

Electif Technologies Web 2.0

Cadrage mini-projet 2024

Sommaire

[Sommaire](#)

[Présentation et Objectifs](#)

[Organisation](#)

[Sujets 2023](#)

[Livrables intermédiaires](#)

[Livrable 1 : Mockups & Cahier des charges](#)

[Livrable 2 : Système d'information](#)

[Livrable 3 : Architecture technique et découpage en tâches](#)

[Soutenances](#)

[Evaluation](#)

Présentation et Objectifs

L'objectif de ce mini-projet est de vous donner l'occasion de développer une application web complète, de la conception à la réalisation. L'intérêt est de découvrir des problématiques différentes de celles évoquées en cours, et de développer vos compétences de travail en équipe et de gestion de projet.

Organisation

Les équipes sont constituées de 4 personnes au minimum, 5 au maximum. Il ne faut pas plus de 8 groupes.

Vous complétez un google sheet ([Groupes miniprojets 2024](#)) pour définir vos projets lors de la première séance.

Vous rendrez compte de votre travail en maintenant un répertoire google drive ([groupes 2024](#)) pour votre projet, contenant tous les documents de votre projet, en respectant l'arborescence suivante :

- Google doc intitulé "minutes", dans lequel vous reporterez votre travail en temps réel, en indiquant la date, la durée et les tâches menées ;

- Google doc intitulé “livrable 1”, dans lequel vous présenterez les mockups de votre application ;
- Google doc intitulé “livrable 2”, dans lequel vous présenterez le système d’information de l’application ;
- Google doc intitulé “livrable 3”, dans lequel vous présenterez l’architecture technique de votre application, le découpage en modules, la répartition des modules aux étudiants ;
- Répertoire intitulé “livraison finale”. Il contiendra notamment le code de l’application, ainsi qu’un google slide intitulé “soutenance” qui servira de support à votre soutenance ;
- Éventuellement, un répertoire intitulé “archives” permettant d’organiser vos ressources.

Vous ferez valider, en séance de TEA, après chaque partie votre travail par un enseignant, qui reportera ses remarques et la date de validation dans le google doc.

Sujets 2024

Un sujet est proposé cette année : “blabla2”. Il est présenté dans ce document : [Sujet Mini-projets TWE 2024](#).

Livrables intermédiaires

Livrable 1 : Mockups & Cahier des charges

Vous présenterez le cahier des charges de votre application sous forme d’une liste de fonctionnalités, permettant d’identifier les “interfaces” du site. Une interface peut correspondre à plusieurs pages Web suivant les cas. Il faut vous concentrer pour l’instant sur le “fonctionnel” en faisant abstraction des problèmes techniques.

Pour chaque interface, vous proposerez un dessin d’écran. Bien entendu, un dessin seul ne suffit pas. Il faut y associer quelques commentaires permettant de comprendre les services rendus par cette interface, quand elle n’est pas évidente (par exemple, tout le monde comprend ce qu’est un formulaire de login, pas la peine de l’expliquer...)

Pour produire vos mockups, vous pouvez utiliser par exemple balsamiq Mockups ou tout autre outil dédié (cf. [prototyping-tools-compared](#) pour une liste de suggestions)

Vous rédigerez votre cahier des charges au format pdf. Plus votre cahier des charges est clair, plus il sera facile d’en déduire la base de données à produire et plus le développement sera “pilote par le besoin”.

Vous prendrez soin de questionner l'expérience utilisateur (UX) de vos propositions d'écran, afin d'éliminer les "cliquodromes" et de converger vers des écrans dotés d'une bonne ergonomie.

Il ne faut pas confondre ergonomie et design :

- L'ergonomie (UX en anglais), c'est un savant mariage entre "utilité" et "utilisabilité". On peut produire des outils très utilisables, mais inutiles. On peut produire des outils très utiles, mais inutilisables.
- Le design, c'est l'esthétique du site. Cela sera traité plus tard au moment de la rédaction de la charte graphique.

On vous suggère de présenter vos interfaces à vos camarades, en leur demandant de réaliser des opérations sur ce site fictif, pour "sentir" quels seront leurs réflexes et faire apparaître des problématiques d'UX.

Livrable 2 : Système d'information

Vous développerez un modèle conceptuel de données ainsi qu'un modèle physique permettant de couvrir les besoins exprimés plus haut.

Vous présenterez votre modèle physique en faisant apparaître des exemples de données réelles permettant de vérifier que votre modèle est effectivement suffisant pour enregistrer les informations requises. Pour cela, vous déposerez sur le drive une archive contenant la base de données et quelques données au format SQL, extraite à partir de phpMyAdmin, ainsi qu'un document expliquant les relations entre les tables, la signification des champs particuliers (pas la peine d'expliquer ce qu'est le champ 'login' , par contre, un champ désignant le chemin d'une image, d'un fichier pdf, d'une vidéo, etc. doit être explicité clairement) et la justification du choix de leur taille.

Si vous accédez à des données provenant d'une autre base (Hyperplanning par exemple), leur structure devra être explicitée.

Les requêtes SQL les plus complexes (avec jointures) devront être présentées au format SQL de manière à valider la pertinence du choix de la base de données.

On indiquera aussi lorsque cela s'y prête, les flux json manipulés par les interfaces, et toute autre source de données exploitée par votre application, en présentant sa structure.

Livrable 3 : Architecture technique et découpage en tâches

Vous définirez l'arborescence et l'architecture de votre application. Pour cela, découpez votre projet en modules, et indiquez la répartition de ces modules aux développeurs.

