**SOMMAIRE**

1. **PLANIFICATION DU PROJET**
   1. Planification des tâches ………………………………………………………………………………………………
   2. Répartition des tâches ………………………………………………………………………………………………
   3. Diagramme de Gantt ………………………………………………………………………………………………
   4. Phase de test du projet ………………………………………………………………………………………………
2. **BASE DE DONNEES** 
   1. Définir la base de données ……………………………………………………………………………………………….
      1. Diagramme entité association ………………………………………………………………………………….
   2. Optimisation de la base de données ………………………………………………………………………………….
   3. Accessibilité de l’administrateur ………………………………………………………………………………….
   4. Liaison avec le site internet ……………………………………………………………………………………………….
   5. Liaison avec le site web (autonome) …………………………………………………………………………………..
3. **PROGRAMMATION** 
   1. Réaliser le programme principal ………………………………………………………………………………….
   2. Définir les fonctions ……………………………………………………………………………………………………………
      1. Fonction « PrisePhoto » ……………………………………………………………………………………………….
      2. Fonction « Serveur/Client » ………………………………………………………………………………….
      3. Fonction « AllumerLED »
      4. Mise a jour de la base de données …………………………………………………………………………….
      5. Phase de test ……………………………………………………………………………………………………………
4. **SITE WEB**
   1. Réalisation de la page de connexion ………………………………………………………………………………….
      1. Design et mise en page ……………………………………………………………………………………………….
      2. Fonctionnement des utilisateurs authentifié ………………………………………………………….
      3. Liaison avec la base de données ………………………………………………………………………………….
   2. Réalisation de la page d‘accueil ………………………………………………………………………………….
      1. Affichage des photos ……………………………………………………………………………………………….
      2. Barre des tâches ……………………………………………………………………………………………………………
      3. Design de la page ……………………………………………………………………………………………….
      4. Fonctionnalités des administrateurs ………………………………………………………………………
   3. Relation avec la base de données ………………………………………………………………………………….
      1. Mise à jour des mots de passe ………………………………………………………………………………….
   4. Mise en service du Raspberry ……………………………………………………………………………………………….
      1. Installation apache2, PHP ………………………………………………………………………………….
      2. Installation phpMyAdmin, Maria DB ………………………………………………………………………
      3. Mise en lien de la base de données et du site web ……………………………………………
5. **PLANIFICATION DU PROJET** 
   1. **Planification des tâches**

Nous avons tout d’abord listé toutes les tâches que nous avons pu déjà prévisualiser.

Une image contenant table

Description générée automatiquement

* 1. **Répartition des tâches**

La répartition des tâches est très importante pour le bon déroulement des tâches. Nous avons donc décider de nous attribuer des rôles selon les parties ou bien même les tâches. Concepteur, Réalisateur et Testeur, seront les rôles répartis sur le groupe afin d’avoir une cohérence entre le temps de travail répartie de chacun.

Concepteur : Personne qui va gérer la tâche et aider le Réalisateur.

Réalisateur : C’est la personne qui va programmer, designer et régler les problèmes.

