

Rapport S2 Projet Thales

Développement d'un système embarqué de prise de photos

Réalisé par:

- Dorian Maissonnier
- Mattéo Recoquillon
- Thibault Rommes
- Océan Colin-Rullière

IUT de Nice – BUT Réseaux & Télécommunications
Année 2024–2025

Sommaire détaillé du Rapport Thales

1. Présentation et Objectif du Projet

1.1. Contexte

1.2. Objectifs généraux

1.3. Technologies utilisées

1.4. Présentation de l'équipe

2. Exigences Fonctionnelles et Techniques

2.1. Exigences S1

2.2. Exigences S2

2.3. Cahier des charges

2.4. Contraintes techniques

3. Schéma Électrique du Montage

3.1. Objectif du montage

3.2. Liste des composants

3.3. Fonctionnement du circuit

3.4. Consignes de sécurité

4. Schéma de la Base de Données

4.1. Structure des tables

4.2. Relations entre entités

4.3. Spécificités S1 vs S2

5. Description de l'Installation et Configuration (Raspberry Pi 3 + PICO WH)

5.1. Installation de l'OS

5.2. Configuration réseau locale

5.3. Prise en main de Thonny et MicroPython

5.4. Tests de fonctionnement des composants

6. Configuration IP – Réseau

6.1. Affectation d'une IP statique

6.2. Ouverture des ports NAT pour SSH

6.3. Communication UDP entre Pi et Pico

7. Plan de Validation, Procédures de Test et Fiches d'Anomalies

7.1. Méthodologie de validation

7.2. Jeux de tests utilisés

7.3. Gestion des anomalies

8. Échanges Sécurisés entre les Pages du Site Web

8.1. Authentification par sessions sécurisées

8.2. Gestion des rôles (ACL)

8.3. Protection contre les accès directs

8.4. Déconnexion et journalisation

8.5. HTTPS et chiffrement

8.6. Sécurité des données transmises

8.7. Historique et logs utilisateurs

8.8. Redirections contrôlées

8.9. Accès aux fichiers protégés

9. Gestion GitHub

9.1. Organisation du dépôt

9.2. Historique des commits

9.3. Répartition des tâches dans Git

10. Gestion de Projet

10.1. Répartition des livrables entre S1 et S2

10.2. Calendrier prévisionnel

10.3. Méthodologie adoptée

11. Retour d'Expérience Individuel

11.1. Dorian Maissonnier

11.2. Mattéo Recoquillon

11.3. Océan Colin-Rulière

11.4. Thibault Rommes

12. Répartition des rôles selon le modèle RACI

12.1. Présentation du modèle RACI

12.2. Rôles détaillés de chaque membre

1. Présentation et Objectif du Projet

Le projet que nous avons mené dans le cadre des SAE 23/24 a pour objectif de développer une solution embarquée et intelligente de prise de photo automatisée, intégrant des technologies web, des microcontrôleurs, de la gestion de base de données et des composants électroniques. L'ensemble du dispositif repose sur une architecture à double niveau : un Raspberry Pi 3 en tant que serveur central, et un Raspberry Pi Pico WH agissant comme module capteur et d'activation d'éclairage.

Ce projet vise à mettre en œuvre un système capable de prendre des photos manuellement ou automatiquement, selon des conditions spécifiques comme l'absence d'activité durant 24 heures. Les photos sont ensuite archivées et organisées dans une base de données robuste, consultables depuis un site web sécurisé, hébergé en local sur le Raspberry Pi. Le site intègre une authentification multi-niveaux (utilisateur, administrateur, superadmin) et des fonctionnalités telles que les filtres, l'archivage, le téléchargement ou encore la suppression des fichiers multimédia.

Le Raspberry Pi Pico WH est utilisé pour la lecture d'un capteur de luminosité. En fonction de la lumière ambiante, il commande l'activation d'une LED afin d'assurer une exposition suffisante lors de la prise de vue. Il communique en Wi-Fi avec le Raspberry Pi principal et répond aux requêtes HTTP envoyées depuis le site web. Le système assure également un retour réseau régulier de l'état du Pico (IP, MAC, disponibilité).

1.1 Objectifs pédagogiques

- Mettre en œuvre une architecture réseau simple entre deux microcontrôleurs.
- Développer un site web complet en HTML/CSS/PHP avec gestion de rôles et sessions.
- Implémenter une base de données relationnelle SQLite avec persistance locale.
- Savoir interagir entre un site PHP et des scripts Python pour commander des périphériques externes.
- Apprendre à gérer un projet en équipe avec une répartition claire (RACI), planification et tests.

