chaque catégorie (tarifs personnel, tarif étudiant sans PCE*, plein tarif, tarif réduit, tarif partenaire, tarif étudiant avec PCE, tarif universitaire de temps libre).

L'utilisateur peut donc choisir et acheter le spectacle qu'il veut en effectuant le paiement en ligne soit avec sa CB visa soit avec sa CB masterCard. Les billets sont à retirer soit au service culturel de l'Université pendant les heures d'ouverture, soit directement à la billetterie 15 mn avant le début du spectacle.

L'application Ticketfac existante permet au service culturel de gérer ses spectacles via un module d'administration (développé en EXT JS) et à tout le monde (pas uniquement les gens de l'université de Tours) d'acheter des tickets en ligne via une application web.

Le spectacle est donc associé au type qui correspond au bon tarif. Pour chaque spectacle il existe une jauge, une date de début et une date de fin pour s'inscrire.

L'accès au site propose plusieurs types d'authentification :

- le CAS pour ceux qui font partie de l'université de Tours (identifiant et mot de passe identiques à l'ENT),
- création d'un compte avec login (adresse mail) et mot de passe pour les extérieurs à l'université.

Refonte :

Refonte d'un site existant	Raison de la refonte
Oui	Car certains besoins ne répondent pas aux besoins des RI. Il faut donc faire évoluer l'application TICKETFAC.

Figure 1 – Refonte de l'application

Comment utiliser le service[WWW4] ?

Pour tous les étudiants de l'université de Tours ou pour les extérieurs à l'université qui veulent acheter de billet de spectacle à venir, ils doivent :

- accéder à la plateforme « TICKETFAC » depuis « l'ENT » ou depuis le site web,¹ en se connectant avec le CAS ou avec un login et mot de passe choisis lors de l'inscription,
- choisir un spectacle et se connecter à la plateforme pour payer.



Figure 2 – Application existante : TICKETFAC

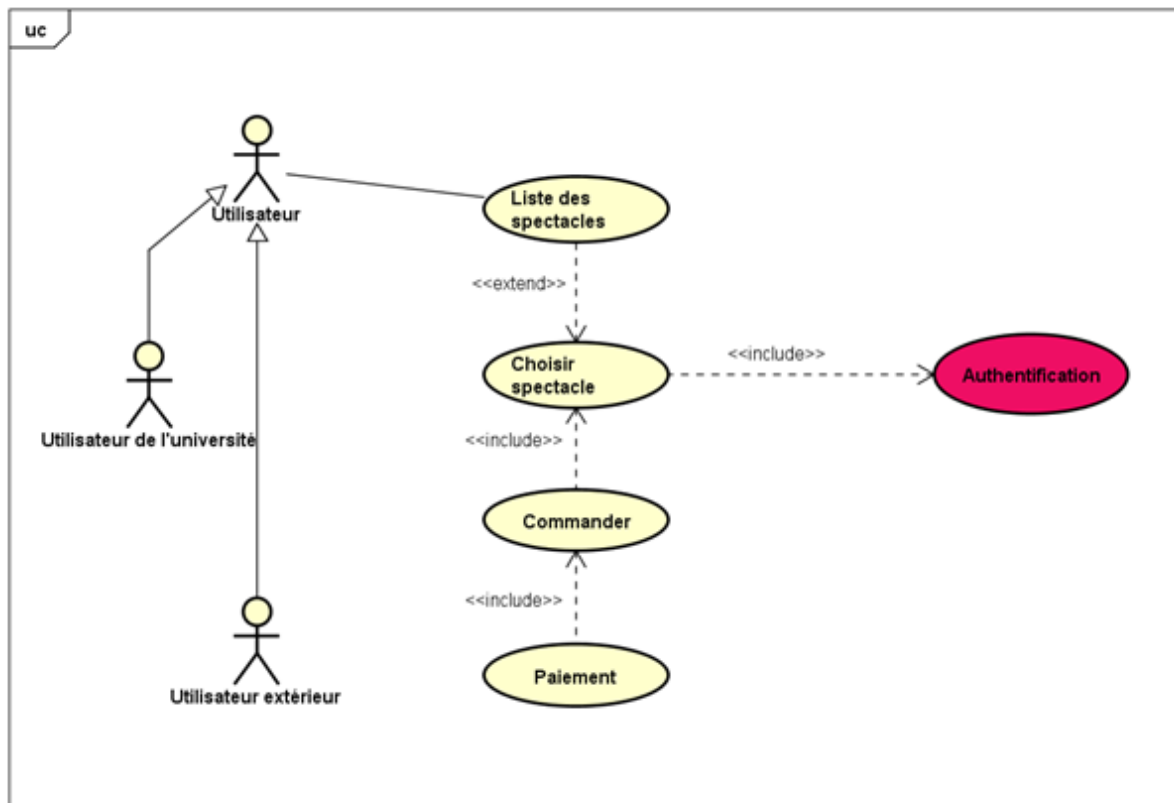
Quelles informations peut-on trouver dans cette plateforme ?

1. Nom du spectacle,
2. Description du spectacle,
3. Date et heure de son passage,
4. Tarifs différents : tarif étudiant sans PCE, tarif université de temps libre, tarif étudiant avec PCE, tarif personnels, tarifs réduits, plein tarif,
5. Les acteurs.

1.1 Modélisation de l'existant

Ce diagramme donne une vision globale de l'utilisation de l'application TICKETFAC.

1. <https://ticketfac.univ-tours.fr/>



powered by Astah

Figure 3 – Modélisation - Existant

1.2 Fonctionnalités de l'existant

1. Authentification : pour pouvoir commander et acheter les billets du spectacle, l'utilisateur doit se connecter avec un couple (login, mdp), cette page assure l'accès sécurisé à l'outil. Deux types d'utilisateurs sont gérés :
 - utilisateurs de l'université : doivent s'identifier avec leur compte « ENT »,
 - utilisateur extérieur : Identification avec les réseaux sociaux tels que « Facebook » ou inscription avec une adresse mail personnelle.
 Une fois l'authentification est validée, l'utilisateur aura l'accès à l'interface qui gère les achats.
2. Liste des spectacles : cette page est accessible à tout le monde pour pouvoir visualiser les spectacles proposés.

Conjurer la peur - Gaëlle Bourges
 le 05/12/2017 à 20:30 [thélème](#)
 Voir la danse en peinture c'est possible avec Gaëlle Bourges ! [Plus d'infos...](#)
 Twitter G+
Tarifs
 Tarif étudiant sans PCE 6.0 €
 Tarif université de temps libre 9.0 €
 Tarif étudiant avec PCE 4.0 €
 Tarif personnels 6.0 €
 Tarifs réduits 6.0 €
 Plein tarif 12.0 €
 Acheter

Spectacle réservé aux personnels de l'Université - Play War - Compagnie Discrète
 le 15/12/2017 à 20:30 [thélème](#)
 Un spectacle de mime à voir en famille ! Tout public (à partir de 7 ans) [Plus d'infos...](#)
 Twitter G+
Tarifs
 Tarifs réduits 0.0 €
 Tarif personnels 0.0 €
 Acheter

Carole Marchais, artiste plasticienne en résidence à l'Université - Rencontre et vernissage
 le 12/12/2017 à 18:30 [maison des sciences de l'homme](#)
 Au terme d'un premier semestre d'immersion dans l'Université, Carole Marchais vous propose de découvrir une première

Figure 4 – Exemple de spectacles

3. Chercher / Choisir un spectacle / Commander : l'utilisateur peut effectuer une recherche sur un spectacle qu'il veut, il ne peut choisir un spectacle que s'il est connecté, pour qu'il puisse effectuer sa commande et son achat.

2 Principe général de l'application cible

La place de la culture dans la vie étudiante

La culture[WWW1] et la découverte de la ville pour les étudiants étrangers occupent une place très importante dans leur vie.

Depuis des années, les activités culturelles jouent un rôle de plus en plus important en Europe. Pour la plupart de ces étudiants, les objets comme *la visite des expositions, des musées, des châteaux, etc.* sont devenus des éléments essentiels de leur style de vie et de leur conscience collective.

En Europe, les villes ont toujours été les principaux centres de l'innovation et du développement culturel. C'est là que, depuis le Moyen Âge, naissent tous les nouveaux courants de la culture européenne; et que, de tout temps, les étudiants français et les habitants sont les plus enclins à découvrir les nouvelles tendances et à y participer.

Depuis des siècles, les villes ont donc mis en œuvre de véritables politiques culturelles, en commençant par la construction d'équipements tels que les théâtres ou les bibliothèques publiques. Elles se sont également attachées à favoriser l'accès à la culture, notamment par l'éducation.

Qu'apportent les sorties culturelles?

Les activités culturelles apportent de l'information, des compétences et des capacités existantes dans tous les secteurs, et indispensables à l'épanouissement de chacun.

Réduire les activités culturelles à des instruments au service de l'attractivité serait toutefois une erreur. Elles peuvent également jouer un rôle déterminant en faveur du développement durable des quartiers, en permettant d'améliorer la confiance que les étudiants ont en eux-mêmes et leur sentiment d'appartenance à la ville.

Comme la culturelle est le propre de l'homme et l'essence de la vie, il faut donc donner la chance aux étudiants étrangers et particulièrement aux étudiants d'échange de découvrir et participer à des sorties organisées par le service culturel de l'université de Tours. Le service culturel propose également l'application TICKETFAC décrite en dessus pour proposer des billets de spectacles à tous les étudiants, des enseignants et des personnels.

L'application cible sera accessible à partir de l'application de Billetterie existante, et ne sera qu'à la portée des étudiants RI. Cela définit le besoin principal de service culturel.

2.1 Environnement du projet

Pour développer l'outil, voici l'ensemble des langages, des logiciels et des frameworks utilisés.