Testeur : Il s’agit de ce qui vont tester les programmes, le site internet, et vont le reporter au Réalisateur.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Planification du projet | Benjamin | Matthieu | Clara |
| 1.1 | Rapport | R | R | R |
| ↳ 1.1.1 | Plan | R | C | C |
| 1.3 | Planification des tâches | T | C | R |
| 1.4 | Répartition des tâches | R | R | R |
| 1.4 | Diagramme de Gantt | R | C | C |
| 1.5 | Phase de test du projet | T | T | T |
| 2 | Base de données |  |  |  |
| 2.1 | Définir la base de données | C | T | R |
| ↳ 2.1.1 | Diagramme entité association | R | T | C |
| 2.2 | Optimisation de la base de données | R | T | C |
| 2.3 | Exécution de la base de données | C | T | R |
| 2.4 | Accessibilité de l’administrateur | C | T | R |
| 2.5 | Liaison avec le site web (autonome) | R | T | C / R |
| 2.6 | Fonctions pour simplifier le code | C / R | T | T |
| 3 | Programmation |  |  |  |
| 3.1 | Réaliser le programme principal | C | C / R | C |
| 3.2 | Définir les fonctions | C | C / R | C |
| ↳ 3.2.1 | Fonction « PrisePhoto » | C | C / R | C |
| ↳ 3.2.2 | Fonction Serveur/Client | T | C / R | T |
| ↳ 3.2.3 | Fonction « AllumerLED » (non reçues) | - | C / R | - |
| ↳ 3.2.4 | Fonction « MajMdp » | C | C / R | C |
| 3.3 | Phase de test | T | T | T |
| 4 | Site Web |  |  |  |
| 4.1 | Réalisation de la page de connexion | C | C | R / T |
| ↳ 4.1.1 | Design et mise en page | C | C | R |
| ↳ 4.1.2 | Fonctionnement des utilisateurs authentifié | T | T | C / R |
| 4.2 | Réalisation de la page d‘accueil | R | C | R / T |
| ↳ 4.2.1 | Affichage des photo | R | T | C / T |
| ↳ 4.2.2 | Barre des taches (en haut) | T | C | R |
| ↳ 4.2.3 | Design de la page | C | C | R |
| ↳ 4.2.4 | Fonctionnalités des administrateurs | C | T | R / T |
| 4.3 | Mise à jour des mots de passes | C | T | R / T |
| 4.4 | Création d'un nouvel utilisateur | C | T | R / T |
| 4.5 | Requête pour l'affichage des images | R | T | C |
| 4.6 | Mise en service du Raspberry | C | R | T |
| ↳ 4.6.1 | Installation apache2, PHP | C | R | T |
| ↳ 4.6.2 | Installation phpMyAdmin, Maria DB | C | R | T |
| ↳ 4.6.3 | Mise en lien base de données et site web | C | R | T |

* 1. **Diagramme de Gantt**

Une fois la répartition des tâches faites, il nous suffit d’estimer le temps passé sur les tâches ainsi que leurs dépendances pour avoir une idée de la « Deadline » du projet. Voici le Gantt que nous avons pu réaliser.

*« Diagramme de Gantt* ***FINAL*** *»*

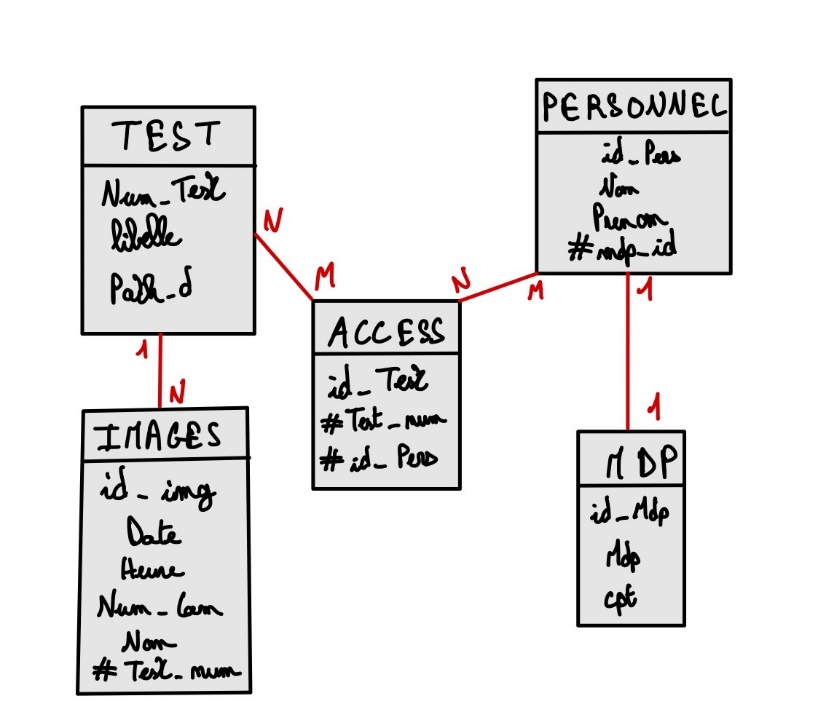
1. **BASE DE DONNEES** 
   1. **Définir la base de données**

Pour définir la base de données nous avons commencé par définir toutes les informations utiles pour gérer entièrement les données des utilisateurs, des tests et des images. Nous avons alors listé toutes ces données.

Tests : Nom du test, ainsi que son chemin (path).

Personnels : Nom, prénom, mot de passe.

Images : Nom, date, heure, numéro de caméra.

* + 1. **Diagramme entité association**

Durant notre première année de formation nous avons appris à réaliser un diagramme nous permettant d’initialiser une base de données, ce diagramme se nomme « Entité / Association ».