1.2 Objectifs techniques

- Permettre la prise de photo via une interface utilisateur web.
- Gérer automatiquement la prise d'une photo après 24h d'inactivité.
- Contrôler une LED en fonction de la luminosité détectée (en dessous d'un seuil).
- Assurer l'archivage, la gestion, le tri et l'affichage des photos prises.
- Créer un système de logs des actions (connexion, déconnexion, archivage, suppression...)
- Garantir la sécurité des pages via des contrôles d'accès et des sessions PHP.

1.3 Technologies utilisées

- Raspberry Pi 3 (serveur local, hébergement Apache2, gestion PHP, SQLite)
- Raspberry Pi Pico WH (capteur analogique, LED, Wi-Fi, HTTP server en MicroPython)
- HTML / CSS / PHP pour la création du site web interactif
- SQLite pour la base de données légère et embarquée
- Python pour les scripts de prise de photo et communication réseau
- SSH, NAT et configuration réseau pour l'accès et la configuration à distance

1.4 Résumé du fonctionnement général

Lorsqu'un utilisateur authentifié accède au site, il peut déclencher une photo, la consulter ou la supprimer selon ses droits. Le site gère les actions via des scripts PHP, qui appellent des scripts Python exécutés localement sur le Raspberry Pi. En parallèle, le Raspberry Pico analyse la luminosité ambiante et permet d'allumer la LED si nécessaire. L'ensemble est conçu pour être résistant, maintenable, accessible en local sans connexion internet, et capable d'évoluer vers d'autres cas d'usage (vidéosurveillance, détection d'événements, etc.).

2. Exigences

Les exigences fonctionnelles et techniques sont les suivantes :

S1

Systèmes :

- 1) Le système devra gérer la prise de photo manuelle.
- 2) Doit gérer la prise de photo automatique (au bout de 24h si aucune autre photo prise).
- 3) Doit archiver les photos prises dans la base de données (elle ne doit pas être supprimé automatiquement).

- 4) Création d'un site web permettant de parcourir l'historique de toutes les photos disponibles.
- 5) Le système devra gérer un système d'éclairage en cas de besoin (si éclairage insuffisant).
- 6) Le système devra gérer les utilisateurs autorisés (système d'authentification).
- 7) Les photos doivent être stockées de manière organisée par rapport à la date, l'heure, l'utilisateur, etc.
- 8) Doit avoir un mécanisme de gestion du stockage (rotation automatique, suppression des photos anciennes ou archivage).
- 9) Les photos doivent être prises et stockées rapidement, avec une latence minimale, (15 secondes maximum).
- 10) Le système doit pouvoir gérer un historique étendu (capacité d'archiver une quantité conséquente de photos).
- 11) Les photos doivent être stockées directement sur le serveur (Raspberry Pi).
- 12) Assurer un système robuste qui sauvegarde correctement les données même en cas de redémarrage du Raspberry.
- 13) Compresser les photos avant de les enregistrer pour économiser de l'espace si nécessaire.

Sur le site web :

- 14) Se connecter au site avec un nom d'utilisateur et mot de passe.
- 15) Nommer automatiquement les photos avec des noms pour chacune des photos.
- 16) Prévoir la possibilité de gérer beaucoup de photos sans compromettre les performances du serveur.
- 17) Savoir quand la photo a été prise (grâce à une date incrémentée au titre par exemple)

- 18) Savoir quel utilisateur a pris la photo ou si elle a été prise automatiquement.
- 19) Un seul utilisateur accède au site à la fois.
- 20) Pouvoir prendre une photo avec un bouton sur la page de connexion.
- 21) Pouvoir prendre une photo avec un bouton sur la page d'accueil du site.
- 22) Refuser l'accès de connexion sur le site si le mot de passe ou l'utilisateur est incorrect.
- 23) Restreindre l'accès au stockage de certaines photos définis aux utilisateurs (l'administrateur peut restreindre la vue de certaines photos).
- 24) Le site web doit être accessible sans connexion internet (Site web local).
- 25) Le système ne devra pas être surcharger en évitant le plus possible les photos chargées, plusieurs pages sur le site web inutiles, ...
- 26) Le site devra informer l'utilisateur du projet sur la page d'accueil.
- 27) Une base de données robuste pour contenir toutes les photos.
- 28) Un site capable de s'adapter selon le device.
- 29) Schémas d'une base de données qui fournit les informations de chaque table.
- 30) Une page de login pour se connecter soit en utilisateur soit en admin.
- 31) Le site doit inclure des filtres (par date, statut, etc.) et des options de téléchargement pour les groupes de photos.
- 32) Site Web fait de manière éco-responsable soit en ne pas surchargeant le site et en supprimant des photos régulièrement par exemple.
- 33) Mettre en place une galerie sur le site pour visualiser les photos stockées.
- 34) Un descriptif du projet Photo_ATB dans la page d'accueil sera sur le site.