Figure 5 – Environnement du projet

2.1.1 Qu'est ce que EXT-JS?

Ext-JS[WWW3] est le framework de choix pour développer de puissantes applications de bureau sur le web en utilisant JavaScript et les standards du web.

Il est à l'origine une branche de la bibliothèque JavaScript développée par Yahoo! sous l'appellation YUI.

Dès sa version 1.0, Jack Slocum, cherche à en faire un produit industrialisé. Ext-JS s'est ainsi rapidement retrouvé au rang des meilleures bibliothèques JavaScript Open Source.

2.1.2 Qu'est ce que EclipseLink ?

EclipseLink^[WWW2] est un framework open source de mapping objet-relationnel pour les développeurs Java.

Il fournit une plateforme puissante et flexible permettant de stocker des objets Java dans une base de données relationnelle et/ou de les convertir en documents XML.

EclipseLink est dérivé du projet TopLink de la société Oracle.

Il supporte un certain nombre d'API relatives à la persistance des données et notamment la JPA.

Le choix des langages de programmation, des frameworks et de base de données est exigé par Mme. Malika.

Pourquoi choisir IntelliJ? Quels sont les avantages et les inconvénients ?

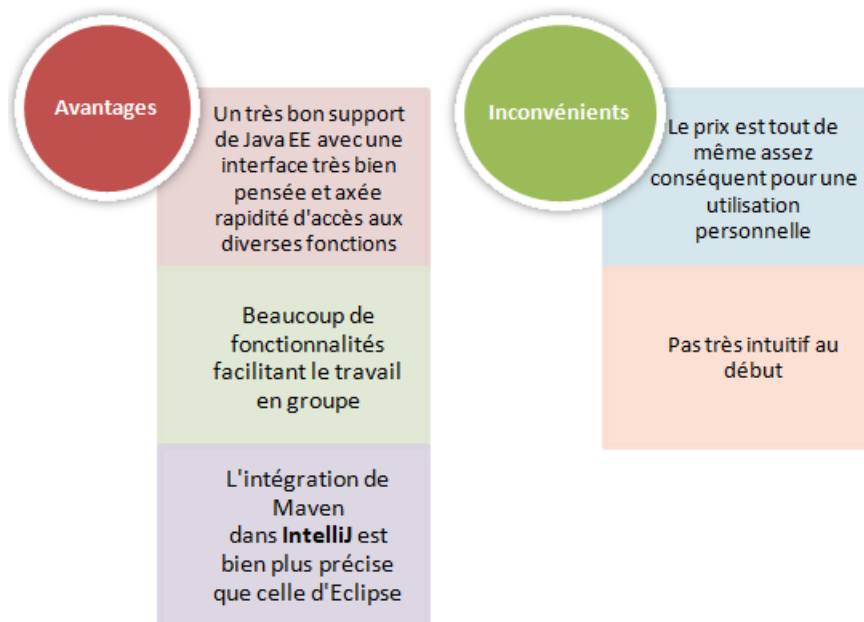


Figure 6 – IntelliJ

2.2 Caractéristiques des utilisateurs

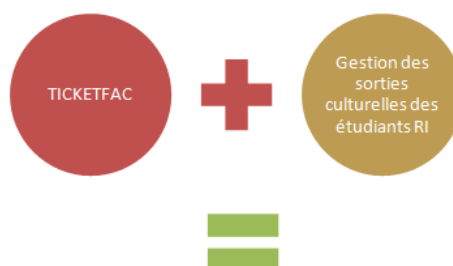
On peut différencier trois types d'acteurs qui auront chacun des autorisations différentes sur la plateforme web : les types des utilisateurs, les caractéristiques de chaque type sont présentés comme le tableau suivant.

Type des utilisateurs	Description	Connaissance ou non de l'informatique	Utilisateurs réguliers ou occasionnels
Etudiants (non RI)	Ces étudiants ne peuvent pas payer ou participer à des sorties proposées.	Non	occasionnels
Etudiants RI	Ces étudiants ont le droit de choisir, participer et payer une sortie ou une activité proposée par la DRI.	Non	occasionnels
Administrateur	Les personnes qui gèrent les sorties culturelles, les tarifs et les droits de l'utilisateur.	Oui	réguliers

Figure 7 – Caractéristiques des utilisateurs

2.3 Fonctionnalités du système

Pour décrire les fonctionnalités de tout le système, voici son diagramme de cas d'utilisation.



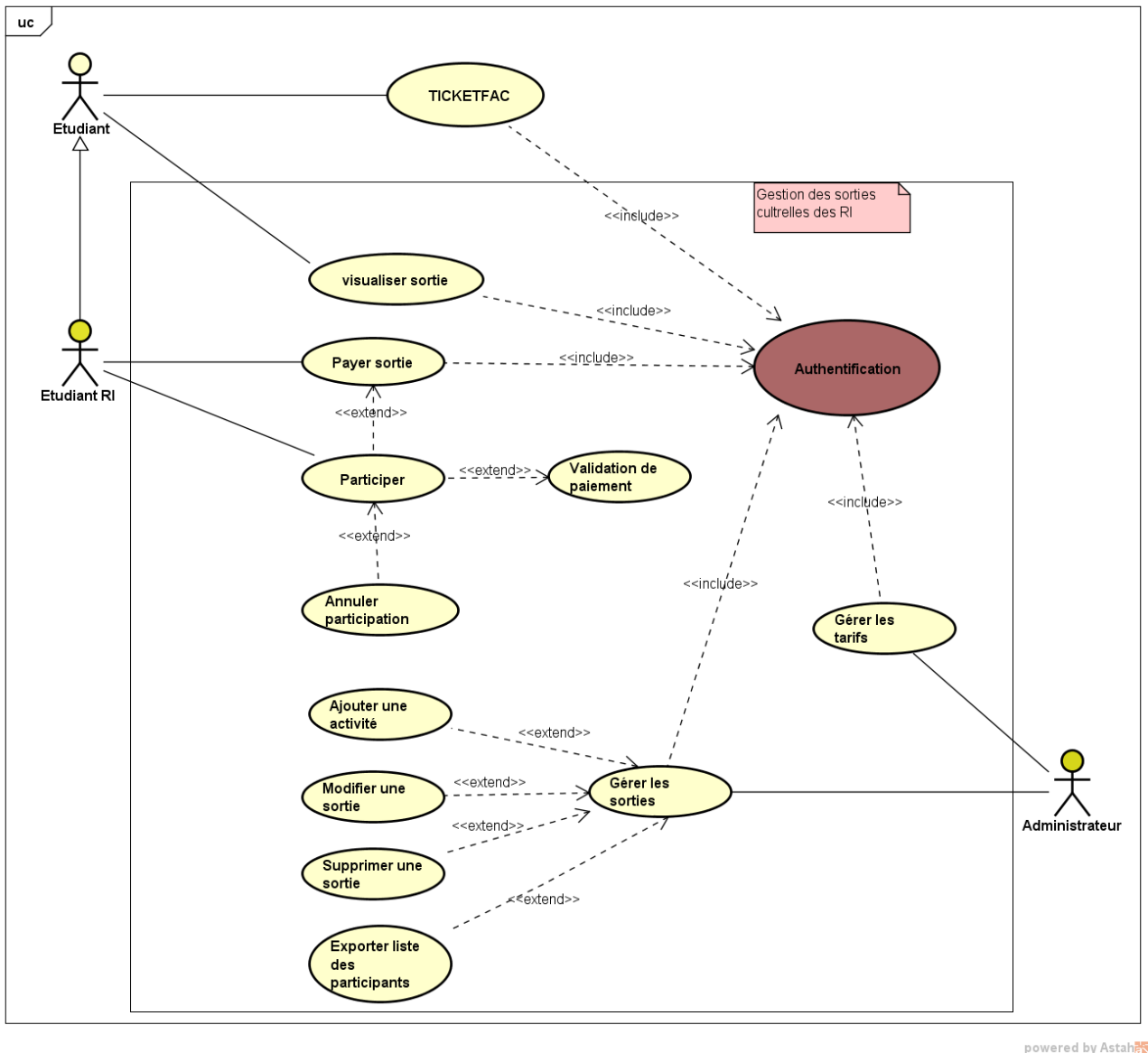


Figure 8 – Modélisation du système

Un étudiant peut être un étudiant RI, donc il peut accéder à la Billetterie pour acheter des billets de spectacles et aussi à l'application de gestion des sorties culturelles pour participer à des sorties culturelles. Il peut visualiser la liste des sorties culturelles proposées par la DRI et effectuer son achat en ligne mais celle-ci (la participation) ne sera pas validée que pour les étudiants RI.

2.3.1 Fonctionnalité principale : gestion du workflow

Gestion du workflow d'un projet

Ce workflow permet de présenter toutes les tâches nécessaires aux sorties culturelles des étudiants d'échanges.

Éléments du workflow pour un étudiant :

- consulter les sorties,

- choisir une sortie,
- valider sa participation : l'étudiant peut annuler ou valider son choix.
S'il valide une activité alors il doit payer le montant indiqué. Sinon il annule et effectue un autre choix.

Éléments du workflow pour l'administrateur du site :

- connexion à la plateforme,
- le gestionnaire effectue les opérations suivantes : ajouter / modifier / annuler une sortie,
- le gestionnaire effectue les opérations suivantes : ajouter / modifier / supprimer un tarif d'une sortie,
- validation de paiement : si le paiement est validé, un justificatif de paiement sera envoyé à l'utilisateur. Sinon, le paiement sera refusé si l'étudiant n'est pas un étudiant RI.

De multiples opérations de modification pourront être faites sur les sorties. Ainsi on pourra, modifier :

- la date de sortie.

