Groupe & Thème du service : Groupe 4 – Mes Locations Vacances
Noms des Membres : Antoine Boisadam, Maxime Dereymez, Timothée Lemaire, Lucas Lesage, Douria Zennouche
Lien Wiki : https://air.imag.fr/index.php/ECOM_RICM5_Groupe4_2017 **Lien Dépôt Source:** https://github.com/ECOM2017-MesLocationsVacances/

Fonctionnalités Système Partie 1

Fonctionnalités	Réponses	Partie réservée Enseignant
Serveur JavaEE utilisé (Glassfish, Jboss, Geromino, ...)	Wildfly	
Bases de données utilisées	MariaDB (SQL)	
Canevas Web utilisés	AngularJS, Bootstrap	
Nombre de Beans facades (Stateful, stateless)	Stateless	
Nombre de Entity Beans	4	
Nombre de EJBTimers	Non	
Transaction XA	Non	
Echanges Client-Serveur (AJAX, DDP, Websocket, REST, WS, JMS ...)	Rest (JAX-RS)	
Gestion des médias (BD, File Systems, Caches, CDN, ...)	Amazon S3 (Gestion de fichiers), BD geree nous meme	
Caching	--	
Ouitls collaboratifs (scm git, bug tracker...)	Git, Waffle.io et Github pour le bug-tracking	
Méthodologies de test (unitaire, intégration, performance)	Test d'integration pour le back	
Frameworks de test client Javascript	--	
Frameworks de test JavaEE	Arquillian	
Gestion des dépendances (coté serveur, coté client)	Maven cote serveur, npm cote front	
Intégration en continue	CircleCI	
Livraison en continue	CircleCI + AWS ECS	
Cloud utilisé (Azure, EC2, ...)	Amazon Web Services (EC2, ECS, S3)	
Gestion de projet (Méthodologie, …) lien vers le journal, ...	SCRUM via Waffle.io	

Fonctionnalités Système Partie 2

Fonctionnalités	Réponses (Non/Oui + commentaires si oui)	Partie réservée Enseignant
Haute disponibilité	OUI avec HAProxy (roundrobin entre 2 serveurs)	
Mécanismes pour le déploiement automatisé (Docker, Docker Compose, ...)	OUI, Docker et Docker-compose	
Déploiement automatisé sur une plateforme cloud	OUI, CircleCI et script bash maison de déploiement chez AWS	
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.	OUI interface d'admin + CLI via la DB	
Gestion de l'internationalisation (i18n) des applications web et mobiles.	Non	
Gestion de la confidentialité avec SSL/TLS lors des phases de login, signin, et de paiement	OUI, SSL géré avec HAProxy	
Gestion de la concurrence et de la reprise sur panne avec des transactions ACID	OUI, reprise sur panne OK mais transaction pas forcement ACID	
Gestion asynchrone et transactionnelle de l'envoi des courriels via JMS et des EJBTimer (voir http://ecom.ow2.org/xwiki/bin/view/Main/fremail)	OUI, envoi email via JMS (smtp.gmail.com)	
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, ...)	Maven test	
Intégration en continue (par exemple avec Travis-CI sur GitHub)	OUI, CircleCI	
Livraison en continue (Rolling update)	OUI, CircleCI + AWS	
Injection de Pannes (Netflix Simian Army, ...)	NON	
Reprise sur panne	OUI, un agent veille a ce que le serveur soit toujours up, le redémarre sinon	
Performances (résultat du injection de charge avec Apache JMeter ou CLIF)	NON	
Infrastructure de supervision du système (Telegraf, ...)	OUI ! CollectD, Telegraf, InfluxDB et Grafana	
Validation des services REST (Swagger, ...)	Non ? (pas compris)	
Renseignement de la notice relative à la protection de la vie privée.	NON	
Utilisation de OAuth ou OpenID pour le login	NON	
Conditionnement de l'application mobile	NON	