Voici notre diagramme Entité / Association finalisé. Cela représente les tables qui seront dans la base de données avec les données de chacune des tables, ainsi que leurs relations.

**Les relations** sont bien évidement les traits reliés entre les tables. Ils permettent une certaine liaison entre les tables pour avoir une base de données dynamique et complète.

Le nombre « **1** », « **N** » et « **M** » sont transcris au part et d'autre des relations. Voici des exemples pour avoir une compréhension plus facile de ces significations.

Exemples : • De la table Images à Test : Il y a un nombre **N** d’images pour **1** test.

• De la table Test et Access : Il y a **N** tests avec **M** accès pour accéder aux tests.

• De la table Access à Personnel : Il y a **N** accès pour plusieurs (**M**) employés.

• De la table Personnel à Mdp : Il y a **1** employés qui a **1** mot de passe.

De nombreux mot-dièse « # » sont devant certains champ des tables. C’est une manière de décrire que le champ est une clé étrangère. C’est-à-dire qu’elle dépend de la clé primaire de la table voisine. Cela permet une certaine relation entre les tables.

* 1. **Optimisation de la base de données**

Après plusieurs utilisations de la base de données, des tests ont été fait pour savoir si la base de données était opérationnelle. On s’est aperçus que certaines relations ne server à rien. Nous avons donc refait l’initialisation de la base de données. *Voir directement le code sql pour la création de la base de données.*

Le principe était de juste faire 3 tables (Personnel, images et logs). Ces tables sont le strict minimum à avoir selon nous. La table « Personnel » aura toutes les informations utiles pour le site web et les programmes python pour fonctionner les cheminements de nos tâches.

* 1. **Exécution de la base données (Test)**

***Le code sql finale se trouve dans le fichier B.B.B.B***

• Création de la base de données sur PhpMyAdmin (PMA) :

PhpMyAdmin est une application web de gestion de base de données MySQL et MariaDB, réalisée principalement en PHP.

Nous avons alors accédé à PMA grâce à isis pour le moment, nous devrions tout simplement par la suite installer PMA sur le Raspberry et créer un serveur pour que cela fonctionne de la même manière qu’avec isis.

Pour la création de la base de données tout se passe depuis l’interface graphique de PMA qui est très intuitif. En étant sur isis nous n’avons la possibilité d’avoir qu’une base de données. C’est pour cela que nous pouvons directement commencer à créer les tables grâce à ces commandes :

Table PERSONNEL :

CREATE TABLE PERSONNEL (

idUser INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

nom VARCHAR(70) NOT NULL,

prenom VARCHAR(50) NOT NULL,

login VARCHAR(30) NOT NULL,

mdp VARCHAR(30) NOT NULL,

admin BOOLEAN,

cpt\_renouvellement INT NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=10;

Table IMAGES :

CREATE TABLE IMAGES (

idImg INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

nomImg VARCHAR(70) NOT NULL,

nomTest VARCHAR(70) NOT NULL,

dateImg DATE NOT NULL,

heure TIME NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=10;

Table LOGS :

CREATE TABLE LOGS (

idLog INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

nomLog VARCHAR(70) NOT NULL,

dateLog DATE NOT NULL,

heureLog TIME NOT NULL,

numUser INT,

numImg INT,

FOREIGN KEY(numUser) REFERENCES PERSONNEL(idUser),

FOREIGN KEY(numImg) REFERENCES IMAGES(idImg)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=10;

Les tests pour la création de la base de données ainsi que les tables sont concluants, ce sont des codes qui sont efficace. Quelque difficulté pour la création des clés étrangères mais très vite réglés.

• Ajouter des données dans les tables :

Si les tables sont bien initialisées alors l’ajout d’enregistrements dans les tables devrais bien ce passer. Pour les enregistrements il préférable de ne pas mettre des espaces pour éviter toute confusion dans le code, nous en reparlerons plus tard dans la partie du site web. Voici les commandes pour ajouter des exemples d’enregistrement :

Ajouts des enregistrements :

INSERT INTO PERSONNEL (nom, prenom, login, mdp, admin, cpt\_renouvellement) VALUES

('faretie','benjamin','fb','fb',TRUE,5),

('perre','matthieu','pm','pm',FALSE,30),

('woodman','clara','wc','wc',FALSE,50);

INSERT INTO IMAGES (nomImg, nomTest, dateImg, heure) VALUES

('Photo1','Test1','2022-04-27','19:08:48'),

('Photo2','Test1','2022-05-27','20:08:48'),

('Photo1','Test2','2022-06-27','21:08:48');