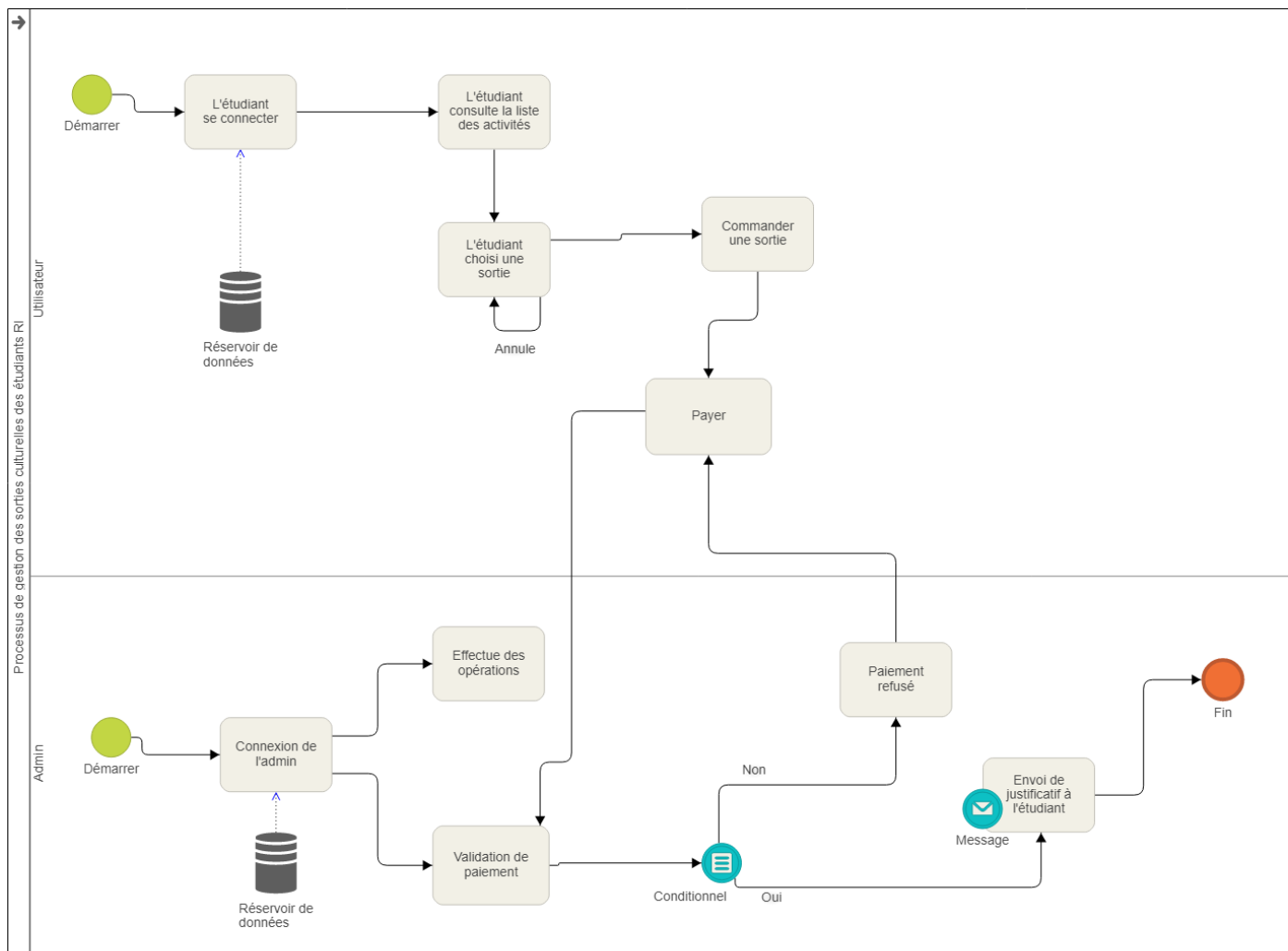


Figure 9 – Workflow du projet

2.3.2 Fonctionnalité développée : système d'authentification

Pour agir sur les activités proposées, il est nécessaire de s'authentifier avec CAS. Cette fonctionnalité est déjà développée, il faut juste l'implémenter dans l'outil.

Description :

- on s'authentifie à la plateforme via un couple login/mot de passe. Le login est l'adresse mail ou le numéro de l'étudiant.

Qu'est ce que CAS[WWW5]?

Le Central Authentication Service (CAS)² est un système d'authentification unique (SSO) pour le web développé par Shawn Bayern de l'Université Yale, partenaire majeur dans le développement de uPortal.

Ce logiciel est implanté dans plusieurs universités et organismes dans le monde.

Il est donc un système d'authentification unique : on s'authentifie sur un site Web, et on est alors authentifié sur tous les sites Web qui utilisent le même serveur CAS. Il évite de s'authentifier à chaque fois qu'on accède à une application en mettant en place un système de ticket.

2.3.3 Fonctionnalité principale : consultation

Cette fonction permet aux étudiants et également aux gestionnaires de consulter les sorties disponibles.

Description :

- la consultation des sorties est accessible à tous les utilisateurs possédant le droit d'accès à la plateforme.
- la consultation comprend les différentes informations sur une activité telle que :
 - nom de la sortie,
 - date et heure de la sortie,
 - tarif de la sortie : tarif plein, tarif sans PCE et tarif avec PCE.

2.3.4 Fonctionnalité principale : modification / ajout / suppression d'une sortie

Cette fonctionnalité permet aux administrateurs de la plateforme de gérer les sorties.

Description :

- le gestionnaire a le droit d'ajouter une nouvelle sortie ou une nouvelle activité destinée aux étudiants RI en insérant les informations liées à cette sortie.
- le gestionnaire a le droit de modifier les informations sur une sortie. Les étudiants seront informés de toute modification apportés.
- le gestionnaire a le droit également de supprimer ou annuler une sortie.

2.3.5 Fonctionnalité principale : modification / ajout / suppression du tarif

Cette fonctionnalité permet aux administrateurs de la plateforme de gérer les tarifs de chaque sortie.

Description :

- le gestionnaire a le droit d'ajouter un nouveau tarif pour chaque sortie, en indiquant le libellé du tarif (tarif avec ou sans PCE),

2. <https://fr.wikipedia.org/wiki>

- le gestionnaire a le droit de modifier le tarif. Les étudiants seront informés de toute modification apportée.
- le gestionnaire a le droit également de supprimer les informations liées à un tarif donné.

2.3.6 Fonctionnalité principale implémentée : service de paiement

Le module de paiement existe et devra juste être légèrement adapté pour les RI.

Ce service permet aux étudiants qui ont choisi une sortie de payer le montant indiqué pour pouvoir participer.

Description :

Lorsqu'un étudiant choisit une sortie, il est nécessaire de payer pour qu'il puisse participer.

Pour participer à une sortie, il faut que l'étudiant valide et paie son choix. Pour cela il faut intégrer un système de paiement sécurisé.

Le payeur communique ses coordonnées :

- nom,
- coordonnées bancaires :
 - le numéro à 16 chiffres de sa carte,
 - les 4 chiffres de la date de validité (mois et année),
 - les 3 derniers chiffres du pictogramme au dos de la carte de paiement.

2.3.7 Fonctionnalité principale : filtrage

Présentation :

- filtrer les étudiants sur le critère « étape » : Lorsqu'un étudiant s'inscrit à l'université, il est affecté à une « étape » (ex : 5^{ème} année du DI de l'EPU). Les étudiants d'échange sont inscrits dans différentes filières (différentes étapes) avec des étudiants « normaux » qui ont donc la même étape,
- filtrer les étudiants sur le critère d'étudiant d'échange sortant.

2.3.8 Fonctionnalité secondaire : système de recherche

Les utilisateurs peuvent rechercher les sorties par mot-clés, par date, ou par le nom de la sortie.

2.3.9 Fonctionnalité secondaire : exportation de liste des participants

Cette fonctionnalité permet d'exporter la liste des étudiants qui ont participé.

Description :

- le gestionnaire peut exporter la liste des participants sous format EXCEL.
- le fichier comprend les informations utiles des participants.

2.4 Interface Homme/Machine - IHM

Les différents utilisateurs autorisés pourront avoir accès à la plateforme. Une gestion de l'authentification devra être mise en place pour différents types d'utilisateurs ayant des droits différents sur la plateforme.

En fonction du type d'utilisateur, l'interface peut être différente.

En matière d'ergonomie du système :

- **organisation** : les différentes opérations telles que : consultation des sorties/ajout des sorties, etc. L'utilisateur n'a pas le droit de modifier ou supprimer une sortie donc ces opérations ne seront pas affichées lorsqu'un utilisateur ou un étudiant se connecte à la plateforme, elles seront affichées que lorsque l'administrateur du site est connecté.
- **couleurs** : l'application prend en compte le traitement de la couleur.
- **fiabilisation** : pour fiabiliser l'application, une gestion des erreurs sera mise en place (Alertes JavaScript).

Au passage de la souris sur la rubrique « Sorties culturelles » la nouvelle plateforme s'affiche. Si l'utilisateur a été connecté (depuis l'application Ticketfac) alors il peut accéder à la liste des sorties proposées chaque WE et peut effectuer l'achat en ligne (avant la date de la sortie) avec une CB visa ou mastercard.

