

UE 2 : programmation d'applications web dynamiques

Bachelor IMT Dakar

3^{ème} année

2023-24

Bruno Defude, Telecom SudParis

Bruno.defude@telecom-sudparis.eu

Organisation

- UE projet transverse programmation d'applications web dynamiques
 - M1 : construction d'interfaces utilisateurs en HTML-CSS-javascript
 - M2 : programmation d'application web en PHP

Objectifs de l'UE 2

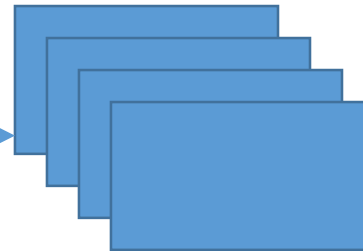
- Modéliser un schéma de BD
- Modéliser une application interactive
- Construire une interface utilisateur en utilisant les technologies du web (HTML, CSS, javascript)
- Adaptation de l'interface à divers terminaux (PC et smartphone)
- Analyser l'ergonomie de l'interface produite
- Apprendre les bases de la programmation en PHP pour le web
- Utiliser les mécanismes de cookie et de variable session
- Interagir avec un SGBD à partir de PHP
- Coder une application agence de voyage en PHP pour le web

UE 2

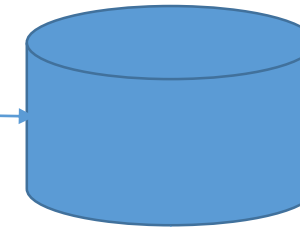
Présentation

A screenshot of a web browser window showing a form titled "Ajouter une adresse". The form contains several input fields: "Organisation", "Nom" (with a sub-label "Ajouter un nom"), "Adresse postale", "Code postal", "Ville", "CEDEX", "Pays/Région" (a dropdown menu), "Téléphone" (with a sub-label "Saisir num tél"), and "E-mail" (with a sub-label "Ajoutez une adresse e-mail"). There are "Annuler" and "OK" buttons at the bottom right.

Logique métier



Données



MySQL

HTTP

Requêtes SQL

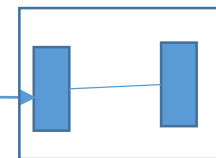
Programmes PHP

Requêtes SQL

Formulaires web
HTML, CSS, javascript

Looping-mcd

Schéma
relationnel



Code
SQL

Conception de BD

Technologies utilisées

- Conception de schéma de BD : looping-mcd
- SGBD : MySQL, phpmyadmin, SQL
- Langages de programmation : PHP, javascript
- Web : HTML, CSS, javascript

Projet agence de voyage

- C'est un projet **professionnalisant**
- Vous n'êtes pas que des étudiants mais de futurs stagiaires et salariés en entreprise
- Ne pas se contenter du minimum mais se mettre dans une logique métier (ce que je fais doit servir à des utilisateurs/clients)
- Si cela a du sens, on peut faire plus que dans l'énoncé
- L'important est de faire soi même, même si moins abouti que ce que l'on peut récupérer d'Internet
- Penser aussi à la qualité de ce que vous faites (est ce maintenable, évolutif, compréhensible par d'autres)

Schéma BD UE2

- Clients(noClient, nom, prenom, email, mdp)
- Billets(idBillet, dateAchat, prix, **idVol**, **idClient**) : idVol clé étrangère sur Vols, idClient clé étrangère sur Client
- Vols(idVol, numVol, dateVol, jourSemaine, heureDepart, heureArrivee, distance, **noCompagnie**, **aeroportDepart**, **aeroportArrivee**) : noCompagnie clé étrangère sur Compagnies, aeroportDepart et aeroportArrivee clés étrangères sur Aeroports
- Aeroports(IATA code, nomAeroport)
- Compagnies(noCompagnie, nomCompagnie)
- À modéliser avec looping-mcd, créer la BD (code SQL fourni) et la peupler (quelques données fournies à titre d'exemple)

construction d'interfaces utilisateurs en HTML-CSS-javascript

- On dispose déjà de la BD et il faut « imaginer » une application web qui interagisse avec cette BD (sans la modifier, juste en l'adaptant si besoin)
- Application = ensemble d'opérations offertes à différents acteurs, en général interactive (interactions avec les utilisateurs)
- Application web = application interactive déployée dans un navigateur web et utilisant les technologies web (html, http, js, css) et un langage de programmation associé (php, js, java, python, ...)