- 35) Le site aura un menu pour changer de page entre l'accueil et la page photo.
- 36) Le menu restera le même pour les deux pages,
- 37) Une possibilité de déconnexion du site web à n'importe quel moment.
- 38) Il n'y aura pas de vidéo ou gif en fond d'écran dans le site (seulement des images).

En tant qu'utilisateur :

- 39) Le mot de passe doit être au moins de 8 caractères, avec au moins une majuscule, un chiffre et un caractère spéciaux.
- 40) Pouvoir modifier son mot de passe.
- 41) Demande de suppression de photos aux administrateurs.
- 42) Parcourir les photos enregistrées.
- 43) Accéder à la page d'accueil du site web.

En tant qu'administrateur :

- 44) Créer ou supprimer un compte utilisateur ou administrateur.
- 45) Attribuer des rôles ou des permissions spécifiques (par exemple : réalisateur, approuvateur, consulter, informer).
- 46) Réinitialiser les mots de passe des utilisateurs si nécessaire.
- 47) Désactiver ou réactiver des comptes.
- 48) Configurer les paramètres globaux du système, comme :
 - 49) Résolutions et qualités des photos (ne doivent pas être de piètre qualité).

50) Formats d'exportation par défaut (JPEG, PNG, etc.).

51) Stratégie d'archivage (durée de conservation, suppression automatique).

52) Accéder à des logs détaillés des actions utilisateurs et des événements système.

53) Débloquer le compte d'un utilisateur.

54) Modifier le nom du programme sur lequel le système est installé.

55) Supprimer définitivement une photo en passant sur le site ou autre.

S2

Cahier des Charges- Exigences Fonctionnelles et Techniques

2.1.1 Exigences Fonctionnelles

2.1.1.1 Gérer la prise de photo manuelle.

2.1.1.2 Gérer la prise de photo automatique si aucune photo n'est prise sous 24h.

2.1.1.3 Les photos doivent être stockées sur le serveur local (Raspberry Pi).

2.1.1.4 Compresser les photos avant enregistrement pour économiser de l'espace si nécessaire.

2.1.1.5 Nommer automatiquement chaque photo de manière unique.

2.1.1.6 Assurer une latence maximale de 15 secondes entre la capture et le stockage.

2.1.1.7 Gérer automatiquement un éclairage d'appoint en cas de luminosité insuffisante.

2.1.2 Gestion des Utilisateurs

2.1.2.1 Implémenter un système d'authentification par identifiant et mot de passe.

2.1.2.2 Un seul utilisateur peut être connecté à la fois.

2.1.2.3 Refuser l'accès si les identifiants sont incorrects.

2.1.2.4 Prévoir une page de connexion dédiée pour utilisateurs et administrateurs.

2.1.2.5 Le mot de passe doit contenir au minimum: 8 caractères, une majuscule, un

chiffre, un caractère spécial.

2.1.2.6 L'utilisateur doit pouvoir modifier son mot de passe.

2.1.3 Interface Web

2.1.3.1 Le site doit être accessible localement sans connexion Internet.

2.1.3.2 Intégrer un bouton de prise de photo sur la page de connexion.

2.1.3.3 Intégrer un bouton de prise de photo sur la page d'accueil.

2.1.3.4 Afficher une galerie organisée avec filtres (date, utilisateur, statut, etc.).

~~2.1.3.5 Offrir une option de téléchargement groupé des photos.~~

2.1.3.6 Intégrer un menu de navigation visible sur toutes les pages.

~~2.1.3.7 Afficher un descriptif du projet Photo_ATB sur la page d'accueil.~~

2.1.3.8 Le site doit être responsive (adaptatif à tout type d'appareil).

2.1.3.9 Ne pas utiliser de vidéos ou GIF en fond, seulement des images fixes.

2.1.3.10 Éviter les surcharges (pages inutiles, trop de photos chargées).