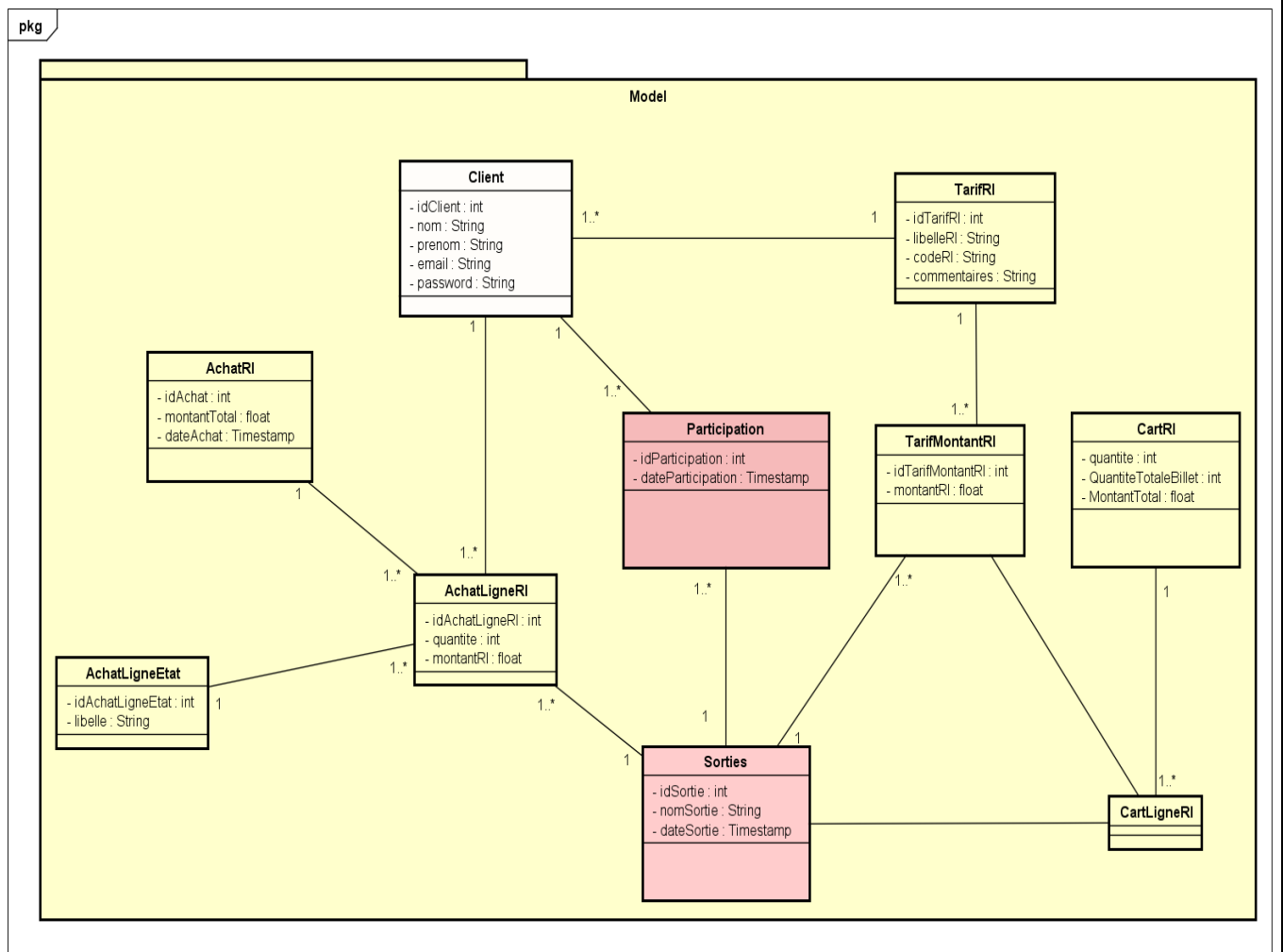
Sinon, il est possible de se connecter depuis la nouvelle plateforme.

L'outil comprend également le côté « Admin » pour modifier, ajouter et supprimer les sorties ou les tarifs des sorties. L'admin a le droit également de filtrer la liste des étudiants d'échange.

Ces pages ne seront visibles que par les administrateurs de l'application.

Pour voir les différentes pages développées, Cf. **Développement des interfaces**.

3 Diagramme de classes



powered by Astah

Figure 10 – Diagramme de classes

3

Développement de l'application

1 Plan de développement

Ce projet utilise la méthode agile pour définir les tâches et faire des retours réguliers. Chaque livrable correspond à un ensemble de tâches effectuées. Voici les différentes tâches du projet :

1.1 Sprint 1

Configurer le serveur pour pouvoir accéder à la plateforme, mettre en place l'environnement de travail, ainsi que la base de données. Pour chacune de ces phases, une documentation technique et utilisateur seront fournis.

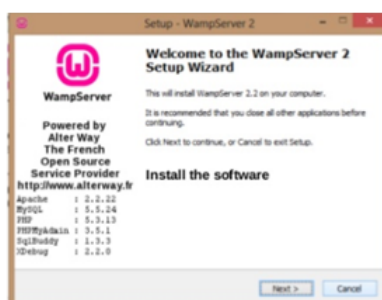


Figure 1 – WampServer

1.2 Sprint 2

Développer les différentes interfaces :

- interfaces pour le gestionnaire,
- interfaces pour l'utilisateur,
- interface d'accueil,

Cette tâche permet de mettre en place les interfaces principales pour réagir avec l'application et créer les classes nécessaires en Java.

1.3 Sprint 3

Développer les fonctionnalités principales du gestionnaire pour gérer les sorties :

- ajouter une sortie, ajouter une date de sortie et des informations relatives aux activités proposées,
- mettre à jour une sortie, sa date ou la deadline,
- filtrer selon différents critères,
- supprimer ou annuler une sortie en informant les participants de chaque nouvelle modification,
- ajouter, modifier ou supprimer un tarif.

Ces fonctionnalités regroupent donc la gestion de la base de données.

1.4 Sprint 4

Développer les fonctionnalités principales de l'utilisateur :

- consulter les activités disponibles sur la plateforme,
- choisir une sortie et un tarif,
- participer à une sortie en effectuant la commande et le paiement en ligne.

Ce sprint permet également de mettre en place un système de paiement sécurisé. Ces fonctionnalités regroupent la gestion de la base de données.

1.5 Sprint 5

Ce sprint regroupe les fonctionnalités secondaires :

- export de la liste des participants de chaque sortie,
- système de recherche.

2 Configuration du serveur GlassFish

Pour reprendre le projet ou exécuter l'application il faut installer ces outils :

1. IntelliJ IDEA,
2. Dernière version du serveur GlassFish (v4.1.2) pour le déploiement de l'application,
3. WampServer pour lancer la base de données MYSQL.

Pour configurer le serveur GlassFish il faut lancer GlassFish soit en ligne de commande, soit depuis IntelliJ.

Depuis IntelliJ :

1. Ouvrir l'interface Run/Debug Configuration,
2. Cliquer sur le bouton + pour créer une nouvelle configuration,
3. Chercher GlassFish, puis cliquer sur Local comme suit :

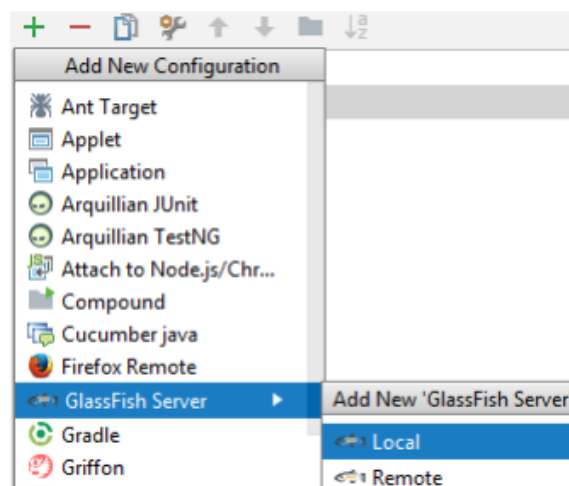


Figure 2 – Configuration GlassFish

4. Ensuite, il faut configurer le serveur en ajoutant le chemin vers le serveur et le port client Http :

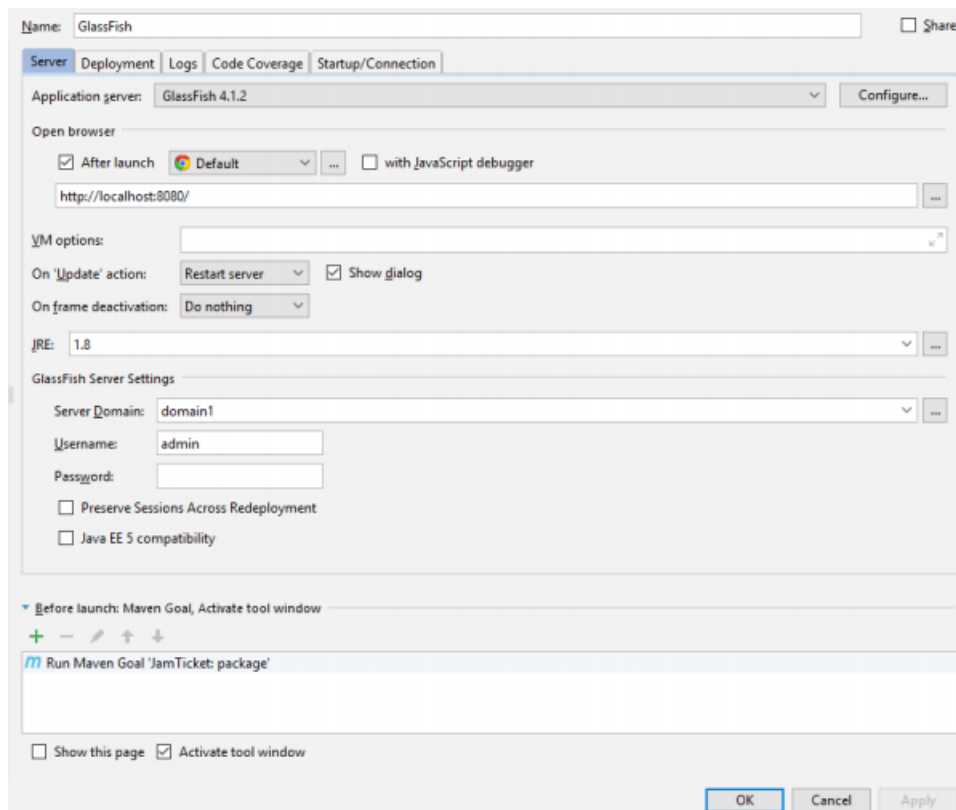


Figure 3 – Configuration GlassFish depuis IntelliJ

En ligne de commande :

1. Il faut ouvrir l'invite de commande en tant qu'administrateur,
2. Accéder au dossier GlassFish et lancer la commande suivante :

```
C:\glassfish4>cd bin
C:\glassfish4\bin>asadmin start-domain domain1
Waiting for domain1 to start .....
Successfully started the domain : domain1
domain Location: C:\glassfish4\glassfish\domains\domain1
Log File: C:\glassfish4\glassfish\domains\domain1\logs\server.log
Admin Port: 4848
Command start-domain executed successfully.
```

Figure 4 – Configuration en ligne de commande

Domain1 est le domaine par défaut.