INSERT INTO LOGS (nomLog, dateLog, heureLog) VALUES

('Delete Photo 1','2022-07-27','20:00:00'),

('Modification du nom de la Photo 2','2022-08-27','20:00:00'),

('Mot de passe modifier du User 1','2022-09-27','20:00:00');

L’ajout d’enregistrement n’est pas compliqué en therme de code mais il faut être judicieux au niveau des contraintes d’enregistrement. Nous avons essayer de manipuler ces enregistrements avec au début des « nomImg » avec des espaces comme « Image 21 » mais l’espace pause problème dans le code php. Nous en tirons qu’il ne faut pas mettre d’espaces dans les champs utiliser dans le code php. Pour supprimer un enregistrement il faut tout simplement utiliser la commande :

DELETE FROM <nom\_table> WHERE <nom\_champ> = <valeur\_champ>

* 1. **Accessibilité de l’administration**

L’accès à la base de données est seulement pour les administrateurs. C’est-à-dire qu’a la création du serveur phpmyadmin sur le rasberry il faut configurer l’identifiant et le mot de passe de l’accès à la base de données. Il sera lors nécessaire de mémoriser les informations de connexions car il y a le risque que tout le monde n’ai plus accès au site à cause d’oublie de mot de passe, alors l’accès à la base de données est plus que rudimentaire.

* 1. **Liaison avec le site Web (autonome)**

Le site web étant codé en html/css et php nous devions se connecter depuis le code php à la base de données. La commande nécessaire à la connexion :

try {

$bd = new PDO("mysql:host=localhost;dbname=","id","mdp");

$bd->exec('SET NAMES utf8');

}catch (Exception $e){

die("Erreur: Connexion à la base impossible");}

Ce petit bout de code permet d’utiliser la base de données prédéfinie dans les options de la fonction PDO(), et si l’accès n’est pas fonctionnel cela retourne une erreur indiquant que la connexion à la base de données n‘ est pas accessible.

Pour conclure la création de la base de données n’est pas très compliquée et n’a pas réellement beaucoup de risques mis à par la perte de mot de passe admin, mais il y a une solution à cela. Il suffirait que l’admin se réfère à l’interface graphique de Phpmyadmin pour ensuite pouvoir retrouver son mot de passe ainsi le modifier. Mais nous en reparlerons dans la partie du site web car le mot de passe sera crypté grâce à la fonction md5() en php. Une fois crypté le mot de passe n’est plus récupérable depuis la base données, alors nous définirons une méthodes pour ce risques dans la partie Web.

1. **PROGRAMMATION**
   1. **Réaliser le programme principal**

***Toutes les fonctions présentées après sont accessibles dans l’archive nommées A.A.A.A***

***(Dossier Raspberry)***

Ce programme permet au Raspberry de prendre des photos et d’agir comme un serveur de requête. Il respecte donc la condition d’être disponible à n’importe quel moment de la journée, pour cela on automatise le démarrage du fichier. On utilise donc la ressource « Crontab » qui dans notre cas démarre le script au démarrage (Image A) du Raspberry. Pour permettre ensuite de garder le programme ouvert en permanence, on utilise une boucle sans fin (« While True »).

Une image contenant texte

Description générée automatiquement

Image A

* 1. **Définir les fonctions**

Plusieurs fonctions sont nécessaires au bon fonctionnement, la 1ere est donc celle qui permet de prendre des photos, la 2nde permet de démarrer un serveur exploitant la fonction de prise de photo, la 3e, elle, teste et reprend une photo en cas de mauvaise luminosité et la dernière fonction nous autorise à venir communiquer avec le serveur.

* + 1. **Fonction « PrisePhoto »**

Cette fonction est assez simple à mettre en place, il nous suffit de récupérer 2 variables, le chemin et le nom de la photo puis d’effectuer une commande dans le terminal.

Pour cela, on utilise le module « FS webcam » qui permet de prendre des photos grâce à des lignes de commandes. Nous devons également pouvoir prendre des photos avec un nombre de caméras variable, ici on suppose que le maximum de caméra connecté est de 4 (4 ports USB sur le Raspberry). Nous avons également besoin de certaines librairies Python : « os », « subprocess » et « datetime »

*Voie d’amélioration : Créer une boucle pour permettre un nombre illimité de caméras connectés et permettre une meilleure lisibilité du code.*

* + 1. **Fonctions « Serveur/Client »**

Après avoir permis l’utilisation de la caméra depuis python, maintenant il nous faut la possibilité de le faire depuis une autre machine. C’est alors que cette fonction prend tout son sens, celle-ci permet donc de communiquer entre un client et le Raspberry. Pour cela, il nous suffit de créer un socket et de récupérer les requêtes émises par le client

Cette fonction est donc double, une coté machine serveur et l’autre du côté du client.

