École de technologie supérieure

Département de génie logiciel et des TI

Cours : LOG210 - Analyse et conception de logiciels

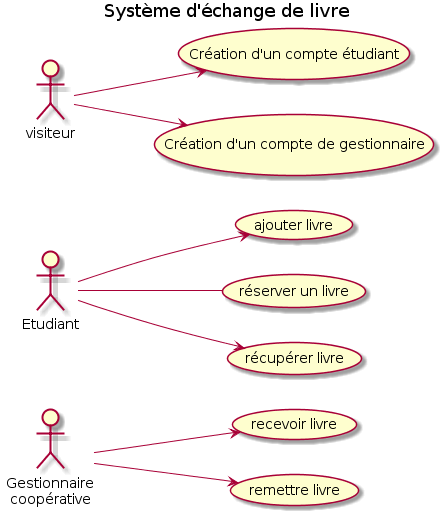
Date de création : 2015-08-27

Date de la dernière modification : 2015-08-27

Système d’échange de livres

Dans le cadre d’un concours national de l’ACEQ (L’association des coopératives des étudiants du Québec) il est demandé à chaque université de développer une application web d’échange de livres pour les étudiants. L’objectif principal étant de permettre aux étudiants d’acheter des livres à prix modique ou de les vendre tout en faisant un bon geste pour l’environnement.

Les coopératives de chaque université devront héberger chacune un serveur leur permettant d’offrir le service aux étudiants. Les étudiants pourront mettre leur livres en consigne pour que la coopérative locale puisse les vendre ou les échanger avec d’autres étudiants. Les coopératives désirant échanger des livres pourront développer une interface REST commune.



L’association a beaucoup d’exigences et sait de toute évidence que les développeurs ne peuvent pas toutes les réaliser à cause des contraintes de budget et de temps. Elle vous a donc préparé une liste d’exigences en incluant son évaluation de l’impact que ces divers facteurs peuvent avoir sur son application. Elle se fie au bon jugement et à l’expérience des membres de votre équipe pour organiser chacune de vos itérations de telle sorte que votre équipe réalise le projet en tenant compte de tous les risques. C’est la force d’une approche itérative et évolutive, après tout.

Le rendement au niveau des laboratoires sera calculé à 50% sur l’approche itérative et évolutive, correspondant aux activités d’analyse de la conception et de la planification selon le Processus unifié, alors que le défi technologique ou, si vous préférez, la satisfaction du client correspondra à l’autre 50%.

Votre équipe a donc l’entière responsabilité de choisir soigneusement les exigences que vous jugez réalisables dans une itération. N’oubliez pas de tenir compte des risques et de l’expérience de votre équipe dans vos choix. Toutes exigences planifiées et non satisfaites durant une itération pénaliseront la note de l’équipe. Veuillez toutefois noter que cette pénalité sera moindre si l’exigence est quand même satisfaite à la fin du projet.

Prenez note que l’association est ouverte à toute suggestion qui permettrait d’améliorer la qualité du produit ou du service. N’hésitez pas à en discuter avec le représentant de l’association (Chargé de laboratoire) qui pourra évaluer votre proposition et en faire part à l’association (Professeur ou chargé de cours) si celles-ci s’avèrent intéressantes pour le projet.

## Conseils pour tous les membres de votre équipe:

Puisque les choix technologiques faits au début du projet peuvent avoir un impact majeur sur toute la session, il faut bien réfléchir à ces choix.

Par exemple, deux personnes dans l’équipe sont très motivées pour utiliser PHP pour la solution, mais deux autres personnes ne connaissent pas cette technologie et ne sont pas trop motivées pour l’apprendre. Si l’équipe décide d’aller avec un choix comme celui-là, il y a un risque de conflit plus tard dans la session lors de la livraison des fonctionnalités. Ceux qui connaissent moins une technologie auront une mise à niveau à faire et vont faire plus d’erreurs, voire être presque inutile dans le projet. Même s’ils sont motivés pour l’apprendre, ils seront moins performants que les autres. Certains types d’étudiants aiment ce genre de défi malgré l’effort supplémentaire, d’autres se contentent de faire un travail adéquat moins risqué.