Démarche de conception d'une application web

- Identification des acteurs
- Identification des opérations pour chaque acteur
- Spécification du schéma de la BD nécessaire pour ces opérations (pas nécessaire ici car on a déjà la BD)
- Spécification des interactions entre les utilisateurs et l'application (au niveau de chaque opération et au niveau de l'application globalement) : graphe de navigation
- Choix de l'interface : structure d'une page, bandeau de navigation, aspects graphiques
- Codage de l'interface : html, CSS
- Codage des opérations applicatives : php, js, python, java, ...

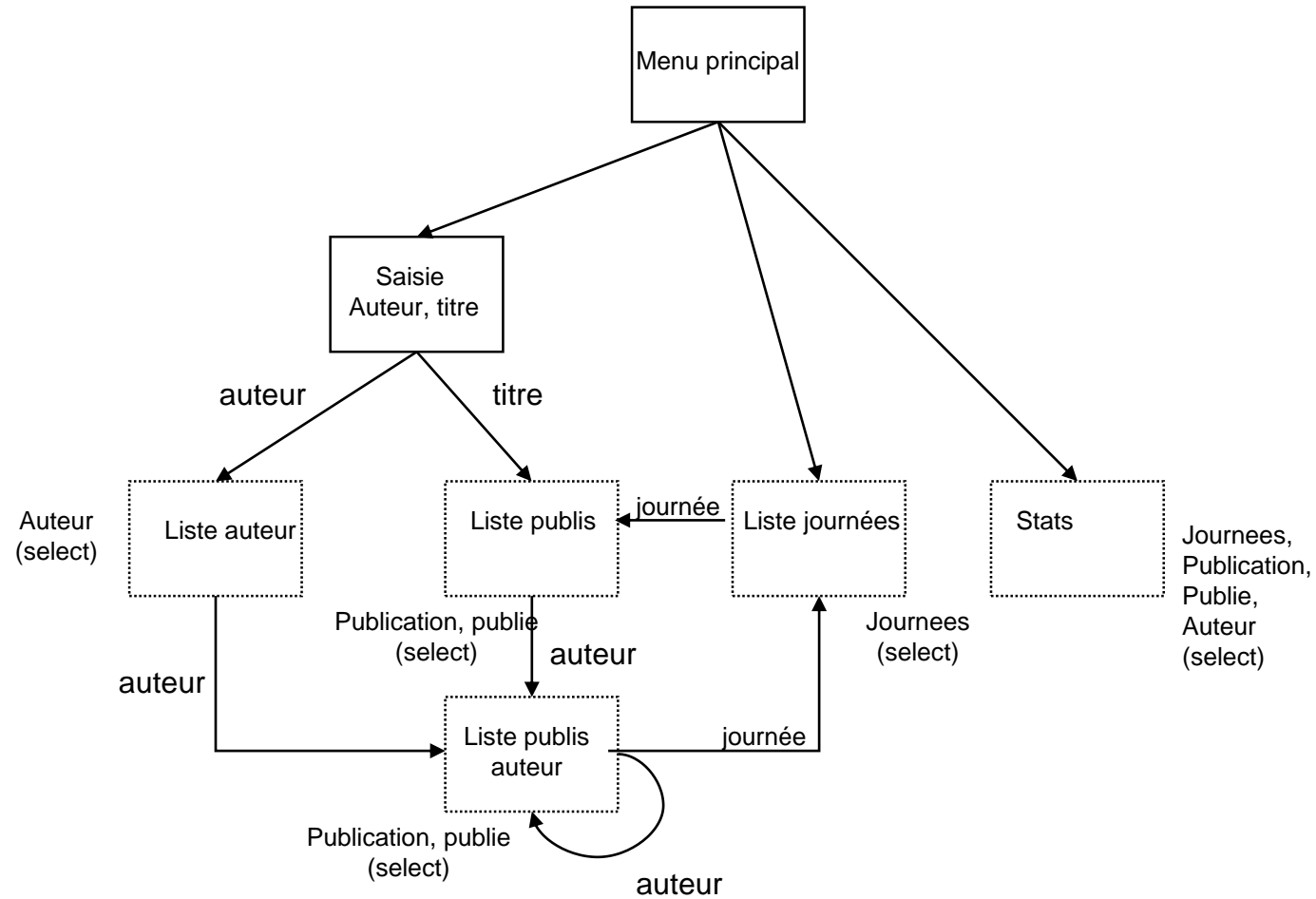
Identification des acteurs

- Il y a plusieurs acteurs concernés par cette application
 - Administrateur de l'application (peut ajouter des compagnies, rajouter/supprimer des vols, ...)
 - Client (peut acheter des billets)
 - Utilisateur non authentifié qui peut juste naviguer dans les vols proposés à la vente

Identification des opérations

- Pour chaque catégorie d'utilisateur et en analysant le schéma de la BD, identifier précisément quelles sont les fonctions qu'il/elle peut exécuter
 - Acheter billet
 - Ajouter vol
 - ...
- Pour chaque fonction, définir la suite d'opérations à faire en précisant
 - Quelles données sont échangées entre l'utilisateur et l'application
 - Quelles sont les données de la BD qui sont accédées/modifiées

Graphe de navigation



Intérêt du graphe

- Détecter des sous graphes isolés
- Avoir une idée de la longueur des chemins (longueur = nombre de clics à faire)
- Vérifier que chaque nœud dispose des bonnes informations pour fonctionner

Choix de l'interface

- Définir la structure type d'une page :
 - Bandeau de navigation
 - Place des logos
 - Zones de saisie
 - Zones d'affichage
 - ...
