

Activité d'intégration - Projet de développement Web

Contenu

Activité d'intégration - Projet de développement Web.....	1
Introduction.....	1
Les objectifs de l'UE.....	2
Thème du projet.....	2
Gestion du projet.....	3
Organisation	3
Déroulement du projet : des sprints et des meetings.....	4
Contraintes du développement	7
Rapport.....	7
Évaluation.....	8
Dates.....	8
Les sources de l'application.....	8
Le rapport	8
Documents fournis	8

Introduction

Ce document rappelle dans un premier temps les objectifs du projet, de l'unité d'enseignement.

Le thème du projet est ensuite détaillé; il s'agit de l'ensemble des fonctionnalités de l'application à développer. Celles-ci sont en fait les besoins du client qui sont exprimés en français au développeur; il est du ressort des étudiants de les traduire dans modèles adéquat (DSD et diagrammes de navigation).

La gestion du projet détaille l'organisation pratique du projet. Les contraintes de développement sont exprimées également.

Enfin, le contenu du rapport attendu est précisé ainsi que le détail de l'évaluation.

Les dates d'échéance de l'application ainsi que les consignes de soumissions sont précisées finalement.

Les objectifs de l'UE

Rappelons dans un premier temps les objectifs de l'UE :

- Mettre en œuvre les acquis des UE suivantes : WEB, IANARCH et ANGLAIS
- Plonger les étudiants dans un contexte se rapprochant du milieu professionnel
- Etre responsable d'un projet informatique depuis le cahier des charges jusqu'aux tests de l'application.

Thème du projet

Le projet à développer consiste en une amélioration de l'existante application ianarch.ipl.be. Cette application, pour rappel, permet actuellement aux étudiants de 1^{ère} année en informatique de gestion de l'IPL de tester par le biais d'une interface web des queries sur une base de données dans le cadre du cours d'Introduction à l'Analyse et l'Architecture des Données.

La nouvelle application à développer doit offrir les mêmes fonctionnalités mais doit permettre en plus de retenir les exercices effectués par un étudiant. Elle doit aussi permettre aux professeurs de visualiser les exercices (les réponses) qu'un étudiant a déjà réalisés.

Les exercices sont les questions existantes posées dans le syllabus ; ils sont décomposés en 4 niveaux. Ils sont fournis dans des fichiers CSV par niveau selon le format suivant :
num;theme;enonce;query ;nb.

- num correspond au numéro du query.
- theme contient l'information complémentaire sur le niveau (par exemple : group-by). Il s'agit d'une indication supplémentaire sur la question.
- enonce est la question à résoudre.
- query contient la réponse du query en sql
- nb contient le nombre de lignes résultants de l'exécution du query

Lors de l'importation du fichier CSV, il faut spécifier le niveau de celui-ci. Si ce niveau existe déjà alors l'importation ne peut pas se réaliser.

L'application à développer doit considérer les utilisateurs suivants :

- Les étudiants
- Les professeurs

Les étudiants doivent se connecter à l'application. Un fichier csv (fourni) contiendra le matricule, le nom et le prénom de chaque étudiant selon le format suivant :

Matric_Info ;Nom_Etudiant ;Prenom_Etudiant.

Authentifié, un étudiant peut parcourir les niveaux et choisir la question qu'il désire résoudre. S'il avait déjà tenté cette question (sans erreur), l'ancienne réponse apparaît ; il peut éventuellement la modifier. Quand il le souhaite, l'étudiant peut demander l'exécution de son query (en cliquant sur un bouton par exemple). L'application doit alors tester la réponse de l'étudiant (son query) sur la base de données **ianarch**. La réponse de cette exécution est alors affichée. La réponse de l'étudiant, si elle ne renvoie pas d'erreur, doit être enregistrée. L'application montre également le résultat de

l'exécution du query correct reçu dans le CSV à titre comparatif. Il ne faut retenir qu'une seule réponse par query par étudiant.

Les professeurs doivent se connecter à l'application. Un fichier CSV (fourni) contiendra le login, le nom, le prénom de chaque professeur du cours selon le format suivant : login;nom;prenom.