La commande est lancée avec succès. Le port indiqué est le port **4848** pour accéder au GlassFish console en local.

Depuis la console Admin de GlassFish, il faut configurer **la partie JDBC** :

- JDBC Connection Pools : il faut indiquer le nom de la base de données et le type de la ressource, dans ce projet la ressource est de type *java.sql.Driver*,
- JDBC Resources : configurer l'unité de persistance utilisée dans les classes EJB et l'associer au nom du **Pool** créé.

Configuration de la base de données

Pour pouvoir utiliser et exécuter l'application en tant que **développeur**, il faut sélectionner le profil **dev** depuis l'interface **Maven Project**.

A partir de l'interface **Database** de IntelliJ, il faut préciser le nom de la base de données et les coordonnées de l'utilisateur comme suit :

The screenshot shows the 'Database' configuration window in IntelliJ. The 'Name' field is 'jamticket@localhost'. The 'Host' is 'localhost' and the 'Port' is '3306'. The 'Database' is 'jamticket', the 'User' is 'root', and the 'Password' is masked with asterisks. The 'URL' is 'jdbc:mysql://localhost:3306/jamticket'. The 'Driver' is 'MySQL'. There is a 'Test Connection' button and a 'Remember password' checkbox which is checked.

Figure 5 – Configuration de la base de données

3 Classes développées

Les classes modèles développées sont :

- **Classe Sorties** : Classe contenant les informations relatives à une sortie telles que :
 - Nom de la sortie,
 - date de la sortie,
 - etc.
- **Classe Participation** : Classe contenant les informations de participants (clients qui ont participé à une ou plusieurs sortie(s)).
 - Elle appelle la classe « Client » et la classe « Sorties ».
- **Classe TarifMantantRI** : Classe permettant de gérer le montant de chaque sortie, le montant d'une sortie pour un étudiant possédant un PCE et différent de celui sans PCE.
- **Classe TarifRI** : Classe contenant le code et le libellé du tarif
 - Libellé : Tarif avec PCE ou sans PCE.
- **Classe AchatLigneRI** : Classe contenant les informations relatives aux achats effectués par les clients, elle fait donc appelle aux classes :
 - Client,

- Sorties,
- TarifMontantRI,
- AchatLigneEtat.

- **Classe CartLigneRI** : Cette classe permet la gestion du panier : Elle récupère la quantité et le tarif de la sortie choisie, elle fait donc appelle aux classes :
 - TarifMontantRI,
 - Sorties.
- **Classe CartLigneRI** : Cette classe permet de gérer le panier. Elle récupère la quantité et le tarif de la sortie choisie.
Elle fait donc appelle aux classes :
 - TarifMontantRI,
 - Sorties.

Les classes développées pour l'application TicketFac et utilisées dans l'outil que je développe sont :

- **Email** : Classe modèle contenant les informations relatives à un email telles que :
 - Le sujet,
 - Le texte.
- **Client** : Classe modèle contenant les informations relatives à un client telles que :
 - Nom,
 - Prénom,
 - Mot de passe,
 - Email,
 - Etc.
- **Properties** : Cette classe permet de récupérer les informations pour pouvoir envoyer des emails :
 - Smtip port,
 - Smtip server,
 - SmtipEmailSender,
 - AdminEmail.
- **AchatLigneEtat** : Cette classe permet de suivre l'état de l'achat ou du paiement.

4 états possibles :

1. En cours,
 2. Annulé,
 3. Refusé,
 4. Traité.
- **ClientEJB** : permet de récupérer le client dans la base à partir de son id, son email, son uid, ajouter ou supprimer le client, etc.
 - **EmailEJB** : pour envoyer des emails aux gestionnaires du site en cas de paiement et pour envoyer un email aux clients pour confirmer leurs achats.
 - **ToJsonConvert** : permet de convertir et sauvegarder les données du panier dans un fichier JSON.

Les classes EJB prennent en charge un certain nombre de traitements tel que la gestion des transactions et la persistance des données. Voici les classes développées :

- **Classe SortiesEJB** : C'est la classe qui gère l'accès aux données et qui implémente les méthodes CRUD (Create, Read, Update et Delete)
 - AddSortie() : permet l'ajout d'une sortie,
 - GetAllSorties() : retourne une liste de sorties,
 - EditSortie() : pour modifier une sortie,
 - RemoveSortie() : pour supprimer une sortie de la base de données.

```

@Stateless
public class SortiesEJB {

    @PersistenceContext(unitName = "JamPu")
    protected EntityManager em;
    EmailEJB emailEJB;
    ParticipationRI part;

    /**
     * <b> addSortie </b>
     * @see Sorties
     * @param sortie
     */
    public void addSortie(Sorties sortie)
    {
        em.persist(sortie);
    }

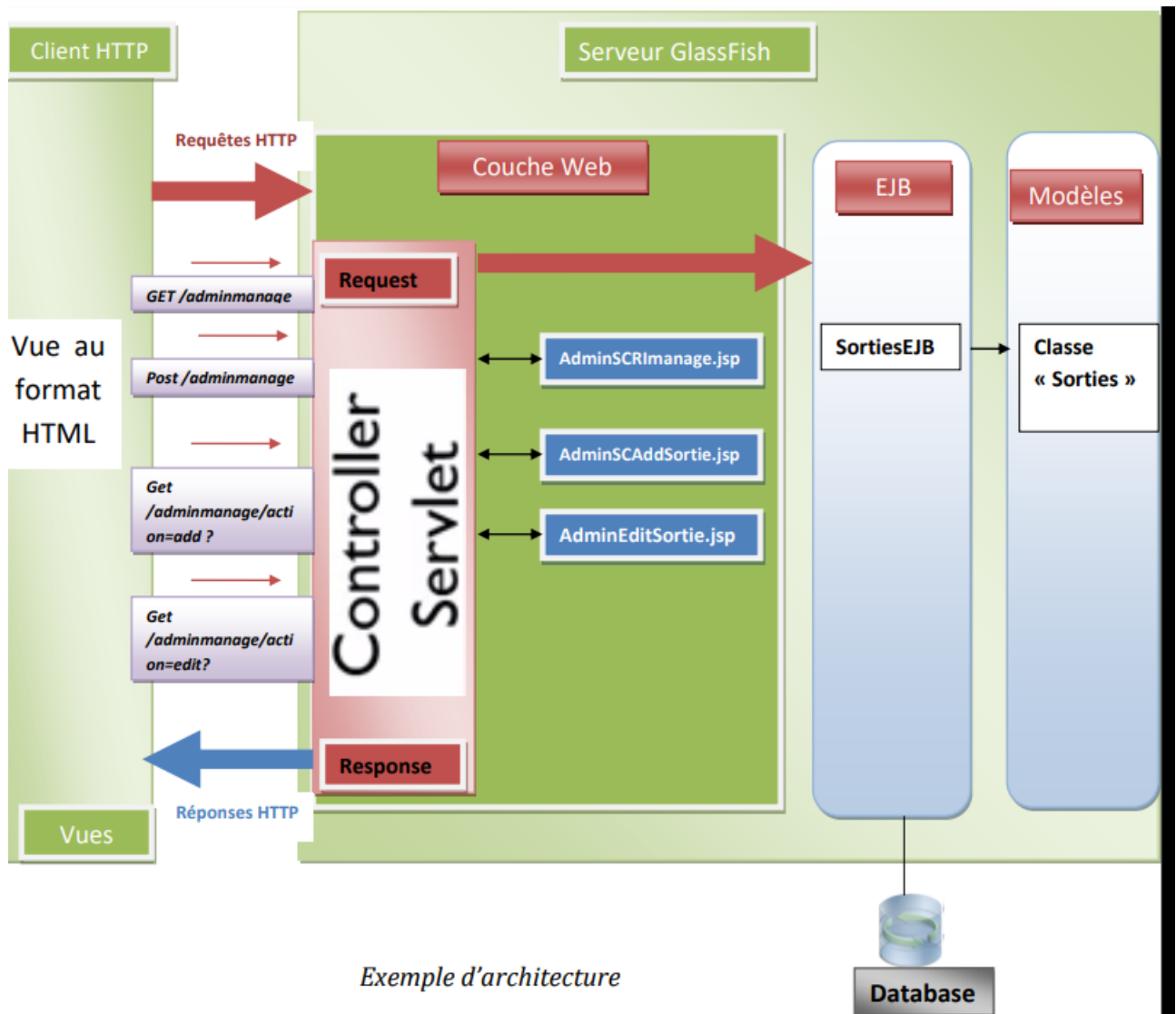
    /**
     * <b> editSortie</b>
     * @see Sorties
     * @param sortie
     */
    public void editSortie(Sorties sortie)
    {
        em.merge(sortie);
    }
}

```

Figure 6 – Bout de code

- **Classe AchatEJB** : C'est la classe qui gère l'accès aux données, elle permet de récupérer la liste d'achat de chaque client, l'état de paiement (annulé, refusé, traité ou en cours) et sauvegarder les données du panier.
- **CartRIEJB** : permet d'ajouter dans le panier la quantité souhaitée, de changer les données du panier et de supprimer du panier la quantité ajoutée.
- **ParticipationEJB** : permet de récupérer la liste des clients ayant participé à une ou plusieurs sortie(s) à l'aide d'une requête SQL.
- **TarifRIEJB** : permet de récupérer la liste de tarifs et implémente les méthodes CRUD (Create, Read, Update et Delete) pour gérer les tarifs de chaque sortie.