Le choix de l’équipe doit être un **consensus**. Vous devez considérer tous ces aspects (expérience, effort, disponibilité, etc. de chaque membre) en tant qu’équipe et non en tant que plusieurs membres individuels.

Considérez l’approche suivante: le choix avec le moins de risque (et qui peut avoir une note raisonnable) est de faire tout en Java avec les technologies simples. Commencez avec cette idée de base et élargissez les objectifs un par un, selon la disponibilité, la motivation et les connaissances de l’équipe **ensemble**.

# Grille des exigences FURPS+

|  |  |  |
| --- | --- | --- |
| Points | Facteur | Exigence |
| 5 |  | F1-CU01 – Création d’un compte gestionnaire coopérative |
| 5 |  | F2-CU02 – Création d’un compte étudiant |
| 15 |  | F3-CU03 – Ajout d’un livre à échanger ou vendre |
| 10 |  | F4-CU04 – Remise du livre à la coopérative de l’école |
| 10 |  | F5-CU05 – Réservation d’un livre |
| 10 |  | F6-CU06 – Récupération d’un livre réservé à la coopérative |
| 5 |  | F7-CU07 – Demande de transfert de livre entre coopératives |
| 5 |  | F8-CU08 – Réception des livres provenant d’une autre coopérative |
| 5 |  | F9 – Intégration dans une page entreprise de Facebook |
|  | 1.025 | U1 – Conception de sites web adaptatifs |
|  | 1.025 | U2 - Ajax |
|  | 1.025 | U3 – Internationalisation |
| 5 |  | P1 – Rapport de temps de réponse |
|  | 1.025 | P2 – Utilisation de cache local |
|  | 1.000 | S1 – Prise en charge web |
| 5 |  | S2 – Prise en charge REST |
|  | 1.300 | S3 – Prise en charge mobile pour le gestionnaire |
|  | 1.300 | S4 – Prise en charge mobile pour les étudiants |
|  | 0.750 | S5 – Aucune prise en charge mobile ou web |
|  | 1.025 | S6 – Persistance avec Hibernate |
|  | 1.025 | S7 – Persistance avec MongoDB |
|  | 1.000 | S8 – Persistance avec MySql, Derby, Postgres |
|  | 1.000 | S9 – Persistance avec une base de données XML |
|  | 0.500 | S10 – Persistance dans des fichiers texte |
| 15 |  | Im1 – environnement de test |
| 5 |  | Im2 – Utilisation de Patron GOF |
| 5 |  | In1 – Changement d’état par SMS |
| 10 |  | In2– Paiement par PayPal |
| 10 |  | In3 - Intégration avec Ebay |
| 5 |  | In4 - Participation active à l’élaboration des interfaces |
|  | 1.025 | E1 – Déploiement sur un serveur externe |

F – Fonctionnalité, U – Convivialité, R – Fiabilité, P – Performance, S – Possibilité de prise en charge, +Im – Implémentation, +In - Interface, +E – Exploitation, +C – Conditionnement, +A – Aspect juridique

Évaluation: Pour le calcul de la note, voir la section Évaluation.

# F1-CU01 – Création d’un compte de gestionnaire de coopérative

**Précondition(s) :**

* Le visiteur n’a pas de compte

**Postconditions(s) :**

* Un gestionnaire de coopérative a été créé

**Acteur principal :**

* Le gestionnaire de la coopérative

**Scénario principal**

1. Le gestionnaire de la coopérative commence la création d’un compte
2. Le gestionnaire fournit son courriel ainsi qu’un mot de passe
3. Le gestionnaire ajouter l’information concernant la coopérative. Soit son nom, et son adresse.
4. Le gestionnaire confirme l’information.
5. Le système enregistre le gestionnaire et crée le compte de la coopérative.

