

CARNET DE BORD

	DATE	Effectué par	TRAVAIL EFFECTUE	TEMPS	
	19/09	Thomas	Analyse du sujet, création du diagramme de classe sur MOCODO	1h30	
		Lenny	Analyse du sujet, création du diagramme de classe sur MOCODO	1h30	
		Luis	Analyse du sujet, prise de note et réflexion sur les contraintes du sujet à implémenter.	1h30	
	21/09	Thomas	Installation des tables / début de la création des classes en JAVA (PERSONNE, CLIENT, MONITEUR, COURS, PONEY, RESERVATION)	3h	
		Luis	Création d'un jeu de données d'insertion conséquents afin de pouvoir tester notre application et nos futur triggers. Insertion des tables : CLIENT/MONITEUR/COURS/RESERVER	1h30	
		Lenny	Insertion des tables PERSONNE et PONEYS + Test de la table RESERVER	2h	
	26/09	Luis	Création triggers verifPoids/VerifHeureRepos/ VerifHeureReservation avec Lenny	3h30	
		Lenny	Trigger ajoutePersonneCollectif / VerifHeureRepos / VerifHeureReservation / VerifHeureMaxCours avec Luis	4h	
		Thomas	Mise en place de la classe ConnexionBD + Requete	1h30	
	03/10	Lenny	Trigger VerifPayement + Fonction JDBC chargerPoney / chargerCours / chargerMoniteur	2h30	
		Thomas	Création de différente requetes, maxIdPersonne,maxIdCours,AffichelesReservations	1h30	
		Luis	trigger ajoutPersonneHoraire / verifPoneyEnCours / verifPersonneReserveDansClient + réflexions sur la qualité de notre Base de données (norme 3FN, redondance de données, questionnement sur l'heritage CLIENT/MONITEUR)	2h30	
	10/10	Lenny	JDBC fonction insererClient / insererMoniteur/ insererPoney / insererReservation / maxIdCours / maxIdPoney / maxIdPersonne	1h30	
		Luis	Création du triggers verifPoids, verifHeureReservation, verifHeuresMaxCours + Test de tout les triggers, afin de vérifier leur bon fonctionnement + création d'un jeu de test d'insertion afin de vérifier que les contraintes (triggers et DF) ont bien été respectées.	2h30	
		Thomas	Commencement de l'interface, Initialisation des différents Menu: Affichage des données, Insertion des données Liaison des affichages avec les différents requetes	1h30	
	17/10	Lenny	Création des tables ANCIENS + Ajout des triggers ajouteTableAncienPersonne / ajouteTableAncienClient / ajouteTableAncienMoniteur + Ajout des triggers before update	3h	
		Luis	Création des maquettes de l'applications : Maquette Accueil Grand Galop Maquette Enfant Grand Galop Maquette Adulte Grand Galop Maquette Compétition Grand Galop Maquette Reservation Grand Galop Maquettes page de Connexion	2h30	
		Thomas	Succes : Affichage possible de toutes les classes : une ou toutes les Personnes , un Client ou les Clients, un Moniteur ou les Moniteurs, une Reservation ou les Reservations, un Cours ou Les Cours, un Poney ou les Poneys - Inserstion possible de toutes les données (New : Personne,Client,Moniteur,Poney, Cours,Reservation)	1h30 + 4h 20/10	
		Lenny	Création de la template gerer_cours.html / fonction deleteCours / getMaxIdCours / ajouteCours / route Cours, AddCours + Ajout des docString du fichier ConnexionSQLAlchemy / Requete.java et views	3h + 3h 20/11	
		Luis	Création du Dossier de Projet avec le RENDU PDF contenant les contraintes, les différentes explications de notre modélisation (MCD, MLD, DF) et autres explications annexes. Renommer tout les fichiers de l'applications (creaPon.sql, insPon.sql et ExePonney.java), modifier l'arborescence des dossiers et effectuer les dernières vérifications / modifications avant le rendu	5h	

	15/11	Thomas	<p>Initialisation d'un nouveau menu : Supprimer les données Liaisons des requetes de suppression avec l'affichages Success : Supprimer toute les données possible : Supprimer une Personne Supprimer un Client Supprimer un Moniteur Supprimer un Cours Supprimer un Poney Supprimer une Reservation</p> <p>Initialisation des fichiers pour le démarrage de flask Création des fichiers : ORM SQLAlchemy Client.py Personne.py Moniteur.py Poney.py Cours.py Reserver.py Ajout des templates : gerer_client.html ,login.html, index.html Création des fonctions de récupération d'information : client / moniteur / cours / poney / reservation Création des routes d'accès aux différents templates/data Création des DataTable Création du bouton supprimer + requete ajax suppression Création du bouton ajouter + modal avec Kendo UI / Bootstrap + requete ajax ajout</p>	3h + 4h 15/10 soir + 4h 16/11	
	17/11	Thomas	<p>Ajust des templates gerer_poney / gerer_reservation Création des routes d'accès aux template + d'accès aux data + passerelle d'ajout et suppression reservation poney Mise en place d'un systeme de sécurité de connexion avec flask_login</p>	4h	
	22/11	Lenny		1h30	
		Luis		1h30	
		Thomas		1h30	
	29/11	Lenny		1h30	
		Luis		1h30	
		Thomas		1h30	
	06/12	Lenny		1h30	
		Luis		1h30	
		Thomas		1h30	
	13/12	Lenny		1h30	
		Luis		1h30	
		Thomas		1h30	
	10/01	Lenny		1h30	
		Luis		1h30	
		Thomas		1h30	
	11/01	Lenny		1h30	
		Luis		1h30	
		Thomas		1h30	
	17/01	Lenny		1h30	
		Luis		1h30	
		Thomas		1h30	

	18/01	Lenny		1h30	
		Luis		1h30	
		Thomas		1h30	