Par exemple, dans le cas d'une application de galerie d'images, il est possible de séparer les pages du module "backOffice" permettant la création des utilisateurs, conversations, dépôt de fichiers image, et les pages du module "frontOffice" où les utilisateurs se loguent et visualisent les galeries, etc.

Votre livrable devra également contenir des informations sur la manière dont vous organiserez vos fichiers utilisateurs (par exemple si vous enregistrez des images, pdf ou vidéos, vous indiquerez où ils seront stockés, les conventions de nommage des sous-répertoires éventuels, et ce que vous stockez dans la base de données), ainsi que vos fichiers sources : répertoire pour les css, les librairies javascript, les librairies php...

Définir également les librairies de fonctions qui vous seront utiles pour réaliser le code : fichier, noms et prototypes des fonctions, afin d'éviter que deux développeurs développent la même fonctionnalité pour deux modules différents.

Si votre projet vous amène à manipuler des technologies "non classiques" : envoi de mail, sms, HTML5, objets multimédia, tâches planifiées, etc. vous indiquerez la solution qui a été choisie et comment elle fonctionne.

Définir la charte graphique en CSS des blocs fonctionnels demandés, et indiquer leur structure XHTML quand ils sont complexes.

Il est important de NE PAS COMMENCER un développement séparé SANS avoir PRÉPARÉ le découpage des tâches, notamment le fichier modele.php. Il faudra faire état de ce découpage à l'occasion de la livraison, ainsi que dans les cartouches de votre code.

L'objectif de ce livrable est de bien anticiper la phase de développement de votre projet, en préparant le découpage des tâches en modules et l'intégration des modules dans l'application finale.

Nous vous conseillons de commencer par lister les différentes fonctions qui vous seront utiles pour réaliser votre application, et les répartir sous forme de librairies, à la manière de ce qui a été proposé pour les Tps de php.

Par exemple, vous listerez précisément toutes les fonctions qui génèrent des requêtes dans la base de données, en indiquant leurs arguments d'appel, et ce qu'elles renvoient (typiquement, un tableau associatif ou un objet JSON). De cette manière, vous pourrez vous répartir les tâches, l'un des membres de l'équipe se chargeant de rédiger ces fonctions, un autre de les utiliser pour produire l'interface finale.

Ce travail préparatoire doit être fait à la fois pour le code côté client et serveur. Veillez également à présenter les variables d'entrées-sorties des interfaces (i.e. les noms des variables de session, les noms des variables apparaissant dans les chaînes de requête), de manière que vous puissiez vous répartir le développement des interfaces et faciliter l'intégration finale.

Prenez soin de développer des conventions de nommage claires de manière que le code généré soit homogène, et ne pose pas de problèmes d'intégration. Il faudra bien indiquer dans chaque fichier quel est l'auteur de ce travail, pour faciliter le débogage et l'évaluation.

Soutenances

Les soutenances auront lieu lors de la dernière séance de l'électif planifiée dans votre emploi du temps, soit le **28 juin matin**. Un mail sera envoyé quelques jours avant pour rappeler cette échéance et l'ordre de passage des groupes. **Tout le monde assiste à toutes les soutenances.**

La présentation durera **vingt minutes, suivie de 10 minutes de questions**. Il s'agira de la présentation du cahier des charges de l'application, de son architecture, suivis obligatoirement d'une démonstration. On vous demande de préparer vos slides à l'aide de google slides pour accompagner cette présentation (**typiquement, un slide cahier des charges, un slide architecture, quelques slides techniques sur les "points durs" techniques de votre projet, une démo, un slide gestion de projet, une conclusion/perspectives**), et de placer vos slides dans le google drive de votre projet.

Il faudra remettre dans le répertoire intitulé "livraison finale" sur le drive, la version finale des documents produits pour le projet (y compris les documents de conception - livrables 1 à 3-mis à jour) :

- Un rapport de fin de projet, contenant typiquement :
 - Les mockups et leur description, avec en dessous de chacun d'eux une explication de ce qui a été réellement réalisé,
 - La structure de la base de données sous la forme d'un schéma (montrant les relations clé étrangères/clés primaires)
 - Une description de l'architecture de votre solution (MVC, organisation des fichiers, choix réalisés en présentant la problématique, et les choix entrepris pour la résoudre)
 - Une présentation de la répartition des tâches entre les développeurs
 - Une conclusion et des perspectives sur votre projet, ainsi que votre avis (constructif !) sur la matière TW2.
- Une archive contenant le code (ainsi que la base de données, et un fichier texte "INSTALL" décrivant les opérations d'installation et les identifiants à utiliser)
 - N'oubliez pas d'indiquer vos sources lorsque vous intégrez des outils existants, et la manière dont ils fonctionnent : recopier sans comprendre est MAL !

Evaluation

L'évaluation portera sur la qualité des documents produits au fur et à mesure, leur date de dépôt, ainsi que sur la qualité de l'application finale et le respect du cahier des charges initial. Un indicateur de difficulté sera indiqué pour nuancer les évaluations.

Des critères d'évaluation des mini-projets sont indiqués sur moodle.

Ce qui sera déposé lors de la livraison finale devra contenir tout ce qu'il faut pour le réinstaller sur une autre machine (export de la base de données, indications concernant la procédure d'installation...).

Vérifiez bien que vos choix techniques permettent d'installer votre solution dans un répertoire différent, avec un nom de base de données différent, en procédant à peu d'ajustements (typiquement, juste la modification du fichier de configuration). Pour cela, tester votre procédure d'installation sur un autre PC, sous un autre répertoire.

Bon travail !