**Scénarios alternatifs**

1. Aucun

# F2-CU02 – Création d’un compte étudiant

**Précondition(s) :**

* Le visiteur n’a pas de compte

**Postconditions(s) :**

* Un compte étudiant a été créé

**Acteur principal :**

* Étudiant

**Scénario principal**

1. L’étudiant commence la création d’un compte étudiant
2. Le système demande l’information sur le courriel et le mot de passe de l’étudiant

**Scénarios alternatifs**

2a. Le système peut aussi utiliser le no de téléphone pour se connecter au compte

1. Le système demande l’information du no de téléphone

# F3-CU03 – Ajout d’un livre à échanger ou vendre

**Précondition(s) :**

* L’étudiant est authentifié

**Postconditions(s) :**

* Un nouveau livre a été ajouté dans le système

**Acteur principal :**

* L’étudiant

**Scénario principal**

1. L’étudiant entame l’ajout d’un nouveau livre
2. Le système demande le code du livre
3. L’étudiant entre le code ISBN du livre
4. Le système fait une recherche du code ISBN pour trouver l’information concernant le titre du livre, l’auteur, le nombre de pages ainsi que le prix de celui-ci.
5. Le résultat est affiché à l’étudiant qui peut manuellement compléter l’information manquante.
6. L’étudiant indique l’état du livre et confirme l’information
7. Le système affiche l’information sur le livre.

Les étapes 2 à 7 sont répétées tant que l’étudiant désire ajouter des livres.

**Scénarios alternatifs**

3a. Le code du livre peut avoir plusieurs formats

1. Le code du livre est ISBN
2. Le code du livre est un code-bar EAN
3. Le code du livre est un code-bar UPC

Note : De préférence, l’extraction de ces informations devrait être réalisée au travers d’une API offerte par un fournisseur. Allez voir sur le Wiki du cours dans Moodle pour de l’information supplémentaire.

4a. Le système recherche l’information sur le livre à différent endroit

1. Le système recherche dans sa base de données pour voir si ce livre aurait déjà été proposé.
2. Le système fait une recherche au niveau des autres coopératives pour voir si elle n’aurait pas ces informations
3. Le système recherche sur Amazon
4. Le système demande à l’usager de remplir l’information puisqu’il ne trouve aucune information sur ce livre.

5a. Le prix du livre

1. Le prix du livre est indiqué directement sur le livre
2. Le prix du livre est obtenu dans une autre coopérative et suggéré à l’étudiant.
3. Le prix du livre est obtenu au travers d’un fournisseur externe.

Pour minimiser l’effort nécessaire à la gestion de l’inventaire, la coopérative veut une méthode simple et efficace pour la gestion des livres. **Elle ne veut pas associer un livre spécifique avec un vendeur puisque les livres n’ont pas de numéro de série**. **Un livre à un seul et unique prix neuf**

Note : C’est la coopérative qui à le dernier mot pour définir le prix du livre. L’étudiant accepte ce prix où refuse la transaction. L’étudiant ne sera payé que lorsque le livre sera vendu. Après 2 sessions, la coopérative pourra demander à l’étudiant de venir récupérer un livre ou en disposer si l’étudiant ne le réclame pas.

6a. Il y a trois états possibles qui définissent le prix du livre.