Authentifié, un professeur peut :

- Voir la liste des étudiants précisant pour chacun d'eux s'il s'est déjà connecté et son pourcentage de progression dans les exercices.
- Visualiser pour un étudiant donné les réponses aux questions.
- Modifier les questions d'un niveau (la question, le numéro, le thème éventuel, le query, l'auteur de la dernière modification et la date de cette dernière modification). Il est demandé de tracer les changements opérés; à savoir il faut retenir quel professeur a effectué le dernier changement et quand. Lors de l'importation des queries du CSV, il n'y a aucun auteur spécifié ni date de modification.
- Exporter les queries sous forme CSV. Il faut permettre de répercuter les changements effectués à la demande dans un fichier CSV. (au même format que celui importé)

Le mot de passe est demandé à l'utilisateur lors de sa première connexion. Le login à introduire pour un étudiant est son matricule, pour un professeur, il s'agit de son login. Après la première connexion, ce mot de passe est enregistré. Il ne pourra pas être modifié dans l'application.

Lors du lancement de l'application, les différents csv sont importés dans le système. Il est évident que les informations non importés sont potentiellement "null" dans la base de données.

Gestion du projet

Organisation

Le projet se déroule sur 3 semaines et chaque semaine comprend 3 séances de 2 heures. Nous vous fournissons le détail du travail de chaque séance.

Les étudiants travaillent en équipe de 2 personnes. Chaque binôme est responsable de tâches prédéfinies et réparties en équipe. **Chaque tâche est sous la responsabilité d'un seul étudiant.** Il est important d'être clair à ce sujet. Chaque étudiant a son rôle dans l'équipe ; les professeurs doivent savoir lequel a réalisé quelle partie (pour l'évaluation ainsi que la défense). Les deux membres de l'équipe doivent comprendre l'entièreté du projet mais chacun n'est responsable que de ses tâches.

Le développement du projet s'inspire de la méthode agile **SCRUM**. Cette méthode de développement découpe le projet en petits morceaux de temps appelés des **sprints** (dure de 1h à 1 mois max). Chaque sprint se termine par un **meeting** durant lequel le développeur démontre ce qui a été fait. Afin d'organiser ce travail, nous vous demandons d'utiliser l'application Trello (<https://trello.com/>). Il s'agit d'une application web et mobile qui gère des tâches. Elle est très simple et définit simplement 3 groupes de tâches :

- A faire
- En cours
- Terminée

Pour plus d'information à ce sujet, consultez éventuellement les articles suivants :

- <http://www.lateamweb.com/trello-to-do-list/>
- <http://format30.com/2014/08/26/etudiants-organisez-votre-travail-avec-le-kanban-personnel-trello/>
- <http://lifehacker.com/how-to-use-trello-to-organize-your-entire-life-1683821040>

Afin de faciliter la lecture de la suite de l'énoncé et éviter toute ambiguïté avec les utilisateurs de la future application, les étudiants seront désormais appelés développeurs.

Chaque développeur doit se créer un compte ou se connecter via son compte google. Ensuite, l'un des membres du groupe doit créer une organisation Il faut ensuite ajouter l'autre développeur comme membre.

On vous demande d'intégrer dans le rapport des printshots de votre board (en fin de la première semaine, de la fin de la seconde semaine et celui de la fin de l'application).

Déroulement du projet : des sprints et des meetings

Dans notre projet, les meetings auront parfois lieu avec un professeur parfois uniquement entre les membres du groupe.

A la fin de chaque séance, les développeurs fixent les tâches de chacun individuellement (dans Trello). Les développeurs travailleront chez eux ces tâches attribuées. En début de séance suivante, ils vérifient ensemble la bonne réalisation de celles-ci.

Le travail (et la présence) de chaque membre du groupe est traqué dans un document appelé *suivi_groupe.docx*. Ce document est complété par un professeur à chaque séance. Il sert de base pour l'évaluation continue du projet.

Il faut également que les développeurs fassent bien attention à ce que l'application ne régresse jamais ; à savoir, il faut que tout ce qui est correct en MEETING 2 le reste en MEETING 3, 4 etc.

Pendant les séances, les professeurs aident les étudiants à structurer leurs raisonnements que ce soit pour les données, l'architecture, le rapport, les tests etc.