2.1.3.11 Intégrer une option de déconnexion accessible à tout moment.

2.1.4 Base de Données

2.1.4.1 Organiser les photos par date, heure, utilisateur et origine (manuelle/automatique).

2.1.4.2 Gérer un historique étendu (capacité importante d'archivage).

2.1.4.3 Assurer la persistance des données après redémarrage du Raspberry Pi.

2.1.4.4 Intégrer une stratégie de gestion du stockage (suppression ou archivage).

2.1.4.5 Fournir un schéma clair de la base de données (structure des tables).

2.2.1 Exigences Utilisateur

2.2.1.1 Modifier son mot de passe.

2.2.1.2 Parcourir les photos enregistrées.

2.1.3 Accéder à la page d'accueil du site web.

~~2.2.1.4 Demander la suppression d'une photo à l'administrateur.~~

2.2.1.5 Visualiser les métadonnées d'une photo (date, utilisateur, prise auto/manuelle).

2.2.2 Administrateur

2.2.2.1 Créer ou supprimer un compte (utilisateur ou admin).

2.2.2.2 Réinitialiser le mot de passe d'un utilisateur.

- 2.2.2.3 Activer ou désactiver un compte.
- 2.2.2.4 Attribuer des rôles et permissions (réalisateur, approbateur, etc.).
- 2.2.2.5 Modifier le nom du programme.
- 2.2.2.6 Débloquer un compte utilisateur.
- 2.2.2.7 Supprimer définitivement une photo.
- 2.2.2.8 Accéder aux logs détaillés (actions et événements).
- 2.2.2.9 Configurer les paramètres globaux :
 - 2.2.2.9.1 Résolution et qualité des photos.
 - ~~2.2.2.9.2 Format d'exportation par défaut (JPEG, PNG...).~~
 - 2.2.2.9.3 Politique d'archivage (durée, suppression automatique).

3. Schéma Électrique du Montage

Notice du Schéma Électrique

3.1. Titre du montage :

Connection Raspberry et Pico sur Breadboard de LED par capteur de lumière.

3.2. Objectif du montage :

Ce montage permet à la Raspberry qui contient le site web d'allumer la LED grâce au capteur de luminosité et de faire la liaison avec le Pico.

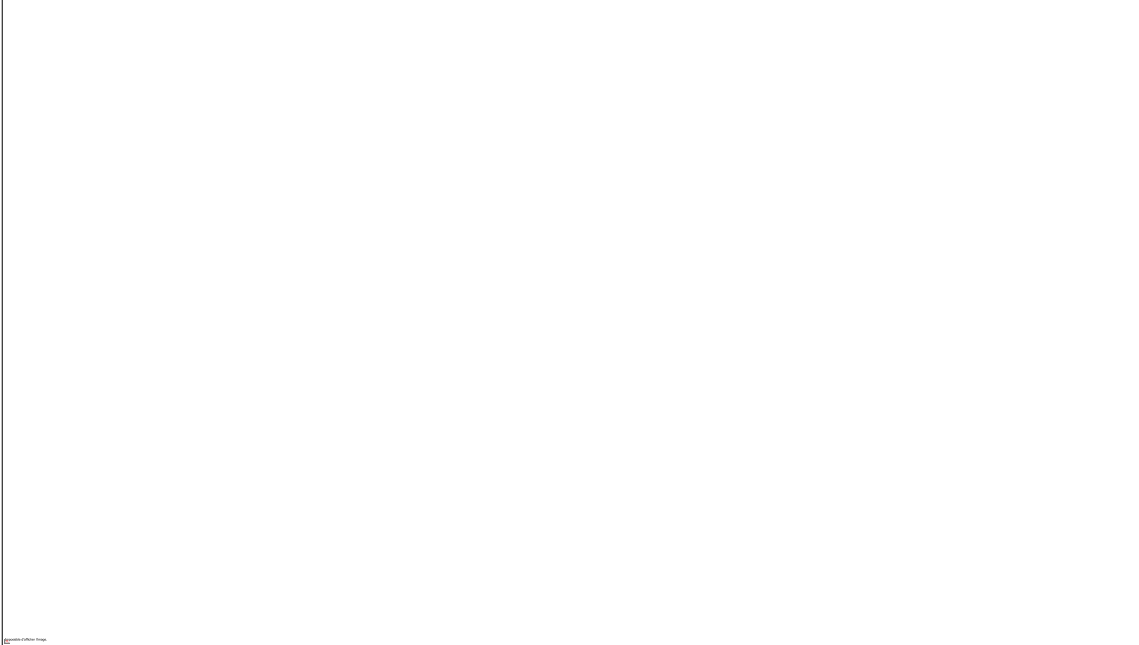
3.3. Liste des composants :

- Résistance (320 Ω)
- LED
- Raspberry PICO WH
- Raspberry PI 3
- Breadborad
- Détecteur niveau luminosité
- Caméra