1. Comme neuf (75% du prix neuf)
2. Quelques pages pliés, utilisation d’un marqueur (50% du prix neuf)
3. Livre très utilisé, pages plié, couverture endommagés (25 du prix neuf)

# F4-CU04 – Remise du livre à la coopérative de l’école

**Précondition(s) :**

* Le gestionnaire de la coopérative est authentifié
* L’étudiant à déjà ajouté le livre dans le système
* L’étudiant à le livre en sa possession et est à la coopérative de son école

**Postconditions(s) :**

* La réception du livre est confirmée

**Acteur principal :**

* Le gestionnaire de la coopérative

**Scénario principal**

1. Le gestionnaire de la coopérative démarre le processus de réception d’un livre
2. Le système demande l’information sur le livre
3. Le gestionnaire fait une recherche par ISBN, UPC, titre du livre ou nom de l’étudiant
4. Le système affiche la liste des livres qu’il s’attend à recevoir.
5. Le gestionnaire identifie le livre à recevoir et confirme l’état de celui-ci
6. Le livre est prêt à être vendu ou échangé.
7. Le gestionnaire termine le processus de réception des livres
8. Le système envoie un courriel à l’étudiant pour confirmer la réception des livres

Les étapes 2 à 6 sont répétées tant que l’étudiant à des livres a remettre.

**Scénarios alternatifs**

3a. La recherche ne donne aucun résultat

1. Le gestionnaire recommence la recherche
2. Le gestionnaire termine le processus de réception de livre.

5b. Le gestionnaire change l’état du livre

1. L’étudiant accepte le changement et le processus continu
2. L’étudiant refuse le changement et demande au gestionnaire d’annuler le processus.

# F5-CU05 – Réservation d’un livre

**Précondition(s) :**

* L’étudiant est authentifié

**Postconditions(s) :**

* L’étudiant a réservé un livre, l’a payé et peut passer le chercher à la coopérative.

**Acteur principal :**

* L’étudiant

**Scénario principal**

1. L’étudiant démarre une recherche de livre
2. Il entre le code ISBN, UPC, le nom de l’auteur ou une partie du titre.
3. Le système affiche l’information sur la liste des livres à vendre incluant l’état et le prix de chaque état.
4. L’étudiant sélectionne le livre qu’il veut acheter
5. Le système demande une confirmation
6. Le livre est réservé pour 48h le temps que l’étudiant se présente à la coopérative pour le récupérer.

Les étapes 1 à 6 sont répétées tant que l’étudiant n’a pas trouvé tous les livres qu’il recherche.

**Scénarios alternatifs**

2a. Le système ne trouve pas de livre dans son inventaire

1. Le système demande à l’étudiant s’il veut entrer le code ISBN du livre et être notifié lorsqu’un étudiant ajouter celui-ci.
2. L’étudiant entre le code ISBN
3. Le système recherche l’information sur le livre et affiche celle-ci à l’étudiant.
4. L’étudiant confirme son intention d’être notifié

2a.3 Le système ne trouve toujours pas de livre dans son inventaire

1. Le système recherche dans les coopératives partenaires

5a. Le livre provient d’une coopérative partenaire

1. L’étudiant accepte de payer les frais de transport qui sont préétablis entre les coopératives.

6a. L’étudiant ne passe pas récupérer le livre avant 48h.

1. Le système conserve l’information sur la réservation, mais ne peut garantir celle-ci puisqu’un autre étudiant peut maintenant réserver celui-ci.

6b. Le livre provient d’une coopérative partenaire

1. Le gestionnaire de la coopérative indiquera la réception du livre lorsqu’il le reçoit physiquement.
2. Le système informe l’étudiant que le livre est reçu et qu’il a 48h pour venir le chercher.

# F6-CU06 – Récupération d’un livre réservé à la coopérative

**Précondition(s) :**

* Le gestionnaire est authentifié
* L’étudiant est à la coopérative

**Postconditions(s) :**

* Le livre est remis à l’étudiant et la transaction est enregistrée dans le système.

**Acteur principal :**

* Le gestionnaire

**Scénario principal**

1. L’étudiant se présente à la coopérative et demande de recevoir son ou ses livres.
2. Le gestionnaire de la coopérative démarre une demande de livraison.
3. Le système affiche toutes les livraisons possibles.
4. Le gestionnaire sélectionne la réservation correspondant à l’étudiant.
5. L’étudiant vérifie la qualité du livre et accepte la transaction.
6. Le gestionnaire valide l’identité de l’étudiant
7. Le gestionnaire indique que la transaction est acceptée.