4 Schéma d'architecture



Exemple d'architecture

Cette architecture est aussi applicable pour toutes les autres classes modèles/EJB et Servlets.

Dans la couche Web, les Servlets et les fichiers JSP se communiquent pour partager de l'information à afficher dans la vue.

La Servlet configure les URI et récupère le fichier JSP correspondant pour l'envoyer sous forme de requête au client HTTP. La réponse est donc un affichage de vue au format HTML.

5 Développement des différentes interfaces

5.1 Interface utilisateur

Interface de visualisation de la liste des sorties.

Bienvenue khelifi saoussen!

Visualiser les sorties culturelles proposées pour vous				Filter
#	Nom de la sortie	Date de la sortie	Action	
1	Maison Leonardo Da Vinci	2019-02-06 11:23:10.0	Participer	
2	Visite château d'Amboise	2019-02-08 00:00:00.0	Participer	
3	Visite ville de Tours	2020-02-10 10:00:00.0	Participer	
4	Visite ville Blois	2018-08-06 11:23:00.0	Participer	
6	Blois	2018-06-12 11:23:00.0	Participer	

Interface de visualisation de tarif et choix de quantité de billets.

Acheter

Maison Leonardo Da Vinci



le 06/02/2019 à 11:23

#	Description	Quantité	Prix
1	Tarif étudiant avec PCE	- 0 +	6.0
2	Tarif étudiant sans PCE	- 0 +	7.0

Voir votre Panier

Page récapitulative, affichant les coordonnées de l'utilisateur, son choix de sortie, ainsi que le montant total à payer.

Votre commande

#	Description	Quantité	Prix
1	Maison Leonardo Da Vinci Tarif étudiant avec PCE	1	6.0 €
			6.0 €

Coordonnées

khelifi saoussen
saoussen.khelifi@gmail.com

☐ J'atteste que je suis un étudiant d'échange (Relation internationale)

☐ J'accepte les [conditions générales de vente.](#)

Paiement CB

Lorsque l'utilisateur confirme sa commande, la **page de paiement s'affiche.**

Païement de
12.00 EUR

TEST UNIVERSITE FRANCOIS RABELAIS - PRESSES

Numéro de carte

Date de fin de validité (MM/AA)

▼

▼

Cryptogramme visuel :
3 derniers chiffres au dos de la carte (?)

<< ANNULER

VALIDER >>

Paybox ©
Infos Sécurité

Figure 7 – Page de paiement Paybox

5.2 Interface gestionnaire


Interface de gestion de sorties, le gestionnaire peut ajouter, modifier, visualiser la liste de participants à chaque sortie et supprimer une activité culturelle.

Ajouter une activité culturelle			Filtrer
Nom de la sortie	Date de la sortie	Action	
Maison Leonardo Da Vinci	2019-02-06 11:23:10.0	Edit	Visualiser Delete
Visite château d'Amboise	2019-02-08 00:00:00.0	Edit	Visualiser Delete
Visite ville de Tours	2020-02-10 10:00:00.0	Edit	Visualiser Delete
Visite ville Blois	2018-08-06 11:23:00.0	Edit	Visualiser Delete
Blois	2018-06-12 11:23:00.0	Edit	Visualiser Delete

Retour

Interface de gestion de tarifs, le gestionnaire peut ajouter, modifier et supprimer un tarif associé à chaque sortie.

Gestion de tarifs des sorties

Ajouter un tarif Filtrer			
Action	Nom de sortie	Montant	Libelle tarif
 Edit  Delete	Maison Leonardo Da Vinci	6.0	Tarif étudiant avec PCE
 Edit  Delete	Maison Leonardo Da Vinci	7.0	Tarif étudiant sans PCE
 Edit  Delete	Visite château d'Amboise	4.0	Tarif étudiant avec PCE
 Edit  Delete	Visite château d'Amboise	6.0	Tarif étudiant sans PCE

Insertion des données.

Montant

Actif

id Sortie

1 : Maison Leonardo Da Vinci ▼

id TarifRI

1 : Tarif étudiant sans PCE ▼

Envoyer ➡
Retour ⬅

5.3 Réception de billets par messagerie

Lorsque l'utilisateur paie le tarif indiqué pour confirmer sa participation, il devra recevoir par mail son billet électronique qui lui indique le tarif total payé, la date de paiement ainsi que l'id de la sortie.

La référence de ce billet commence par **RI** pour que l'agent comptable de l'université puisse distinguer entre les billets de sorties culturelles (**RI**) et les billets de spectacles (**CULT**).

+-----+
! ATTENTION CECI N'EST PAS UN VRAI PAIEMENT !
! IL N'Y A PAS EU DE VRAIE AUTORISATION !
+-----+

Référence commande: RI-TCK_4

CARTE BANCAIRE

Le 01/04/2018 à 18:08

UNIVERSITE NET

2325805

111122-----
1912

00 001 12834524
M DEBIT @

AUTO: XXXXXX

MONTANT = 20.00 EUR

TICKET A CONSERVER

Figure 8 – Ticket de confirmation

6 Tests et validation

6.1 Tests d'intégration HTML

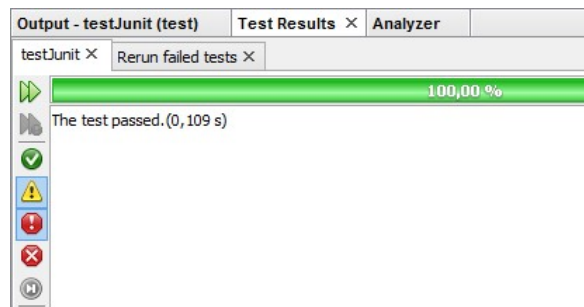
Les pages web s'affichent et fonctionnent correctement sur tous les navigateurs utilisés au quotidien. Comme chaque navigateur garde une interprétation différente de certains styles CSS, les pages fonctionnent bien sur Internet Explorer, Mozilla Firefox, MicrosoftEdge et Google Chrome.

6.2 Tests non fonctionnels

Les fonctionnalités qui ne sont pas implémentées sont :

- connexion avec CAS,
- envoi de mails aux étudiants qui ont participé à une sortie, lorsque la sortie sera modifiée ou annulée,
- gérer la table des gestionnaires RI.

6.3 Tests unitaires



Les tests unitaires permettent de vérifier individuellement que chaque composant de l'application est implémenté conformément aux spécifications.

Classes modèles

Dans un premier temps, j'ai testé les classes modèles, ainsi que les getters et les setters de chaque classe développée.

Ce test fonctionne car je récupère bien la valeur attribuée dans la méthode setter.

Classes EJB

Dans un second temps, j'ai testé les classes DAO qui font le lien avec la base de données. Pour réaliser ce test, il n'est pas suffisant d'appeler l'annotation « Test » du package « org.junit » sur les méthodes de test, il est obligatoire de rajouter d'autre annotation tel que « Before » qui dit que cette méthode sera appelée avant chaque test. Donc, la classe EntityManager et les classes qu'on veut appeler seront initialisées dans la méthode SetUp() qui porte l'annotation Before.

J'ai testé la récupération de l'identifiant (la clé primaire) de chaque table.

Ce test fonctionne car j'ai pu insérer l'« id » avec la méthode « merge() » de la classe « EntityManager » et ensuite le récupérer et assurer que sa valeur ne vaut pas « null » avec la méthode «

assertNotNull() ».

J'ai testé la récupération du nom de la sortie de la table « sorties ».

Ce test fonctionne car j'ai pu récupérer l'« id » avec la méthode « find() » de la classe « EntityManager » et ensuite récupérer le nom de la sortie qui correspond à l'id mentionné.

J'ai testé également la méthode Add() permettant d'insérer des données dans une table.

Ce test fonctionne car j'ai pu insérer dans ma table « sorties » des données sur une sortie en se servant de la méthode « entityManager.persist(Sorties.class) ».