-6 fils male male

-4 fils male femelle

3.4. Schéma électrique :



3.5. Fonctionnement du circuit :

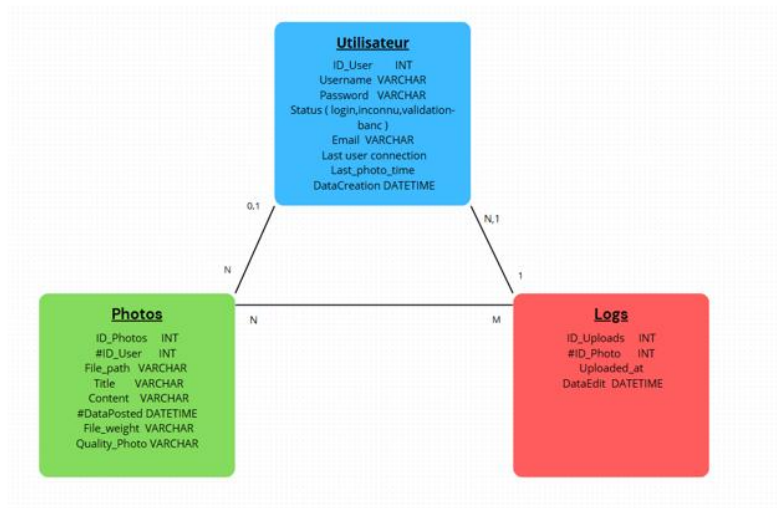
Si le capteur de luminosité capte plus de 20 lux alors la LED reste éteinte cependant si elle va en dessous alors elle s'allume, la PICO va permettre ça en liaison avec la Raspberry et la caméra permettra de prendre une photo.

3.6. Remarques et consignes de sécurité :

- Vérifier toutes les connexions avant l'alimentation.
- Ne jamais connecter la LED sans résistance.
- Respecter la polarité et les tensions maximales des composants.

4. Schémas de la Base de Données

Base de Données S1



Base de Données S2

Détail des entités et champs :

User

- ID_User (*PK*) : Identifiant unique.
- Username : Nom d'utilisateur (unique).
- Password : Mot de passe.
- Role : Rôle de l'utilisateur (user, admin, etc.).
- Status : Statut du compte.
- NB_Tentative : Nombre de tentatives de connexion.

Photo

- ID_Photo (*PK*) : Identifiant de la photo.
- ID_User (*FK*) : Utilisateur associé.
- File_Path : Chemin du fichier photo.
- Title_Photo : Titre de la photo.
- File_Weight : Taille du fichier.
- Quality_Photo : Qualité perçue ou mesurée.
- Date_Uploaded : Date d'envoi.