**Scénarios alternatifs**

5a. L’étudiant refuse la transaction

1. Le gestionnaire indique que la transaction est annulée
2. Le système rembourse l’étudiant
3. Le système envoie un courriel à l’étudiant pour résumer la transaction

# F7-CU07 – Demande de transfert de livre entre coopératives

**Précondition(s) :**

* Le gestionnaire de la coopérative est authentifié

**Postconditions(s) :**

* La date et l’heure d’expédition des livres sont enregistrées dans le système

**Acteur principal :**

* Le gestionnaire

**Scénario principal**

1. Le gestionnaire demande la liste des livres à expédier au système
2. Le gestionnaire expédie les livres
3. Le gestionnaire indique quels sont les livres expédiés et à quelles coopératives

**Scénarios alternatifs**

Aucun

# F8-CU08 – Réception des livres provenant d’une autre coopérative

**Précondition(s) :**

* Le gestionnaire de la coopérative est authentifié

**Postconditions(s) :**

* Les livres reçus sont identifiés et l’étudiant est notifié.

**Acteur principal :**

* Le gestionnaire

**Scénario principal**

1. Le gestionnaire consulte la liste des livres à recevoir et indique ceux qu’il vient de recevoir.
2. Le système envoie un courriel pour indiquer à l’étudiant que la coopérative vient de recevoir son ou ses livres et qu’il a 48 heures pour en faire la récupération.

L’étape 1 est répétée tant que le gestionnaire à des livres à recevoir.

**Scénarios alternatifs**

Aucun

F9- Intégration dans une page entreprise de Facebook Convivialité

L’association aimerait qu’il soit facile pour les étudiants d’utiliser le site en passant par Facebook. Il faudra donc intégrer quelques pages du site dans les pages Facebook. Facebook nécessite un serveur sécurisé pour interagir avec une application externe. Je vous suggère donc l’utilisation du gem ngrok qui permet d’utiliser une url externe pour accéder au site héberger sur votre ordinateur personnel.

Ex : ngrok 3000



Vous pouvez maintenant utiliser l’adresse http://2b059861.ngrok.com pour accéder à votre serveur hébergé à l’adresse locale 127.0.0.1:3000

## U1 – Conception de sites web adaptatifs

(nécessite S1) L’association veut que l’application soit facilement utilisable sous plusieurs formats de résolution (moniteur 4 :3, 16 :9, i Pad, téléphone intelligent). On demande le support d’au moins 2 formats qui peuvent être sur la même plateforme.

Voir [http://fr.wikipedia.org/wiki/Site\_web\_adaptatifhttp://fr.wikipedia.org/wiki/Site\_web\_adaptatif](http://fr.wikipedia.org/wiki/Site_web_adaptatif)

## U2 - Ajax

L’association aimerait avoir une application fluide en utilisant des technologies évitant un rafraîchissement complet de la page.

Voir [http://fr.wikipedia.org/wiki/Ajax\_(informatique)http://fr.wikipedia.org/wiki/Ajax\_(informatique)](http://fr.wikipedia.org/wiki/Ajax_(informatique))

## U3 – Internationalisation

L’association offre déjà des services dans différentes langues. Il demande donc que le système puisse supporter plusieurs langues. Vous devez traduire votre système en au moins une langue supplémentaire et les utilisateurs doivent pouvoir changer celle-ci lors de l’utilisation de l’application. Le texte pour la langue doit être séparé du code source.

