Lancement fil rouge => réa d’un projet finalisé, avancé avec les fonctionnalités apprises dans l’année : REA 1, 2...

Mémoire + PPT + Projet : enchaînement de projet => se construisent de la même manière

(CDC  : contexte, contraintes, planif, description client, dictionnaire des données)

Numéroter les pages

**INTRO** : présentation du contexte, j’ai préparé telle application pour valider telles compétences. J’ai choisi de vous présenter mon stage/tel projet fil rouge.

***Abstract*** = même chose mais en anglais

Sommaire, pas obligatoire

1. **Analyse du besoin / Cahier des charges**

Faire juste les grands titres pour éviter de tout lire : prez et objectifs, cibles, besoins…

Étude de marché, référencement

Ensemble des données à collecter, charte graphique, …

Contexte

Présentation

1. **Analyse fonctionnelle**

                   => partie technique

Fonctions de l’appli, fonctionnalités

1. **Use Case** : cas d’utilisation complet //pas dans le cahier des charges// Exemples de maquettage

Représente les acteurs de mon système d’informations

Il va permettre au dev d’organiser les pages de son site web, car il décrit l’ensemble des fonctionnalités du site

un cas d’utilisation  = une fonctionnalité

Fonctionnalité utilisée par des acteurs différents (utilisateur,membre, modérateur)

Expliquer l’héritage ; s’il y a des include et extend, les expliquer également

Le *extend* donne la possibilité de.

1. **Diagramme d’activité** (facultatif) : 2 diagrammes

Expliquer l’état initial au niveau du serveur applicatif (en attente d’une action) et final

Déroulement d’une action : que va t-il se passer dans chacun de mes cas d’utilisation

C’est l’algo de la fonctionnalité

L’acteur envoie une demande qui doit être acceptée par le serveur

L’user envoie un accept signal

Flow final vs état final (qu’est- ce qui est attendu à l’état initial)

1. **Diagramme de séquence** 2 ou 1 (Alternatif + erreur)

Interaction entre les acteurs et le système d’information : chronologique

Flèche pleine : envoi

Flèche pointillée : retour ou réponse

Scénario nominal (tout se passe bien / vs scénario erreur ou alternative

Le cas d’erreur met fin au scénario nominal

Le cas alternatif, je re-propose une saisie à l’utilisateur

1. **Diagramme de classe** (facultatif)
2. **Maquettage**

liste déroulante : stockage d’une entité

1. **Conception d’un système d’information**

*C’est du stockage de données. Je fais une analyse pour les concevoir*

Je vais stocker la donnée de façon structurée => support physique (SGBDR)

1e étape : MCD (entité, associations) pour structures. Définir l'ensemble des objets à stocker, que l’on appelle entité. Composée d’attributs.

Je leur met une clé primaire pour identifier chacun de mes objets, de mes utilisateurs

2e étape : MLD (notion de table, de relations, d’associations) pour voir les relations entre les données, sur la base de clés étrangères.

Donner un exemple

Cardinalité : nombre minimum de fois que l’entité participe à l’association et le nombre max

ex : 1 droit est détenu par un ou plusieurs

ex : le menu inclut 0 ou plusieurs commandes

Ex : un type 1 est absorbée et devient clé étrangère

Un type n = table d’association. L’association devient une table à part entière dans le MLD. clé primaire composée de 2 clés étrangères, absorbée des tables voisines.

Parler de create et de alter table

1. **MCD en entier**

Pour l’oral, ne mettre que quelques tables

et expliquer les cardinalités de type 1 et type N

Si on a 1,N : le 1 impose la saisie des 2 côtés, sur les 2 tables

mais c’est dangereux.

On peut mettre 1,n d’un côté et 0,n de l’autre côté

1. **MLD en entier** : démontrer qu’on a compris les associations de type 1 et n.

Expliquer l'absorption de la clé primaire par la table voisine

1. **SQL**

 “structure” - écrire les requêtes. Écrire les *create* avec table de type 1 et une autre de type n, les *alter*, les *drop*, une de chaque. *select*, *create*, *delete*, *update*, (SCRUD)

Expliquer si on a dû faire des changements dans le MLD, car il est plus simple de coder de telle ou telle façon

**IV. Arborescence**

Ne pas dire que ce n’est pas fini. Garder cette info pour la conclusion

**V. Fonctionnalités : code**

Expliquer des bouts de code

Ne mettre que des choses pertinentes. Mettre les fonctionnalités de mes diagrammes d’activité

**VI. Conclusion**

Valide le mémoire

Le Power Point suivra le même plan. Une slide pour chaque partie

Expliquer chacun des diagrammes en détail, décision, merge, le code….

Réalisation 1 : Git mais évaluée à la fin de formation

Réalisation 2 : HTML / CSS

Réalisation 3 : UML

Réalisation 4 : Conception - MCD, MLD, SQL

Réalisation 5 : Langage (variable, conditions, boucles ....)

Réalisation 5 bis : Responsivité, graphisme, bootstrap

Réalisation 6 : Back end

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_