Groupe & Thème du service : Sport – Mes Courses
Noms des Membres : FOMBARON, LEPAGE, AUBERT, GROS-DAILLON
Lien Wiki : https://air.imag.fr/index.php/ECOM-1FO_1819_Sport Lien Dépôt Source: Web : https://gricad-gitlab.univ-grenoble-alpes.fr/grosdaih/ecom-mescourses / Mobile : https://github.com/TimLepage/MesCoursesApp

Fonctionnalités Système Partie 1

Fonctionnalités	Réponses	Partie réservée Enseignant
Authentification (JWT, OAuth2, …)	JWT	
Bases de données utilisées	MySQL	
Canevas Web utilisés	Angular, Bootstrap	
Nombre de ressources	???	
Nombre de Entity	4 Entities – 3 enums	
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	Non	
Ouitls collaboratifs (scm git, bug tracker, code review,…)	Gitlab, github	
Méthodologies de test (unitaire, intégration, e2e, performance,…)	Gatling, Cucumber, Protractor (auto generated) + Junit and Jest	
Frameworks de test client (swagger, …)	Non	
Performances : injection et supervision (JMeter, Gatling, …)	Gatling (auto generated)	

Gestion des dépendances (coté serveur, coté client,…)	Client : npm / Serveur : maven	
Intégration en continue/ Déploiement en continue (CD-CI)	Oui / Oui	
Livraison en continue	Oui	
Cloud utilisé (Azure, AWS, Heroku, GCP, OVH, on premise ...)	Heroku	
Gestion de projet (Méthodologie, …) lien vers le journal,...	Scrum, https://air.imag.fr/index.php/ECOM-1FO_1819_Sport_L9_LOG	

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, déploiement automatique sur Heroku après que le build et les tests soient passés suite à un push sur gitlab, directement dans le script de CI/CD : **.gitlab-ci.yml**	
Déploiement automatisé sur une plateforme cloud	Oui, déploiement automatique sur Heroku après que le build et les tests soient passés suite à un push sur gitlab.	
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	
Gestion de la confidentialité avec SSL/TLS lors des phases de login, signin, et de paiement, RGPD	Oui, automatique avec Jhipster (pour le login)	
Gestion de la concurrence et de la reprise sur panne avec des transactions ACID	Oui : grâce à MySQL, script détection et reprise sur panne	

Gestion asynchrone et transactionnelle de l'envoi des courriels via AMPQ	Envoi de mail avec les méthodes générées par Jhipster.	
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, ...)	Gatling, Cucumber, Protractor (auto generated) + Junit and Jest	
Intégration en continue (par exemple avec Jenkins, Travis-CI sur GitHub)	Oui, avec Gitlab-CI/CD	
Livraison en continue (Rolling update)	Oui, avec Gitlab-CI/CD	
Injection de Pannes (Netflix Simian Army, ...)	Non	
Reprise sur panne	Oui, Heroku garantit que votre application récupère automatiquement des pannes de serveur.	
Performances (résultat du injection de charge avec Apache JMeter ou Gatling)	Gatling (auto generated with JHipster)	
Infrastructure de supervision du système (Telegraf, Prometheus ...)	Oui, Jhipster le gère dans le registry pour les services enregistrés.	
Validation des services REST (Swagger, ...)	Non	
Renseignement de la notice relative à la protection de la vie privée. (RGPD …)	Non	
Utilisation de OAuth ou OpenID pour le login	JWT	
Conditionnement de l'application mobile (Ionic, Cordova, plugins utilisés …)	Ionic, Cordova	
Utilisation d'API tiers (Stripe, PubNub, Firebase, Google analytics, GeoIP, Criteo…)	https://adresse.data.gouv.fr/api, Pour pouvoir récupérer le lieu entré par l'utilisateur (en string) et le transformer automatiquement en coordonnées longitude, latitude pour la base de données. (Et pouvoir marquer les courses sur une carte leaflet).	

Autres (listez les autres fonctionnalités intégrées) : *-Intégration d'une carte Leaflet OpenStreetMap, reliée à nos courses en base de données.* *-Chargement « Lazy » des images pour ne pas qu'elles soient toutes chargées lors de l'ouverture de l'application.* *-ElasticSearch.*	