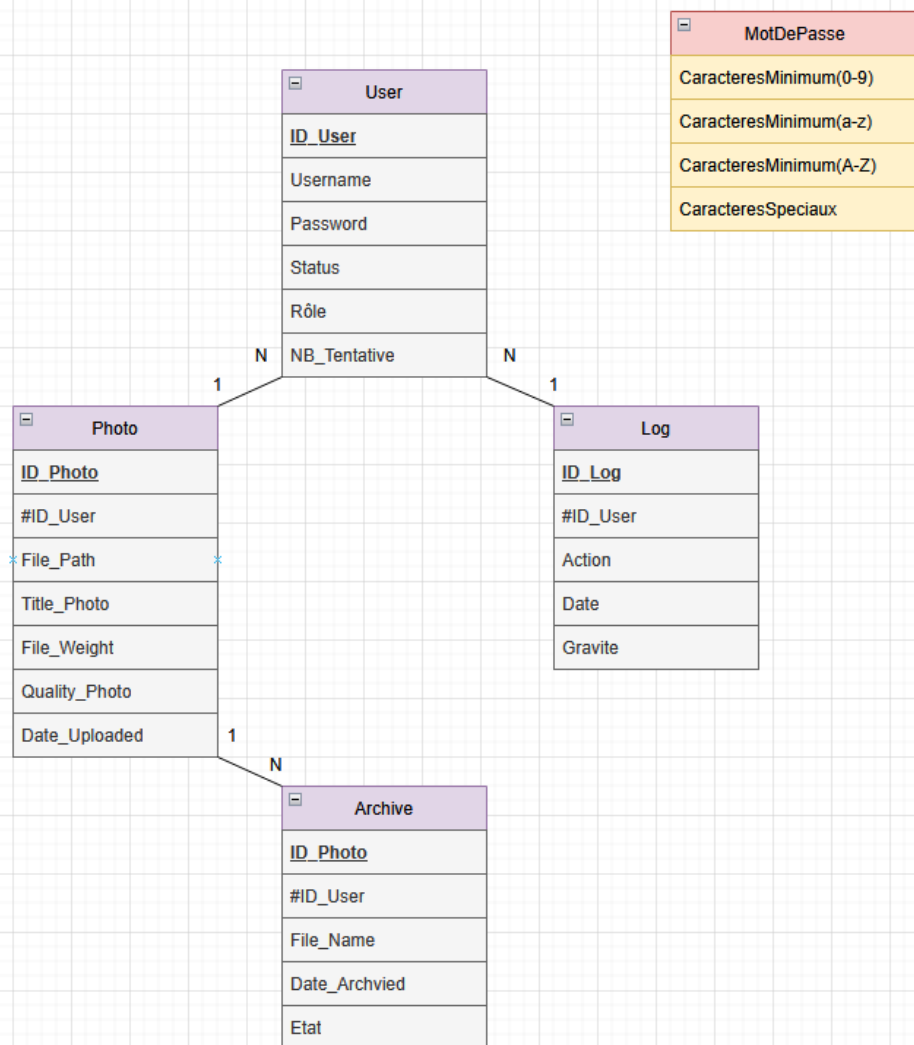
Archive

- ID_Archive (*PK*) : Identifiant d'archive.
- ID_User (*FK*) : Utilisateur ayant archivé.
- File_Name : Nom du fichier archivé.

- Date_Archived : Date d'archivage.
- Etat : État de l'archive.

Log

- ID_Log (*PK*) : Identifiant du log.
- ID_User (*FK*) : Utilisateur concerné.
- Action : Action réalisée.
- Date : Date/heure de l'action.
- Gravite : Gravité de l'événement.



5. Description de l'Installation et Configuration (Raspberry Pi 3 + PICO WH)

Installation et Configuration du Raspberry Pi 3 et du Pico WH

Introduction

Nous allons vous accompagner de A à Z dans l'installation complète d'un système Raspberry Pi 3 en interaction avec un Raspberry Pi Pico WH. L'objectif est de mettre en place une solution autonome capable de prendre des photos automatiquement, mesurer la luminosité ambiante via un capteur connecté au Pico, et allumer une LED si besoin. Toutes les étapes sont détaillées afin que même un débutant puisse tout refaire depuis zéro.

1. Matériel nécessaire

Avant de commencer, il est essentiel de rassembler tous les composants matériels et logiciels nécessaires. Vous aurez besoin d'un Raspberry Pi 3 avec son alimentation, une carte microSD d'au moins 16 Go, une caméra USB compatible UVC, un Raspberry Pi Pico WH (doté du Wi-Fi), une LED avec résistance de 220Ω, un capteur de luminosité (photocellule ou autre capteur analogique), ainsi qu'une breadboard avec des fils Dupont pour les connexions. Vous devrez également avoir un PC avec accès à Internet pour télécharger les outils nécessaires.

2. Téléchargement des logiciels nécessaires

Avant d'installer quoi que ce soit, commencez par télécharger les outils suivants :

- **Raspberry Pi Imager** pour flasher le système d'exploitation sur la carte SD.
- **Thonny IDE**, qui servira à programmer le Raspberry Pi Pico WH en MicroPython.
- **MicroPython (fichier .uf2)** pour le Pico W depuis le site officiel.
- (Optionnel mais utile) **SD Card Formatter** pour formater proprement la carte SD.

Ces outils sont gratuits et disponibles pour Windows, macOS ou Linux. Assurez-vous d'avoir une connexion stable et que les téléchargements sont bien complets avant de passer à l'étape suivante.

Installation de MicroPython et Thonny sur le Raspberry Pi

Pour programmer le Raspberry Pi Pico WH en MicroPython, il est nécessaire d'installer l'environnement de développement Thonny sur le Raspberry Pi, puis d'y intégrer MicroPython. Voici les étapes détaillées à suivre :

Commencez par ouvrir un terminal sur le Raspberry Pi. Assurez-vous que le système est bien à jour avec les commandes suivantes :

```
sudo apt update && sudo apt upgrade -y
```

Ensuite, installez l'environnement Thonny :

```
sudo apt install thonny -y
```

L'installation de Thonny peut prendre quelques minutes. Une fois terminée, vous pourrez le lancer depuis le menu principal, catégorie "Programmation" > "Thonny Python IDE".

À ce stade, il est également important d'installer certains paquets Python nécessaires pour le pilotage de composants électroniques. Le plus important pour notre projet est le package `rpi.gpio`.

```
sudo apt install python3-rpi.gpio -y
```

Ce package permet d'interagir avec les broches GPIO du Raspberry Pi, c'est-à-dire les broches physiques présentes sur la carte, que l'on utilise pour envoyer ou recevoir des signaux électriques. Cela est indispensable dans un contexte où vous devez contrôler une LED (en envoyant un signal haut ou bas) ou lire les données d'un capteur de luminosité (en lisant la tension sur une broche d'entrée).

Grâce à `rpi.gpio`, vous pourrez programmer ces interactions depuis des scripts Python exécutés sur le Raspberry Pi. Ce package fonctionne directement avec le Raspberry Pi, mais notez que la communication avec le Pico WH se fera principalement par Wi-Fi, et non par les broches GPIO locales du Pi.

