Groupe & Thème du service : ECOM-1FO_1819_Spectacle MySpectacle
Noms des Membres : Cédric LAFRASSE Joffrey FERREIRA Thibaud VEGREVILLE Bastien TERRIER
Lien Wiki : https://air.imag.fr/index.php/ECOM-1FO_1819_Spectacle Lien Dépôt Source: https://gitlab.com/Shadsa/ecom-myspectacle

Fonctionnalités Système Partie 1

Fonctionnalités	Réponses	Partie réservée Enseignant
Authentification (JWT, OAuth2, …)	JWT	
Bases de données utilisées	POSTGRE	
Canevas Web utilisés	ANGULAR MATERIALS	
Nombre de ressources	?	
Nombre de Entity	7	
Nombre de Schedule	?	
Transaction XA	non	
Echanges Client-Serveur (Websocket, REST, Stream (Kafka, RabbitMQ) ...)	REST	
Gestion des médias (BD, File Systems, Caches, CDN, ...)	BD	
Caching	Hibernate 2nd level cache	
Ouitls collaboratifs (scm git, bug tracker, code review,…)	Trello Discord bit.ai	
Méthodologies de test (unitaire, intégration, e2e, performance,…)	Gitlab CI, SonarCube	
Frameworks de test client (swagger, …)	non	
Performances : injection et supervision (JMeter, Gatling, …)	Tests Gatling générés	
Gestion des dépendances (coté serveur, coté client,…)	Maven	

Intégration en continue/ Déploiement en continue (CD-CI)	Gitlab CI	
Livraison en continue		
Cloud utilisé (Azure, AWS, Heroku, GCP, OVH, on premise ...)	Heroku	
Gestion de projet (Méthodologie, …) lien vers le journal,...	SCRUM	

Fonctionnalités Système Partie 2

Fonctionnalités	Réponses (Non/Oui + commentaires si oui)	Partie réservée Enseignant
Haute disponibilité	non	
Mécanismes pour le déploiement automatisé (Docker, Docker Compose, Kubernetes, Swarm, Rancher, ...)	Oui : docker, docker compose, maven	
Déploiement automatisé sur une plateforme cloud	Oui : bug pour le déploiement de la version finale	
Interface CLI ou Shell pour l'administration et le bulk loading (initialisation du catalogue du service, l'ajout de nouveaux produits). Vous pouvez utiliser l'interface EJB facade directement ou bien une interface RESTful.	non	
Gestion de l'internationalisation (i18n) des applications web et mobiles. Remarque : vous pouvez utiliser les principes et outils appris dans l'UE Communication Langagière.	Non : IHM demandée en Français Mais i18n complété en Français et utilisé dans la home page	
Gestion de la confidentialité avec SSL/TLS lors des phases de login, signin, et de paiement, RGPD	non	
Gestion de la concurrence et de la reprise sur panne avec des transactions ACID	non	
Gestion asynchrone et transactionnelle de l'envoi des courriels via AMPQ	non	
Suivi du click stream avec des Filters en vue d'une analyse Big Data avec un ESP (Click Analytics, Recommender System).	non	
Framework d'automatisation des tests (Junit, ...)	non	
Intégration en continue (par exemple avec Jenkins, Travis-CI sur GitHub)	Gitlab CI	

Livraison en continue (Rolling update)	non	
Injection de Pannes (Netflix Simian Army, ...)	non	
Reprise sur panne	non	
Performances (résultat du injection de charge avec Apache JMeter ou Gatling)	Oui : tests Gatling générés mais pas exécutés	
Infrastructure de supervision du système (Telegraf, Prometheus ...)	Oui (Jhipster fourni une interface)	
Validation des services REST (Swagger, ...)	oui	
Renseignement de la notice relative à la protection de la vie privée. (RGPD …)	non	
Utilisation de OAuth ou OpenID pour le login	non	
Conditionnement de l'application mobile (Ionic, Cordova, plugins utilisés …)	Application Ionic Cordova avec plugin de QR code (non liée à la base Jhipster car version non déployée)	
Utilisation d'API tiers (Stripe, PubNub, Firebase, Google analytics, GeoIP, Criteo…)	non	
Autres (listez les autres fonctionnalités intégrées) :		