- Définition du visuel
 - Couleurs
 - Fontes
 - ...

Critères à prendre en compte dans l'interface

- Terminaux cibles : écrans d'ordinateurs, smartphones, tablettes, ...
 - Peut déterminer la structure de la présentation, influence le choix des technologies (responsive design)
- Accessibilité : penser aux utilisateurs ayant des déficiences visuelles (daltoniens, faible vision, ...)
- Expérience utilisateur : combien d'actions avant d'arriver à réaliser une opération, éviter de demander plusieurs fois la même chose, comment se repérer dans l'interface, « régularité » de l'interface

Codage de l'interface

- définition des feuilles de style
- Définition des pages (html, CSS)
- En option
 - Choix d'une solution responsive design (bootstrap ou autres)
 - Ajout d'aspects dynamiques (vérification de la saisie en javascript, listes dynamiques en ajax jquery, ...)

Codage des opérations applicatives

- Utilisation du langage php (déjà installé avec wamp/xampp ou équivalent)
- Beaucoup d'autres langages sont possibles
- Pour un codage plus professionnel et s'appuyant sur des services de haut niveau, on peut utiliser des frameworks
 - symfony pour php
 - Django pour python
 - Angular ou react pour javascript
 - ...
- Frameworks sont cependant difficiles à prendre en main et nécessitent un apprentissage assez long (plutôt adapté pour de gros projets)

Codage d'une application php

- Besoin d'accéder à une base de données (utilisation de l'API PDO ou d'un ORM comme doctrine)
- Besoin de générer du html (à la main ou utilisation de moteur de template comme twig)
- Besoin de créer et gérer des formulaires de saisie (à la main ou utilisation d'un générateur)
- besoin de gérer de l'authentification et des sessions utilisateurs (à la main ou utilisation d'un framework)

