CENTRE DE FORMATION PROFESSIONNELLE MARIE-ROLLET

Cahier de l'élève
Tache 2
(GUIDE DE PROJET)

Soutien informatique
5229

Programmation d'un utilitaire
462068

Version 1N (PHP)
Révisé le : 2014-09-16

Durée de la tâche : 15h	**Seuil de réussite : 30/40**

1. **DIRECTIVES**

1.1 Vous devez débuter votre projet de programmation après avoir complété le bloc 17 ou après avoir atteint 100h dans les activités du module

1.2 Vous devez compléter votre projet de façon individuelle sous la supervision d'un enseignant[1] du plateau

1.3 Toute communication et toute forme d'aide est autorisée sous la supervision de l'enseignant pendant la réalisation de votre projet

1.4 S'il se produit un bris en cours d'épreuve, vous devez avertir l'enseignant du plateau

1.5 Vous avez la responsabilité de sauvegarder vos données et votre projet. Aucun délai supplémentaire ne sera accordé à la suite d'une perte d'information

1.6 Vous devez respecter l'ordre normal du déroulement de la réalisation du projet

1.7 Vous devez informer l'enseignant du plateau de l'avancement de votre projet (verbalement ou par courriel) et fournir l'ensemble des fichiers s'il en fait la demande

1.8 Vous devez personnaliser les fonctions et la mise en forme

1.9 Lors de la correction, si deux projets sont identiques, les auteurs se verront attribuer la mention ÉCHEC pour cause de plagiat. Chaque candidat doit adapter et taper son code pour son projet

1.10 Tout manquement grave aux règles de sécurité entraîne l'arrêt immédiat de l'épreuve et, par conséquent, l'échec

1.11 Le seuil de réussite est de 30/40

1.12 Advenant un échec, vous recevrez une liste de tous les manquements qui devront être corrigés. Vous aurez un délai maximum de 10 jours ouvrables pour faire les corrections et déposer de nouveau votre projet. Après ce délai, si toutes les corrections demandées n'ont pas été effectuées alors le verdict final sera un échec et vous devrez vous réinscrire

[1] Dans ce document, la forme masculine désigne aussi bien les femmes que les hommes.

2. INFORMATION SUR LA NOTATION

2. Les points alloués pour cette épreuve se répartissent de la façon suivante.

ANALYSE DES BESOINS	
Reconnaissance du type de données devant être traitées par le programme et des traitements à effectuer :	10 points
RÈGLES DE PROGRAMMATION	
Respect des règles de programmation structurée et orientée objet :	10 points
Traduction juste de l'algorithme :	20 points
Total	**/ 40**

3. DESCRIPTION DE L'ÉPREUVE

3.1 Description de la tâche

Votre tâche consiste à développer un programme utilitaire pour la gestion de la liste de contacts à partir d'un projet initial tout en respectant les étapes ci-dessous :

1) Faire la lecture du projet à réaliser
2) Élaborer un schéma des interfaces graphiques
3) Décrire le déroulement de l'application sous forme de pseudo-code
4) Programmer
5) Tester le programme
6) Faire la documentation en lien avec le projet
7) Assurer la distribution de votre projet

3.2 Étapes du déroulement de l'épreuve et de l'évaluation

Étape 1 *(Lecture et analyse du projet)*

À cette étape, vous devez faire une lecture et une analyse du projet afin de vous s'assurer de bien comprendre avant de commencer le travail. A partir de croquis des différentes interfaces et de la description générale de l'application vous devez analyser le projet de programmation pour vous l'approprier. À cette étape vous devez aussi planifier les tables nécessaires au projet. Pour compléter cette étape vous devez :

- Lire la description détaillée du projet
- Récupérer les fichiers du projet initial sur le site du cours
- Créer les tables avec les champs nécessaires pour l'enregistrement des demandes

Étape 2 *(Esquisser les interfaces, voir les entrées / sorties)*

À l'aide d'un traitement de texte, d'un logiciel de dessin, d'un logiciel de modélisation d'objets, vous devez esquisser les interfaces graphiques avec les détails de saisie pour les utilisateurs. (ex.: case à cocher, liste déroulante, zone de texte, bouton etc..) Identifiez les éléments graphiques qui auront une incidence sur l'exécution du programme. (entrées/sorties)

- Faire les schémas des principales interfaces du système (affichage de la liste, l'ajout et la modification des demandes)
- Identifier les champs de saisies et les zones d'affichage