D'autres tests ont été aussi effectués et qui fonctionnent permettant de récupérer toutes les données d'une table en appelant la méthode « getAll() ».

```
@Test
public void testAddSortie() {
    Sorties sorties = new Sorties();
    sorties.setIdSortie(1);
    sorties.setNomSortie("test");
    Date date = new Date();
    Timestamp timestamp = new Timestamp(date.getTime());
    sorties.setDateSortie(timestamp);

    entityManager.persist(Sorties.class);

    sortiesEJB.addSortie(sorties);
    System.out.println(sorties);
    assertEquals("test", sorties.getNomSortie());
    assertEquals(1, sorties.getIdSortie());
}
```

Figure 9 – Exemple de Test

6.4 Tests fonctionnels

Tests effectués	Résultats attendus des tests	Commentaires (test correct ou incorrect)
Gestion du Panier		
Je suis sur une session et je clique sur le bouton « Voir panier » et mon panier est vide	Une information apparaît pour indiquer que le panier est vide	Test validé
Je possède d'un PCE et je choisis un tarif sans PCE	Rien ne se passe	Test validé
Je possède d'un PCE et je choisis un tarif avec PCE	Le billet se rajoute dans le panier selon la quantité choisie	Test validé
Je clique sur « Voir panier »	Des informations sur la sortie choisie, le tarif en euro et la quantité apparaissent	Test validé

Tests sur les liens du site

Tests effectués	Résultats attendus des tests	Commentaires (test correct ou incorrect)
Gestion du menu principal		
Je clique sur le lien vers la page d'accueil du site	La page d'accueil apparaît	Test validé
Gestion de la page d'authentification		
Je clique sur le lien vers la page de connexion	Une identification est demandée	Test validé
Je me connecte avec CAS	Authentification avec les données de CAS	Test non validé
Je clique sur le lien Déconnexion	La page d'accueil apparaît	Test validé
Je clique sur Profil	La page de profil de l'étudiant apparaît	Test validé
Je clique sur Liste des sorties culturelles (user)	La page de liste des sorties culturelles apparaît	Test validé
Je clique sur Gestion des sorties culturelles (admin)	La page de gestion des sorties culturelles apparaît	Test validé
Je clique sur le lien « billets » (pour voir le panier)	Le sous menu contenant la quantité et le prix total apparaît	Test validé

Figure 10 – Exemple de Test

Tests sur les boutons : pour les utilisateurs (ETUDIANT RI)

Tests effectués	Résultats attendus des tests	Commentaires (test correct ou incorrect)
Gestion de la page d'authentification		
Je clique sur le bouton « Filtrer »	Le tableau se met à jour selon le filtre	Test validé
Je clique sur le bouton « Participer »	La page d'achat de billets apparaît	Test validé
Je clique sur le bouton + (pour augmenter la quantité de billets)	La quantité de billet augmente Quantité = quantité + 1	Test validé
Je clique sur le bouton - (pour diminuer la quantité de billets)	Quantité = quantité – 1	Test validé
Je clique sur le bouton « Voir panier »	La page de commandes apparaît	Test validé
Je clique sur le bouton « Commander »	La page de récapitulation apparaît	Test validé
Je clique sur le bouton « Modifier »	La page de commandes apparaît pour modifier la quantité	Test validé
Je clique sur le bouton « paiement CB »	La page de paiement apparaît	Test validé

Figure 11 – Exemple de Test



Conclusion

Problèmes rencontrés

La difficulté rencontrée lors de ce projet a été d'abord au niveau du lancement de serveur GlassFish, la bonne compréhension du code de l'existant et surtout l'implémentation des fonctionnalités existantes.

Bilan

Ce projet est très enrichissant pour moi. J'ai eu le sentiment de progresser de manière significative en organisation, développement et en gestion de projet car cela me permet de lire et de chercher des solutions aux problèmes rencontrés.

Annexes

A

Spécifications non fonctionnelles

1 Sécurité

D'un point de vue confidentialité, l'accès à la plateforme ne sera accessible que par les étudiants concernés.

Le paiement est également sécurisé : l'étudiant utilise sa carte bancaire classique pour payer.

La plateforme sera équipée d'un système de paiement sécurisé. C'est un mode de cryptage des données personnelles (nom, adresse, coordonnées bancaires) qui les rend invisibles, inattaquables et donc qui ne peuvent pas être récupérés par les hackers.

2 Maintenance et évolution du système

Pour répondre à son évolution et aux besoins des utilisateurs, des mis à jour réguliers pourront être effectués. Ainsi, pour améliorer les fonctionnalités et l'interactivité des interfaces, l'application utilise des technologies récentes et des frameworks très utilisés.

B

Conduite du projet

Pour conduire un tel projet, il est nécessaire d'avoir une vue globale du projet.
Ce chapitre se concentre particulièrement sur la façon dont le projet a été conduit.

1 Planning réel

Le planning suivant reconstitue finalement le déroulement des tâches effectuées depuis le lancement du projet.

Nom	Date de début	Date de fin
• Configuration	11/01/18	30/01/18
• Création des classes en JAVA	22/01/18	02/02/18
• Implémentation et développement des IHM	05/02/18	15/02/18
• Communiquer la base avec le projet	20/02/18	01/03/18
• Authentification CAS	01/03/18	20/03/18
☐ • Développement des fonctionnalités principales de l'admin	22/02/18	27/03/18
• Ajouter une sortie	22/02/18	23/02/18
• Visualiser sorties	22/02/18	23/02/18
• Supprimer une sortie	01/03/18	02/03/18
• Modifier une sortie	05/03/18	09/03/18
• Ajouter un tarif	23/03/18	23/03/18
• Modifier un tarif	26/03/18	27/03/18
• Visualiser liste tarifs	22/03/18	23/03/18
• Tests	22/02/18	09/03/18
☐ • Développement des fonctionnalités principales de l'utilisateur	12/03/18	02/04/18
• Consulter	12/03/18	14/03/18
• Participer	14/03/18	19/03/18
• Commander	20/03/18	23/03/18
• Payer	23/03/18	30/03/18
• Tests	14/03/18	02/04/18
☐ • Développement des fonctionnalités secondaires	26/03/18	28/03/18
• Exportation des données des participants	26/03/18	28/03/18

Figure 1 – Liste des tâches

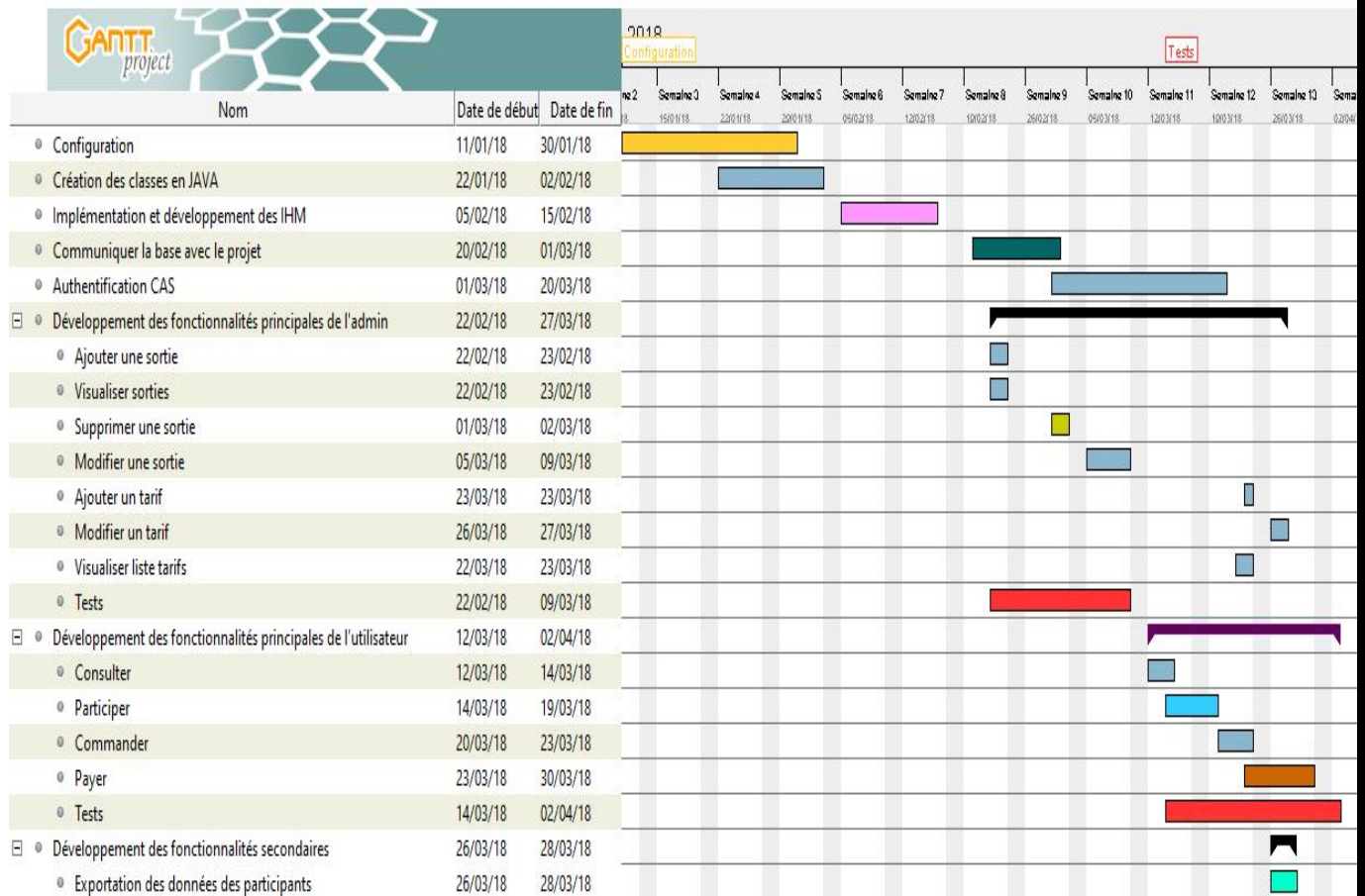


Figure 2 – Diagramme de Gantt du planning du projet

C

Documents annexes

Un cahier de tests qui définit tout les cas de tests effectués, un cahier de développement qui explique en détail chaque classe, chaque fichier et chaque méthode ainsi qu'un manuel d'utilisation pour guider les intervenants sur le projet seront fournis à Mme. Malika. Ces documents seront également disponible sur GIT.



Comptes rendus hebdomadaires

Compte rendu n°1 du 21/09/2017

Bonjour,

voici en PJ le cahier des charges fonctionnel pour la partie "Sorties culturelles des étudiants RI", c'est une première version pour savoir si j'ai bien compris les explications de Mme. Malika et selon vos remarques je modifie ce qui vous semble incorrecte, et je mets à jour la version.