Une image contenant table

Description générée automatiquement

* + 1. **Fonction « AllumerLED »**

Cette fonction comme son nom l’indique permet l’allumage de la LED et donc la reprise des photos dans des conditions « meilleures ».

Elle se décompose en 2 principales actions :

* Celle de tester le taux de pixel noir présent sur l’image (50% de pixel noir sur une image de dimensions 1920\*1080px).
* Celle d’allumer la LED et de recapturer les images.

A cause de matériel non fourni, elle n’a pas pu être mise en place. Elle permet quand même de tester le niveau de noir d’une image (non raccordé au programme principal).

* + 1. **Mise à jour de la base de données**

Maintenant que nous avons pris en photo et stockée sur le Raspberry l’image, il nous faut donc mettre en lien ces images et projets avec le site. On a donc besoin de 2 fonctions :

* La mise à jour dynamique des mots de passe
* L’ajout automatique des images dans la base de données

La 1ere fonction consiste à envoyer une requête et modifier tout les champs de « cpt\_renouv » (compteur de 90 jours qui permet de vérifier si le mot de passe et encore valable) en enlevant 1 a ces enregistrements par l’appel journalier d’un fichier python via Crontab.

La 2nde fonction, elle, permet d’enregistrer les nouvelles images dans la base de données en envoyant une requête contenant le nom du projet et le chemin absolu de l’image celle-ci est appelé a la suite de la prise de photo.

* 1. **Phase de test**

De nombreux tests ont été prévus :

* Utilisation normale (Dossier classique, 1 caméra et pas de problème de luminosité
* Nombre de caméras variable (2 caméras connectés)
* Reprise de photos (Problème de luminosité)

Utilisation Normale :

La prise de photo enregistre bien à l’endroit convenu et l’image s’affiche correctement

Envoie de la requête :

Réception de la requête :

Enregistrement de la photo :

Nombre de caméras variable :

On peut distinguer la prise de photo de la caméra 1 et de la caméra 2 car on ajoute le nom de la caméra qui a été utilisé pour la prise de photo.

Reprise de photos :

Impossible car matériel incomplet (cf. 3.3.3)

1. **SITE WEB**
   1. **Réalisation de la page de connexion**

Le site web à forcément une page de connexion pour avoir une certaines restrictions selon l’utilisateur qui utilise l’interface du site. Alors la page de connexion est essentielle à notre site.

* + 1. **Design et mise en page**

Pour le design nous avons opter sur un fond d’une couleurs similaire au logo Thales pour avoir une uniformisation des couleurs. Ainsi au centre et en premier plan le bloque de connexion où il faut renseigner l’identifiant et le mot de passe. En cas d’oublie du mot de passe il faudra systématiquement envoyés un mail à l’administrateur. En cas de mauvais mot de passe ou d’identifiant alors un message d’erreur le signalera…

* + 1. **Fonctionnement des utilisateurs authentifié**
    2. **Liaison avec la base de données**
  1. **Réalisation de la page d‘accueil**
     1. **Affichage des photos**
     2. **Barre des taches**
     3. **Design de la page**
     4. **Fonctionnalités des administrateurs**
  2. **Relation avec la base de données**
     1. **Mise à jour des mots de passe**
     2. **Création d'un nouvel utilisateur**
     3. **Requête pour l'affichage des images**
  3. **Mise en service du Raspberry**

Maintenant que tout fonctionne, il faut mettre en place le site sur le Raspberry, Nous avons donc besoin de diffèrent paquets pour gérer la partie Web :

* Apache2 (Service Web),
* PHP (Non installé de base dans le paquet Apache2)

et ensuite des paquets de base de données :

* PhpMyAdmin (gestionnaire de base de données avec interface graphique)
* Maria DB (Base de données)
  + 1. **Installation apache2, PHP**

**Utile ?**

* + 1. **Installation phpMyAdmin, Maria DB**

**Utile ?**

* + 1. **Mise en lien base de données et site web**