Séance 1 - SPRINT 1 :

Les développeurs débutent ensemble par

- l'analyse des données afin de produire le DSD
- l'analyse des fonctionnalités sous forme de prototypes d'interfaces

N'oubliez pas de fixer en fin de séance les tâches à réaliser par chaque développeur pour la prochaine séance dans Trello !

Les tâches proposées sont * :

- achever le DSD et le mettre au propre
- achever les prototypes et les mettre au propre

Séance 2 ou séance 3 - MEETING 1 :

Les développeurs présentent leur DSD et prototypes d'IHM.

Un professeur vérifie la bonne compréhension des données véhiculées dans le DSD et des fonctionnalités attendues dans les prototypes d'IHM. Il prend note des remarques faites aux développeurs dans le document appelé suivi du groupe.

Séance 2 et séance 3 - SPRINT 2 :

Les développeurs implémentent la base de données et récupèrent les scripts de la bd d'ianarch pour les exécuter en local.

Les développeurs réalisent le design de base de l'application MVC OO (pages html avec le menu). Une première ébauche du diagramme de navigation est réalisée.

N'oubliez pas de fixer en fin de séance les tâches à réaliser par chaque développeur pour la prochaine séance dans Trello !

Les tâches proposées sont :

- Implémenter la base de données dans wamp.
- Récupérer et exécuter les scripts d'ianarch (create et insert) dans wamp.
- Effectuer le design de base de l'application avec un début de diagramme de navigation

Séance 3 - MEETING 2 :

Les développeurs valident le travail du groupe et effectuent les ajustements nécessaires.

La répartition en tâches est à définir par les membres du groupe.

Les développeurs doivent effectuer un printshot du board Trello pour l'intégrer dans le rapport.

Séance 4 - SPRINT 3 :

Dans ce sprint, on distingue deux grandes tâches détaillées ci-dessous. Décidez dans votre groupe qui s'attaque à l'une ou l'autre.

Les développeurs implémentent les connexions des différents acteurs (professeur et étudiant) avec des login et password codés en dur. Il s'agit de gérer les actions propres à chaque compte au niveau de l'IHM. Veillez toujours à ce que le diagramme de navigation évolue avec votre application. Pour le développement de cette partie, le login du prof est PROF et le mot de passe est PROF et pour l'étudiant, le login est ETU et le mot de passe est ETU.

* Les tâches peuvent être différentes dans chaque groupe. Il s'agit d'un canevas de base proposé dans ce document. A vous de l'adapter selon la synergie de votre groupe.

Il faut également permettre l'importation des fichiers csv qui contiennent les comptes professeurs et les comptes étudiants. Ceci doit s'effectuer une seule fois dans la "vie" de l'application. Les informations doivent être retenues dans la base de données ; l'usage de classes est approprié.

N'oubliez pas de fixer en fin de séance les tâches à réaliser par chaque développeur pour la prochaine séance dans Trello !

Les tâches proposées sont :

- Gérer les connexions
- Importer les CSV des comptes étudiants et professeurs
- Réaliser (mettre à jour) le diagramme de navigation

Séance 5 ou séance 6 – MEETING 3 :

Les développeurs mettent en commun les deux parties. Les comptes PROF et ETU ne doivent plus exister ; il faut que l'application fonctionne avec les comptes importés.

Les professeurs vérifient la gestion des connexions et l'importation des CSV ainsi que le diagramme de navigation de l'application.

La répartition en tâches pour le sprint suivant est à définir par les membres du groupe.

Séance 6 – SPRINT 4 :

Les développeurs implémentent l'exécution des queries pour les étudiants dans les bd1, bd2 et bd3 (sans affichage de la bonne réponse).

Ils doivent également permettre aux professeurs qui se connectent d'importer une liste csv pour les queries d'un certain niveau et de voir la liste des étudiants précisant pour chacun d'eux s'il s'est déjà connecté ainsi que son pourcentage de progression dans les exercices.

Les tâches proposées sont :

- Importer les queries à partir d'un CSV. Le séquençement de la numérotation des queries du CSV ne doit pas être validé par l'application.
- Développer l'exécution des queries pour les étudiants (sans affichage de la bonne réponse).
- Développer la liste des étudiants vue par les professeurs
- Tenir à jour le diagramme de navigation

Les développeurs doivent effectuer un printshot du board Trello pour l'intégrer dans le rapport.