Gestion de session

- Réécriture d'URL : passage de variables en mode formulaire GET
- Variables sessions : scripts serveurs
- Cookies : stockage côté client de variables
- Stockage BD : enregistrement du contexte dans une BD identifié par un identifiant de session

Programmation de sessions

- Script login.php ouverture d'une session avec stockage de 3 variables
 - Idf = valeur numérique
 - Nom = chaîne
 - Prénom = chaîne
- Scripts traite.php
 - Test existence session
 - Affichage nom+prénom
- Script logout.php
 - Fin de la session en cours

Solution cookie

- Script login.php

```
Setcookie(idf, "1234", now()+600, "/", ".telecom-sudparis.eu", 0);
Setcookie(nom, "defude", now()+600, "/", ".telecom-sudparis.eu", 0);
Setcookie(prenom, "bruno", now()+600, "/", ".telecom-sudparis.eu", 0);
```
- Script traite.php

```
if (!isset($_COOKIE['idf'])) { print 'erreur session non démarrée'; }
else {
print $_COOKIE['nom'].'-'. $_COOKIE['prenom'];
```
- Script logout.php

```
Setcookie(idf, "", 0, "", "", 0);
Setcookie(nom, "", 0, "", "", 0);
Setcookie(prenom, "", "", "", 0);
```

Solution session

- Script login.php

```
session_start();
$_SESSION['idf']=1234;
$_SESSION['nom']='defude';
$_SESSION['prenom']='bruno';
```
- Script traite.php

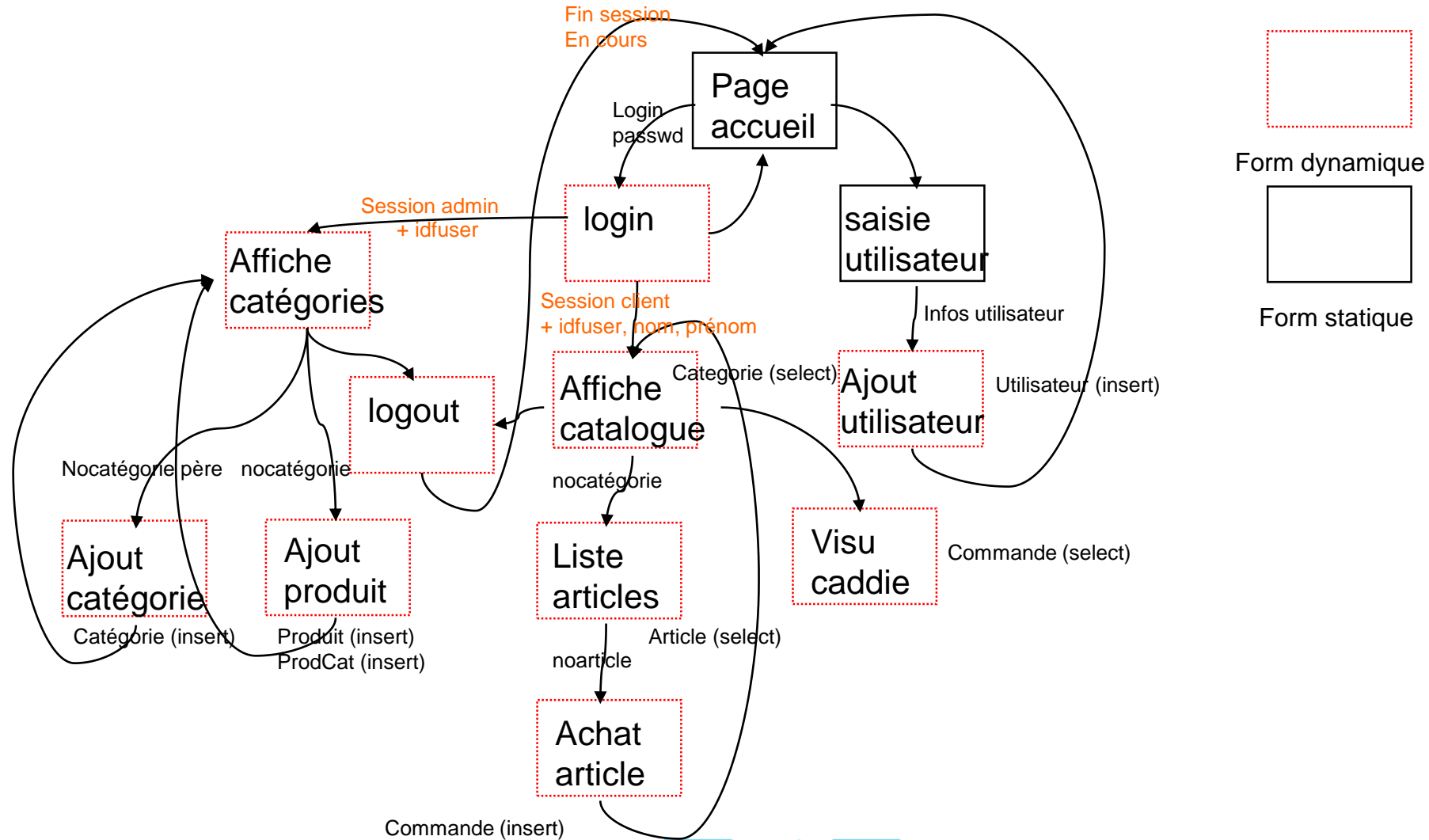
```
session_start();
if (!isset($_SESSION['idf'])) { print 'erreur session non démarrée';}
else {
print $_SESSION ['nom'].'-'. $_SESSION['prenom'];}
```
- Script logout.php

```
session_destroy();
```