# Fiabilité (Reliability)

## R1 – N/A

# Performances

## P1 – Rapport de performance

L’association aimerait que l’application ait une bonne rapidité de réaction. L’association s’attend à un rapport contenant les temps de réponse de chaque fonction du système. Vous devrez donc documenter adéquatement les essais et dispositifs mis en place pour améliorer la performance de votre système. En plus de soumettre votre système à un test de performance réalisé par le chargé de laboratoire consistant à ajouter plusieurs dizaines de livres en utilisant l’interface REST de votre serveur. Les tests seront réalisés à l’aide de l’outil en ligne de commande curl.

Curl <http://2b059861.ngrok.com/livres/add?isbn=9780471690078>

Curl <http://2b059861.ngrok.com/livres/add?upc=9780471690078>

Vous pouvez aussi utiliser la commande ab pour vérifier les performances de votre serveur.

ab -c 100 -n 100 http://service/path/to/resource

* "-c 100" means 100 concurrent requests and
* "-n 100" means 100 requests

Je recommande aussi l’utilisation du site gtmetrix.com pour vous aider à améliorer les performances de votre site.

Quelle sera l’équipe ayant l’application la plus performante ?

## P2 – Utilisation de cache local

Amélioration de la performance du système en utilisant le processus de mise en cache local selon la section 30.2 du livre de Larman. UML 2 et les patrons de conception. Vous devez être en mesure de démontrer la différence de performance pour une requête qui se rend jusqu’au serveur et une requête qui est optimisée grâce à la mise en cache.

Redis est un serveur de donnée ultra rapide qui pourrait vous permettre de faire du cache de données.

# Possibilité de prise en charge

## S1 - Prise en charge web

L'entrepreneur aimerait que son application soit utilisable avec un fureteur aux fins d'accessibilité."

## S2 – Prise en charge REST

Pour supporter la réutilisation de modules, l’association aimerait que son application soit utilisable au travers d’un API REST. Ainsi, un tiers parti pourra inclure le module de recherche de livres dans son propre site web.

Voir [http://fr.wikipedia.org/wiki/Representational\_State\_Transferhttp://fr.wikipedia.org/wiki/Representational\_State\_Transfer](http://fr.wikipedia.org/wiki/Representational_State_Transfer)

<https://docs.google.com/a/etsmtl.net/document/d/1vo31SBRya-enxQTNSJWuV2RtrhmtlkfizwAWX4uKtM4/edit?usp=sharing>

## S3 – Prise en charge mobile pour le gestionnaire

Les gestionnaires aimeraient avoir une application mobile pour la réception et la cueillette des livres.

## S4 – Prise en charge mobile pour le client

Les clients quant à eux aimeraient bien être en mesure de scanner leurs livres pour les ajouter rapidement dans leur liste de livre à vendre.

## S5 – Aucune prise en charge web ou mobile

L'entrepreneur a absolument besoin d’une application accessible à distance. Si vous faites une version sans accès à distance, il y a une pénalité.

## S6 – Persistance avec Hibernate

L'entrepreneur prévoit un changement d’infrastructure dans un ou deux ans et la solution ne devrait pas trop dépendre du type de base de données. Utiliser Hibernate pour persister vos données. Cette exigence nécessite l’exigence S7.

## S7 – Persistance avec MongoDB

Utilisez une base de données orientée documents pour persister les données de vos clients. Voir [http://fr.wikipedia.org/wiki/Base\_de\_données\_orientée\_documentshttp://fr.wikipedia.org/wiki/Base\_de\_donn%C3%A9es\_orient%C3%A9e\_documents](http://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_orient%C3%A9e_documents)

## S8 – Persistance avec MySql, Derby, Postgres

Utiliser une banque de données relationnelle pour sauvegarder les données.

## S9 – Persistance avec une base de données en XML

Utiliser une base de données XML pour sauvegarder vos données

## S10 – Persistance dans des fichiers texte

Réalisez vous-même la sauvegarde et l’extraction des données à partir de fichiers texte.