Étape 3 *(Pseudo-code ou commentaire de programmation)*

À cette étape, vous devez décrire le déroulement du programme sous forme de pseudo-code. Dans l'entête du pseudo-code vous devez écrire votre nom, la date de création du pseudo-code et le but de l'application. Au début de chaque structure de code (fonction et/ou fichier) vous devez identifier les **entrées** et les **sorties.** Vous devez déterminer les types de variables qui sont nécessaires pour le bon fonctionnement du programme. Après avoir identifié les entrées et les sorties, vous devez décrire le déroulement de l'exécution à l'aide d'instructions pseudo-code. (voir règles de base)

Règles de base

1) Numéroter chaque ligne d'instruction
2) Utiliser un verbe d'action au début de chaque ligne pseudo-code suivi d'une précision sur l'instruction.
3) Déclarer les variables et les types utilisés
4) Mettre en retrait les conditions et les boucles.
5) Identifier les formulaires et les blocs de code
6) Identifier les noms des variables et des contrôles selon les consignes vues en classe.
7) Bien identifier les débuts et les fins des procédures.

Étape 4 *(Programmer)*

Codage du programme. Cette étape consiste à traduire les instructions de l'ordinogramme pseudo-code dans un langage de programmation à l'aide d'un environnement de développement. Le codage se divise en plusieurs étapes :

- Élaborer l'interface graphique (formulaire)
- Inscrire au début du code votre nom, la date de création de l'application et son but.
- Nommer les contrôles et déclarer les variables selon les règles établies (voir règles pour les noms des contrôles et des variables)
  - Choisir les bons types de variables
  - Choisir des noms valides et significatifs
  - Déclarer les variables aux endroits appropriés
- Traduire chaque instruction dans le langage de programmation
- Commenter chaque instruction programme selon les critères vues en classe
- Mettre en retrait les instructions à l'intérieur des conditions et des boucles de programme
- **Sauvegarder** régulièrement sur un lecteur réseau, sur un média amovible et aussi localement sur le disque interne. Enregistrer à chaque cours l'avancement de votre projet sur différents moyens de sauvegarde. (**Aucun délai ne sera accordé pour la perte de données**)

Étape 5 *(Tester le programme)*

En programmation WEB, la vérification de l'application passe inévitablement par la publication.

- Tester les différentes procédures et évènements en cours de développement.
- Faire la trace des variables à l'aide d'instructions d'affichage
- Tester les saisies de l'utilisateur à l'aide de conditions «limites»
- Faire tester le programme par d'autres personnes dans différents environnements

Étape 6 *(Faire la documentation du projet)*

- Faire un fichier *lisezmoi.txt* qui contient :
  - Votre nom et la date de début et de fin du projet;
  - Le nom du projet et une brève description du programme (son but);
  - La liste des fichiers / répertoires importants du projet;
  - Toute autre information pertinente (nom d'utilisateur mot de passe, version, etc.)
- ***(optionnel)*** *À l'aide de saisies d'écrans des interfaces dans l'application, faire un guide d'utilisation dans un document HTML, PDF ou autre format reconnu.*

Étape 7 *(Assurer la distribution du projet)*

- Créer un fichier compressé avec l'ensemble des répertoires et des fichiers de votre projet.
- Liste des éléments que l'on doit retrouver dans le dossier de l'application:
  - Le fichier principal du projet **index.php**
  - Tous les autres fichiers de scripts **(*.php)**
  - Une extraction **COMPLÈTE** des données SQL dans un fichier **(.sql)**
  - Un répertoire d'images avec les images pour l'application
  - Le fichier d'information sur l'application **lisezmoi.txt**
  - Tous les autres fichiers nécessaires au bon fonctionnement du projet (Images, exécutables, fichiers de données, etc.)

N.B. Pour éviter les conflits entre les projets et permettre le fonctionnement de tous les projets sur le même serveur d'applications, le nom de la base de données doit être le même que votre nom d'utilisateur Centre ***Exemple: duche123***

- Déposer votre projet (fichier zip) sur le site du cours à l'endroit prévu.
- Publier et tester votre application finale sur le serveur de publication à **http://pub.lan/prog/xxxxxxxx.** Utilisez le sous-répertoire que vous avez déjà créé lors de l'activité de publication. *(nom d'usager: **prog**, mot de passe : **prog**)*

- À des fins de correction, seulement le dernier fichier archive déposé sur le site du cours sera utilisé. (ZIP ou RAR)

N.B. La publication finale du projet doit être complétée à la 115^{ième} heure du module. Informez l'enseignant du plateau afin qu'il puisse prendre connaissance de votre dépôt.

3.3 À la séance d'examen en salle d'évaluation

Lors de la séance d'examen de connaissance pratique (tâche 1) qui se tient en salle d'évaluation, vous aurez à **SIGNER** et **REMETTRE** à l'examinateur la version officielle du guide de projet. Ce guide de projet (tâche 2) vous sera remis lors de la séance d'examen.