Merci à vous et bonne soirée.
Respectueusement,
Saoussen KHELIFI

Compte rendu n°2 du 25/10/2017

Bonjour Mme. Malika,

j'aimerais vous demander quelque chose au niveau des fonctionnalités de l'application. Pour faire le diagramme UML, j'ai défini que les étudiants RI peuvent uniquement s'inscrire, consulter et choisir une, voire plusieurs sorties proposées par le service. L'administrateur de l'application peut gérer les sorties en ajoutant, modifiant, supprimant une ou plusieurs sorties en précisant les horaires et les informations associées à chaque sortie. Il peut également exporter l'ensemble de participation de chaque étudiant à telle date et telle heure.

J'aimerais donc savoir, voulez-vous que je rajoute d'autres fonctionnalités ? Si oui, que proposez vous ?

Merci à vous.
Respectueusement,
Saoussen Khelifi

Compte rendu n°3 du 25/01/2018

Bonjour Mme. Malika,

cette semaine j'ai fini le développement des IHMs pour l'admin et pour les utilisateurs.

Respectueusement,

Saoussen Khelifi

Compte rendu n°4 du 05/02/2018

Bonjour Mme. Malika,

voici le CR de la semaine :

- 1) J'ai ajouté dans la table Client un champ "role" permettant d'identifier le rôle de chaque client : admin ou user
- 2) J'ai créé une méthode qui appelle et qui récupère l'utilisateur qui veut se connecter et son rôle.
- 3) J'ai passé du temps sur la redirection selon le rôle du client, pour le moment ne marche pas totalement mais j'ai compris le principe, c'est juste au niveau du code JS que je n'arrivais pas à récupérer son rôle.

Voilà.

Bien cordialement,

Saoussen Khelifi

Compte rendu n°5 du 11/02/2018

Bonjour Mme. Malika,

voici ce que j'ai fait cette semaine :

- j'ai commencé à créer les tables pour l'outil de gestion de sorties des RI
- j'ai fait un 1er diagramme de classe représentant le lien entre les tables
- j'ai commencé à faire le lien entre mes classes (tables créées) et la base de données (Affichage de données). Je voulais ce samedi avancer mais j'avais un problème avec Glassfish qui ne voulait plus démarrer ! Ce problème je l'ai résolu en ligne de commande ce matin mais je n'ai pas encore testé avec IntelliJ. Je vous tiens au courant en cas de problème. Également, d'ici la fin de la semaine prochaine, je vais essayer de mettre le diagramme de classes et le modèle de données sur git.

Bien cordialement,

Saoussen Khelifi

Compte rendu n°6 du 25/03/2018

Bonjour Mme. Malika,

voici ce que j'ai fait cette semaine :

- 1) gestion du panier : tout comme la gestion du panier de ticketfac.
- 2) gestion de tarifs : Modification de tarif, suppression et visualisation des tarifs pour chaque sortie.
- 3) page de commandes contenant aussi les données du panier, ainsi la page récapitulative d'achat.
- 4) j'ai réalisé un cahier de tests, de ce qui marche et ce qui ne marche pas.
- 5) j'ai fait également un manuel d'utilisation qui explique par détail la partie configuration de

glassfish, la base de données et IntelliJ.

Ces documents sont sur git, dans "issues".
voilà.

Bien cordialement,
Saoussen Khelifi



Webographie

- [WWW1] DIV. *Les activités culturelles, les industries créatives et les villes - Les éditions de la DIV*. URL : http://www.ville.gouv.fr/IMG/pdf/urbact-villes-culture-2008_cle16a848.pdf (visité le 29/11/2017).
- [WWW2] EclipseLink TEAM. *Framework open source de mapping objet-relationnel pour les développeurs Java*. URL : <http://www.eclipse.org/eclipselink/> (visité le 25/09/2017).
- [WWW3] EXT JS TEAM. *Framework JavaScript Développement de puissantes applications de bureau sur le web*. URL : <http://www.eclipse.org/eclipselink/> (visité le 25/09/2017).
- [WWW4] Université de TOURS. *Application web Billetterie*. URL : <http://ticketfac.univ-tours.fr> (visité le 20/09/2017).
- [WWW5] WIKI. *Le Central Authentication Service - CAS*. URL : <https://fr.wikipedia.org/wiki> (visité le 10/10/2017).

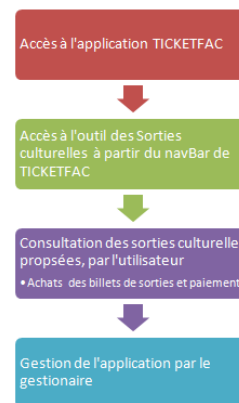
Développement d'une application WEB permettant de gérer les sorties culturelles des étudiants d'échange

Saoussen Khelifi

Encadrement : Malika Labane

Description du projet

Le projet consiste à mettre en place une application WEB permettant de gérer les sorties culturelles des RI. L'application TICKETFAC propose des billets pour assister aux spectacles alors que la nouvelle application propose des sorties et des activités culturelles pour les étudiants d'échange, elle sera accessible à partir de l'application TICKETFAC.



Processus

Objectifs du projet

- Adapter et faire évoluer l'application TICKETFAC,
- Concevoir et développer l'outil afin de proposer et gérer les sorties culturelles,
- Filtrer les étudiants sur le critère d'étudiant d'échange sortant.



Interface

Fonctionnement de l'application

D'abord l'utilisateur choisit une sortie et un tarif avec ou sans pce. Ensuite, il effectue sa commande en ligne. Puis, il paie son choix grâce au service de paiement Paybox.



Paiement en ligne

Développement d'une application WEB permettant de gérer les sorties culturelles des étudiants d'échange

Saoussen Khelifi

Encadrement : Malika Labane

Description du projet

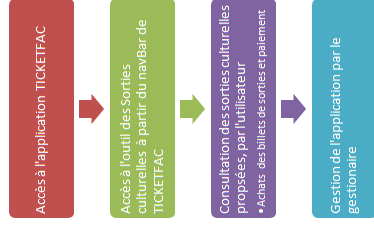
Le projet consiste à mettre en place une application WEB permettant de gérer les sorties culturelles des RI. L'application TICKETFAC propose des billets pour assister aux spectacles alors que la nouvelle application propose des sorties et des activités culturelles pour les étudiants d'échange, elle sera accessible à partir de l'application TICKETFAC.

Objectifs du projet

- Adapter et faire évoluer l'application TICKETFAC,
- Concevoir et développer l'outil afin de proposer et gérer les sorties culturelles,
- Filtrer les étudiants sur le critère d'étudiant d'échange sortant.

Fonctionnement de l'application

D'abord l'utilisateur choisi une sortie et un tarif avec ou sans pce. Ensuite, il effectue sa commande en ligne. Puis, il paye son choix grâce au service de paiement Paybox.



Processus

Payement de 12.00 EUR

****TEST*** UNIVERSITÉ FRANÇOIS RABELAIS - PRESSES

Numéro de carte

Date de fin de validité (MM/AA)

Cryptogramme visuel

3 derniers chiffres au dos de la carte (?)

<< ANNULER VALIDER >>

Paybox ©

FR DE ES

Infos Sécurité

Paieement en ligne



Bienvenue sur le site de billetterie de l'Université de Tours

Vous êtes étudiant, vous souhaitez accéder à l'outil de billetterie de l'Université de Tours ?

Vous êtes enseignant, vous souhaitez accéder à l'outil de billetterie de l'Université de Tours ?

Vous êtes administrateur, vous souhaitez accéder à l'outil de billetterie de l'Université de Tours ?

Interface



Développement d'une application WEB permettant de gérer les sorties culturelles des étudiants d'échange

Résumé

Ce rapport présente le compte rendu de sept mois de recherche et développement effectué pendant les heures de PRD, dans le cadre de la validation de mon année 2017/2018 à l'école polytechnique de Tours.

Ma mission est de modéliser et développer une application qui gère les sorties culturelles des étudiants d'échange tout en améliorant l'application existante TICKETFAC.

L'outil développé est accessible à partir de TICKETFAC. Cet outil devra permettre uniquement aux étudiants RI (relations internationales) de consulter les programmes disponibles par semestre, s'inscrire et choisir une des activités proposées.

Ma première tâche est de rédiger le cahier des charges fonctionnel pour identifier le besoin du client.

La prochaine étape consiste à rédiger le cahier de spécification qui explique l'implémentation des fonctionnalités définies dans le cahier des charges et présente sa conception.

Ensuite, il faut faire les maquettes pour l'étude et la visualisation des fonctionnalités de l'application.

Et finalement, réaliser la partie développement.

Ce projet est développé sous IntelliJ en JAVA et en utilisant les langages HTML, JS (EXT JS) et CSS3, la base de données utilisée sera MySQL.

En conclusion, Ce projet m'a permis de mettre en pratique les connaissances acquises au cours de ma formation.

Mots-clés

Gestion de sortie culturelle, JEE, TICKETFAC, Étudiants d'échange

Abstract

I began my project on September 15th, 2017 under the supervision of Ms. Malika Labane.

The purpose of the project is to design and develop an application that manages students' cultural outings, my mission also requires an improvement of the existing application called "TICKETFAC".

The tool to be developed will only be accessible to IR students from TICKETFAC app. This tool should only allow RI (International Relations) students to consult the available programs per semester, register and choose one of the proposed activities.

My first task is to write the functional specifications to identify the customer's need.

The next step is to write the specification book that explains the implementation of the features defined in the specifications and presents its design.

In conclusion, this project is a very rewarding experience because it allows me to improve my skills in JAVA programming which I appreciate.

This report describes a synthesis of my work during seven months of my project. First, I will start by describing the context of the project.

Secondly, I will explain the principles and basic concepts of the project.

Finally, I will describe my tasks and achievements.

Keywords

students' cultural outings, JEE, TICKETFAC, International Relations