Comparaison des modes de gestion de session

Mode	Inconvénient	avantage
Réécriture URL	En clair Intrusif Juste valeurs	simple
Cookies	Juste valeurs Global au navigateur	simple
Variables session	Volume limité	Valeurs + références
Stockage BD	Prog. Plus élaborée	Gros volume

Modélisation d'une application web



Hypothèses de travail

- Il existe un utilisateur administrateur qui a tous les droits et dont on connaît le email et password (pas besoin de le stocker dans la BD)
- Les clients sont créés par eux-mêmes (fait partie de l'application)
- Les vendeurs sont créés exclusivement par l'administrateur
- Les produits sont créés/modifiés/supprimés exclusivement par l'administrateur

Processus de développement en PHP

- Mise en place de l'authentification / création de clients
 - Connexion sur un compte existant
 - Déconnexion
 - Ajout d'un nouveau client
- Mise en place de la gestion des sessions
 - Choix d'une technique de gestion de session
 - Codage des sessions et vérification qu'un utilisateur est bien connecté avec les bons privilèges pour chaque page qui le nécessite
- Codage de chaque opération
 - Choix de ce qui est statique (connu à l'avance) et de ce qui est dynamique (dépend d'une entrée utilisateur)
 - Bien déterminer les informations à échanger entre pages (si nécessaire)
 - Bien déterminer quelles sont les données qui vont être accédées par l'opération
- Test
 - Fonctionnel : penser test unitaire (opération par opération) puis test global
 - Non fonctionnel : évaluer les critères à prendre en compte dans l'interface
- Documentation
 - Une documentation pour les utilisateurs de l'application
 - Une documentation pour des développeurs qui reprendraient le projet pour l'améliorer

Liens utiles

- Ressources du projet :
<https://drive.google.com/drive/folders/1mNiisCOe5EKH4iMdFGt2eXmlpELICsPC?usp=sharing>
- Avec notamment le déroulé du projet semaine par semaine
 - Les données fournies
 - Des exemples de code python + SQL

Liens utiles

- Ressources du projet :
<https://drive.google.com/drive/folders/1mNiisCOe5EKH4iMdFGt2eXmIpELICsPC?usp=sharing>
- Avec notamment le déroulé du projet semaine par semaine
 - Schéma BD en SQL fourni
 - des données fournies en SQL

Bon travail!