ANNEXE A

Grille d'éléments de contenu

Les éléments de contenu suivants ainsi que les critères et les conditions sont nécessaires à la réalisation du projet.
TOUS LES ÉLÉMENTS MARQUÉS INCOMPLET DEVRONT ÊTRE CORRIGÉS ET LE PROJET REMIS AVANT LA DATE FINALE

Cochez si l'élément est présent	Élément de contenu	Référence élément de compétence	Bloc / Référence	Critères	Conditions ou exemple de contenu
	Fonctionnement de l'application	2.2	14	Aucune erreur d'exécution	L'application doit s'exécuter correctement et sans aucune erreur fatale d'exécution pour l'ajout, la modification, la suppression et l'affichage des demandes de service.
	Formulaires	4.1	5	Avoir au moins trois formulaires	Ex.: un formulaire de connexion, un formulaire d'ajout et de modification, un formulaire de configuration, etc
	Zone de textes	4.1	5-6	Contenir une ou plusieurs zones de texte et étiquettes	Contenir au moins zone de texte simple, et une zone de texte multi ligne
	Variables	4.1	5-6-13	Contenir au moins cinq variables	Contenir au minimum une variable globale et une variable locale.
	Opération date/heure	4.1	11	Contenir au moins une opération sur une variable date/heure	Ex.: time(), Date(...)
	Structures conditionnelles	4.1	7	Contenir deux formes de structures conditionnelles	Contenir deux formes de structures conditionnelles différentes
	Boucles de programme	4.1	12	Contenir au moins une boucle de programme	Instruction de bouclage au choix. (while, for, foreach...)
	Cases à cocher et boutons options	4.1	5	Contenir un ou l'autre	Au moins une case à cocher ou un bouton option `<INPUT TYPE="CHECKBOX"…<INPUT TYPE="RADIO"…`
	Boutons de commandes	4.1	5	Contenir au moins un bouton de commande	Au moins un bouton
	Zone de liste	4.1	5	Contenir au moins une zone de liste de sélection.	Contenir au moins une zone de liste avec au minimum deux choix.
	Images	4.1	5 à 17	Contenir au moins cinq images.	Au moins cinq fichiers images différents.
	Table de données	4.1	17	Contenir une table ou un fichier pour enregistrer des données	Contenir une base de données avec au moins une table d'enregistrement pour les demandes ou un fichier de données
	Bouton pour le transfert e fichier (Facultatif)	4.1	18	Contenir au moins un bouton pour transférer un fichier sur le serveur.	Au moins un bouton pour le téléchargement de fichiers `<INPUT TYPE="FILE"…`
	Gestion des événements (Facultatif)	4.1	Télécom / web	Faire la gestion d'au moins un événement à l'aide de javascript	Ex.: onClick, onSubmit, onChange, onMouseOver, etc
	Accès à une base de données à l'aide des fonctions spécialisées	2.1	17	Accéder à une base de données en lecture ou en écriture sur au moins deux tables	Utilisation des fonctions particulières mysql_fecth_ array ou rows ou ODBC
	Opérations sur une chaîne de texte	2.1	6-7	Contenir au moins une opération sur une chaîne de texte	Concaténation, découpage, longueur de chaîne...
	Opérateurs logiques	2.1	6-7	Contenir au moins un opérateur logique	OR, || AND && , NOT !,
	Opérateurs mathématiques	2.1	6	Contenir au moins un opérateur mathématique	+, - , * , /, pow, round, …
	Fonctions prédéfinies et personnalisées	2.1	9	Contenir au moins deux fonctions personnalisées avec au moins un paramètre en entrée et/ou en sortie	Exemple.: `function validation($user,$pass){… return $param;`
	Commentaires	2.1	8	Contenir des commentaires d'entête généraux avec nom, prénom, date et description.	Commentaire sur au moins 80% des instructions et au début de chaque fonction avec des paramètres
	Distribution et empaquetage	2.1	16	Transmettre l'ensemble des fichiers de l'application (lisez-moi.txt inclus) (voir étape 7)	Utilisation d'un logiciel d'archivage (WinZip, WinRar…) ou d'un utilitaire de déploiement
	Contrôles avancés (Facultatif)	2.1	18 à 22	Contenir au moins un objet ou un élément de programmation avancé	Utilisation d'au moins une classe, une fonction ou une technique de programmation contenue dans l'un ou l'autre des blocs de formation 18 à 22.

Notes sur le projet :

DATE DE REMISE (JJ/MM/AAAA) _____/_____/_____