# Implémentation

## Im1 - Environnement de test

Un des aspects très importants de l’implémentation est la robustesse du code. Vous devez vous assurer que le code est robuste et le reste tout au long du processus itératif de développement. Une méthode pour y arriver consiste à implanter un environnement de test. **Vous devez donc démontrer une progression des tests couvrant la couche logique applicative, et ce à chaque itération**. La note finale sera de 15 points multipliés par le pourcentage de couverture de code que vous aurez obtenu à la livraison finale du projet. Chaque membre de l’équipe devra démontrer qu’il est en mesure d’exécuter les tests et d’en écrire au moins un.

## Im2 – Utilisation de Patron GOF

Pour cette exigence, vous devez démontrer et documenter l’utilisation des patrons GOF dans votre application.

# Interfaces

## In1 – Changement d’état par SMS

En sachant que la clientèle des coopératives est jeune et très techno, nous aimerions supporter la notification par SMS pour informer les étudiants de tout changement concernant leurs livres.

## In2 – Paiement par PayPal

Dans le but d’accélérer et de faciliter le travail des gestionnaires de coopérative, toutes les transactions financières du site avec les étudiants doivent être effectués par PayPal.

Voir [https://developer.paypal.com/webapps/developer/docs/api/ - api-endpointshttps://developer.paypal.com/webapps/developer/docs/api/ - api-endpoints](https://developer.paypal.com/webapps/developer/docs/api/#api-endpoints)

Voir [https://developer.paypal.com/webapps/developer/docs/api/https://developer.paypal.com/webapps/developer/docs/api/](https://developer.paypal.com/webapps/developer/docs/api/)

## In3 - Intégration avec Ebay

Dans le but de maximiser leurs ventes, les coopératives veulent avoir la possibilité d’offrir les livres sur Ebay. Il devrait cependant avoir cette possibilité seulement lorsque la session est débutée et qu’il devient peu probable que le livre soit vendu à un étudiant. Il faut toujours que les étudiants aient priorité sur les ventes externes. On doit donc pouvoir ajouter et enlever facilement les livres sur Ebay. On suggère d’utiliser l’api de Ebay pour traiter ces requêtes.

Voir: <https://go.developer.ebay.com/what-ebay-api>

## In4 - Participation active à l’élaboration des interfaces

(Necessite S2) En considérant que l’élaboration d’interface prend quand même un certain temps, nous accorderons un certain nombre de points à l’équipe qui participera activement à documenter l’interface publique que toutes les équipes devront utiliser pour l’échange d’information entre les coopératives. Ceci vous demandera d’interagir avec les membres d’autres équipes pour définir conjointement l’interface. Je vous suggère de nommer un responsable par équipe pour se charger de ce travail.

# Exploitation

## E1 – Déploiement sur un serveur externe

La coopérative désire s’assurer que vous êtes en mesure de déployer votre application sur un serveur externe. Ceci permettra de la rassurer quant au fait que le serveur pourra être porté sur une plateforme infonuagique dans la seconde phase du projet.

# Évaluation

Le travail de laboratoire sera évalué en deux volets, soit la partie processus et planification et la partie implémentation. Cette dernière sera évaluée vers la fin de chaque itération (*e1*, *e2*, *e3*) mais ce n’est que l’évaluation (*e*) de l’implémentation à la dernière itération qui comptera pour la note.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Évaluation | Itération 1 | Itération 2 | Itération 3 | Itération 4 | Total |
| Processus/planification | *a* | *b* | *c* | *d* | (*a*+*b*+*c*+*d*) |
| Implémen- tation | *e1* | *e2* | *e3* | *e* | *e* |