Installation de MicroPython sur le Pico WH

Avant de programmer le Pico WH, vous devez y installer le firmware MicroPython. Pour cela, suivez les étapes ci-dessous :

1. Appuyez et maintenez le bouton `BOOTSEL` sur le Pico.
2. Tout en maintenant ce bouton, connectez le Pico à un port USB du Raspberry Pi.
3. Une nouvelle clé USB nommée "RPI-RP2" devrait apparaître automatiquement.
4. Depuis votre navigateur, allez sur le site officiel de MicroPython : <https://micropython.org/download/rp2-pico-w/>

-
5. Téléchargez le fichier .uf2 correspondant au Pico WH.
 6. Glissez-déposez ce fichier dans la clé USB "RPI-RP2". Le Pico redémarrera automatiquement avec MicroPython installé.

Configuration de Thonny pour programmer le Pico

Une fois MicroPython installé sur le Pico, vous pouvez le programmer via Thonny. Voici comment configurer l'interpréteur :

1. Ouvrez Thonny.
2. Allez dans le menu "Outils" > "Options...".
3. Cliquez sur l'onglet "Interpréteur".
4. Dans le menu déroulant, choisissez "MicroPython (Raspberry Pi Pico)".
5. Sélectionnez le port série correspondant au Pico (exemple : /dev/ttyACM0).
6. Validez avec "OK".

Thonny est maintenant prêt à programmer votre Pico WH. Vous pouvez créer un nouveau fichier Python et tester un petit script comme :

```
from machine import Pin
led = Pin(15, Pin.OUT)
led.value(1)
```

Cliquez sur le bouton "Exécuter" pour voir la LED s'allumer. Ensuite, vous pourrez sauvegarder ce fichier sur le Pico en tant que main.py, ce qui le fera s'exécuter automatiquement au démarrage de l'alimentation.

Ces étapes permettent d'assurer une configuration fiable, reproductible et entièrement locale de votre environnement de développement pour le Pico WH. Une fois cette base installée, vous pouvez intégrer vos propres scripts pour communiquer avec le Raspberry Pi, allumer des LEDs ou lire des capteurs comme dans les chapitres suivants du tutoriel.

3. Installation du système sur la carte SD

Une fois les outils prêts, insérez la carte microSD dans le PC. Formatez-la si besoin avec SD Card Formatter. Ensuite, ouvrez Raspberry Pi Imager, sélectionnez "Raspberry Pi OS (32-bit)", choisissez la carte SD détectée, et cliquez sur "Écrire". Cette opération peut prendre plusieurs minutes. Une fois terminée, éjectez proprement la carte.

4. Premier démarrage du Raspberry Pi

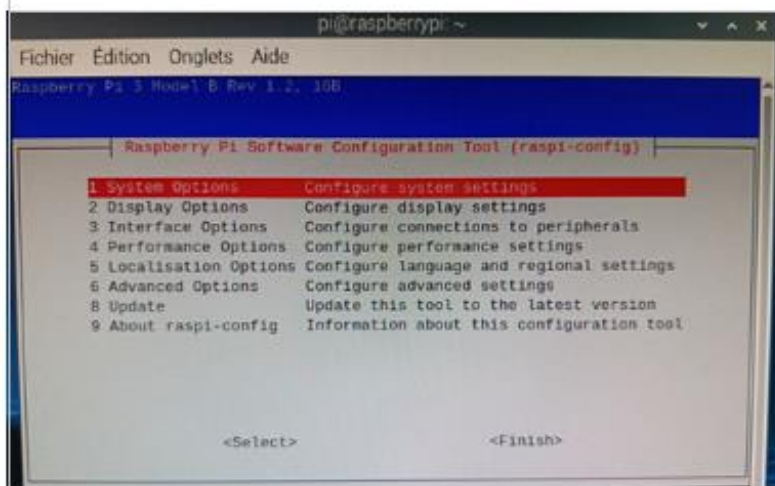
Insérez la carte SD dans le Raspberry Pi, puis branchez le clavier, la souris et l'écran HDMI. Connectez l'alimentation pour démarrer. La première fois, l'assistant de configuration s'affichera. Suivez les étapes :

Sélection de la langue, configuration du clavier (passage en AZERTY), connexion Wi-Fi, mise à jour système. Une fois l'installation initiale complétée, ouvrez un terminal depuis le bureau du Raspberry Pi.

5. Configuration système via le terminal

Dans le terminal, tapez la commande configuration vous permettant :

Cela ouvrira `sudo raspi-config` une interface de configuration vous permettant :



- de changer le nom du Raspberry Pi (hostname).
- de configurer l'heure, la langue, et le clavier.
- d'activer les interfaces nécessaires : SSH (pour la connexion à distance), I2C (communication avec capteurs), SPI et le port série.