Séance 7 – MEETING 4 :

Les développeurs valident le travail du groupe, mettent en commun, testent et effectuent les ajustements nécessaires.

Les tâches proposées sont :

- Tester la partie développée par l'autre membre du groupe.
- Valider les pourcentages de progression affichés

Séance 7 et séance 8 - SPRINT 5 :

Voici les derniers points à développer :

- Les étudiants qui se connectent à l'application doivent pouvoir voir la dernière réponse (sans erreur) qu'ils ont donnée pour un query.
- Les étudiants doivent pouvoir voir les bonnes réponses du query.
- Les professeurs authentifiés doivent pouvoir modifier les questions d'un niveau (la question, le numéro, le thème éventuel, le query, l'auteur de la dernière modification et la date de cette dernière modification). Il est demandé de tracer les changements opérés; à savoir il faut retenir quel professeur a effectué le dernier changement et quand.
- Exporter les queries sous forme CSV. Il faut permettre de répercuter les changements effectués à la demande dans un fichier CSV. (au même format que celui importé)

La répartition en tâches est à définir par les membres du groupe.

Séance 8 ou séance 9 - MEETING 5

Les développeurs doivent effectuer un dernier printshot du board Trello pour l'intégrer dans le rapport. Il s'agit du board final de l'application.

La répartition en tâches restantes est à définir par les membres du groupe.

Les professeurs vérifient l'état général de l'application.

Contraintes du développement

Les étudiants doivent fournir une application PHP/MySQL/HTML5.

Il est impératif de respecter le design MVC OO vu au cours de PHP. Les pages HTML doivent être valides selon la norme HTML5. La base de données doit respecter les règles et conventions du cours d'IANARCH.

Les étudiants doivent utiliser 2 bases de données : celle en provenance du cours d'ianarch (bd1, bd2 et bd3) et leur base de données qui gèrera les données de l'application. Ces deux BDS sont stockées sur le serveur local de Wamp.

Aucun script (js ou autres) n'est permis !

L'usage d'un framework tel que Bootstrap pour les css est toléré mais ne sera pas considéré dans l'évaluation.

Les étudiants rédigent les commentaires et le code en anglais.

Rapport

Chaque groupe de développeurs doit soumettre un rapport relatant différents éléments du projet. Le format de ce rapport est imposé par le document *rapport_groupe.docx*.

On vous demande de modifier le nom du document en spécifiant les noms des développeurs du groupe selon la convention suivante : *rapport_nom1_nom2.docx*. *nom1* et *nom2* sont les noms de famille des développeurs en minuscule sans aucun espace ni caractères spéciaux et dans l'ordre alphabétique.

Évaluation

La présence en séance de cours est obligatoire et aura une incidence directe la note individuelle du développeur qui est absent.

Les évaluations des parties individuelles et communes impacteront la note individuelle de chaque membre du groupe ; les développeurs sont évalués individuellement.

Le document *suivi_groupe.docx* sert de base à l'évaluation continue. Le rapport impactera également cette note.

L'examen oral permettra de valider que chaque développeur est réellement développeur de l'application. Cet examen consistera en une démonstration des fonctionnalités. Les professeurs demanderont également à l'étudiant d'effectuer une petite modification dans l'application.

Evaluation continue	50 %
Rapport	25%
Suivi	25%
Examen oral	50 %

Dates

Les sources de l'application

Les sources du projet doivent être soumises au plus tard sur ecampus au plus tard le **vendredi 15 mai à 15h**.

Les sources doivent se trouver dans un répertoire dont le nom est *projet_nom1_nom2* selon les mêmes conventions que pour le rapport. Ce répertoire doit être compressé (.zip) avant d'être soumis sur ecampus.

Le rapport

Le rapport doit être soumis (version pdf) sur ecampus au plus tard le **lundi 18 mai à 10h**.

Documents fournis

- etudiants.csv
- professeurs.csv
- projet.docx (ce présent document)
- queries_1.csv
- rapport_groupe.docx
- suivi_groupe.docx

En plus, il y a la base de données ianarch (voir ecampus/ianarch).