L’équipe remet les artéfacts pour chaque itération selon le tableau suivant :

|  |  |  |
| --- | --- | --- |
| Itération | Minimum des points d’exigences fonctionnelles[[1]](#footnote-0) | Artéfacts à remettre |
| 1 | 10 | Plan d’itération[[2]](#footnote-1) Modèle du domaine (MDD)[[3]](#footnote-2) Diagrammes de séquence système (DSS)[[4]](#footnote-3)  Contrats d’opération5  Diagramme de séquence pour chaque opération du système[[5]](#footnote-4) |
| 2 | 20 | Plan d’itération3 Modèle du domaine (MDD)4  Diagrammes de séquence système (DSS)5 Contrats d’opération5  Diagramme de séquence pour chaque opération du système6 |
| 3 | 30 | Plan d’itération3 Modèle du domaine (MDD)4  Diagrammes de séquence système (DSS)5 Contrats d’opération5 Réalisation de cas d’utilisation (RDCU)5 |
| 4 | 40 | Plan d’itération3 Modèle du domaine (MDD)4  Diagrammes de séquence système (DSS)5 Contrats d’opération5 Réalisation de cas d’utilisation (RDCU)5 |

*Par exemple, pour l’itération 1, une équipe peut viser les fonctionnalités CU02 et CU03 (20 points) et ça respectera le minimum. Il y aura un MDD pour l’ensemble des cas d’utilisation CU02 et CU03, au moins un DSS pour CU02 et au moins un DSS pour CU03. Il y aura les diagrammes de séquence pour esquisser chacune des opérations systèmes des DSS.   
  
Pour l’itération 2, l’équipe vise CU04 et CU05 (nouvelles fonctionnalités) ajoutant 20 points pour un total de 40. L’ensemble des artéfacts à remettre comprendra un MDD pour CU02, CU03, CU04 et CU05, au moins un DSS pour chaque fonctionnalité (CU02, CU03, CU04 ,CU05), et les diagrammes de séquence respectant les postconditions des contrats, etc.*

*À l’itération 3, l’équipe doit remettre les RDCU’s correspondant aux fonctionnalités des itérations 1 à 3. On peut commencer avec les diagrammes de séquence existants, et y appliquer les principes GRASP. Veuillez noter que les chargés de laboratoire porteront une attention particulière au réusinage que vous aurez fait au code pour qu’il corresponde aux RDCU.*

Critères d’évaluation de la partie processus de chaque itération (*a*, *b*, *c*, *d*)

|  |  |
| --- | --- |
| **Critère** | **Points** |
| Qualité des artéfacts (processus d’analyse et de conception) | 10 |
| Minimum des points pas respecté (processus itératif) | -5 |
| Aucune nouvelle fonctionnalité démontrable (processus évolutif) | -5 |

Le calcul de la note pour la satisfaction du client est le suivant :

e = ∑(Points X (1 + (Facteuri-1)) / 100

*Par exemple, une équipe réalise une solution qui répond adéquatement aux exigences suivantes :*

*F1, F2, F3, F5, F6, U1, U2, S3*

*Le calcul est donc (5 + 5 + 15 + 10 + 10) \* (1 + (0.025 + 0.025 + 0.3)) =60.75/100*

Si une équipe ne réussit pas à répondre adéquatement à une exigence, elle ne sera pas utilisée dans le calcul.

Pour les notes dépassant 100%, le maximum est 120%. Bon laboratoire!

1. Points du tableau des exigences. [↑](#footnote-ref-0)
2. Le plan d’itération est remis et vérifié avant l’itération, sauf pour la première où il est remis au début du second laboratoire. [↑](#footnote-ref-1)
3. Le modèle du domaine doit être cohérent pour tous les cas d’utilisation depuis le début de la première itération. Voir la figure 2.4 pour plus d’informations. [↑](#footnote-ref-2)
4. Il doit y avoir un DSS (et un contrat d’opération et une RDCU, le cas échéant) pour chaque cas d’utilisation analysé depuis le début de la première itération. [↑](#footnote-ref-3)
5. Selon la notation du chapitre F14/A15 du livre de Larman. Dans l’itération 2, les postconditions du contrat doivent être satisfaites par le diagramme. [↑](#footnote-ref-4)