Une fois les modifications effectuées, redémarrez le Raspberry Pi. Ensuite, mettez le système à jour avec :

```
sudo apt update && sudo apt upgrade -y
```

6. Configuration réseau en IP statique

Pour faciliter la communication entre le Pi et d'autres appareils, il est conseillé de lui assigner une adresse IP fixe. Éditez le fichier de configuration :

```
sudo nano /etc/dhcpd.conf
```

Ajoutez les lignes suivantes à la fin :

```
interface wlan0
static ip_address=192.168.1.50/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Sauvegardez avec Ctrl+O, puis quittez avec Ctrl+X.
Redémarrez le Pi. Tapez que l'adresse a bien été prise en compte.

Pour vérifier

7. Installation des outils web et caméra

Vous allez maintenant installer Apache pour héberger un mini-site local, PHP pour le script de déclenchement de photo, et fswebcam pour utiliser la caméra USB. Tapez :

```
sudo apt install apache2 php libapache2-mod-php fswebcam
```

Branchez la caméra et vérifiez sa détection :

```
ls /dev/video0
```

Prenez une photo test :


```
fswebcam -r1280x720 --no-banner /home/pi/test.jpg
```

Créez un dossier pour stocker les photos dans Site-photo-ATB/uploads, avec :

```
sudo mkdir /var/www/html/uploads  
sudo chmod 777 /var/www/html/uploads
```

8. Installation de MicroPython sur le Pico WH

Branchez le Pico WH à l'ordinateur en maintenant le bouton BOOTSEL. Une fois détecté comme clé USB,

Glissez le fichier  de MicroPython. Ensuite, ouvrez Thonny. Dans le menu Outils > Interpréteur,

Sélectionnez "MicroPython (Raspberry Pi Pico)" et le bon port série.

9. Connexion SSH : Ouverture de port NAT

Si vous voulez accéder à votre serveur à distance :

1. Connectez-vous à votre box (<http://192.168.1.1>)
2. Allez dans les paramètres NAT / redirection de port
3. Redirigez le port 80 vers l'adresse IP fixe du Pi (ex: 192.168.1.50)

Pour établir la connexion SSH depuis un autre appareil on va utiliser l'application Putty avec les paramètres définis au préalable tel que :

L'adresse IP du Raspberry et le port sur lequel on a établi le SSH.

6. Configuration IP – Réseau

Décrire les étapes nécessaires pour configurer le Raspberry Pi avec une adresse IP spécifique sur un sous-réseau donnés :

Algorithme main.py

```

import network
import socket
import time
from machine import Pin, ADC
import ubinascii

# === Wi-Fi ===
ssid = "Redmi Note 14" # ou ton réseau
password = "jasmine06"

wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect(ssid, password)

while not wlan.isconnected():
    print("Connexion...")
    time.sleep(1)

ip = wlan.ifconfig()[0]
mac = ubinascii.hexlify(wlan.config('mac'), ':').decode()
print(f"✅ Connecté à {ip}, MAC : {mac}")

# === Envoi régulier de MAC/IP par UDP (si connecté) ===
udp = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
udp.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)

def envoyer_presence():
    try:
        if wlan.isconnected():
            msg = f"PICO_HELLO|MAC={mac}|IP={ip}"
            udp.sendto(msg.encode(), ("255.255.255.255", 12345))
            print("📡 Message envoyé :", msg)
        else:
            print("⚠️ Non connecté au Wi-Fi, envoi annulé")
    except Exception as e:
        print("❌ Erreur envoi UDP :", e)

# === Capteur et LED ===
led = Pin(15, Pin.OUT)
capteur = ADC(26)
SEUIL = 20000

# === Serveur HTTP ===
http = socket.socket()
http.bind(("0.0.0.0", 80))
http.listen(1)
print("🌐 Serveur HTTP actif sur", ip)

```

```

# === Boucle principale ===
while True:
    envoyer_presence()

    # Attente max 10 sec d'un client HTTP
    http.settimeout(10)
    try:
        cl, addr = http.accept()
        request = cl.recv(1024).decode()
        print("🔗 Requête HTTP :", request)

        if "/prepare_photo" in request:
            lum = capteur.read_u16()
            if lum < SEUIL:
                led.value(1)
                reponse = f"LED ON - Luminosité : {lum}"
            else:
                led.value(0)
                reponse = f"LED OFF - Luminosité : {lum}"
        else:
            reponse = "Commande inconnue"

        cl.send("HTTP/1.0 200 OK\r\nContent-type: text/plain\r\n\r\n")
        cl.send(reponse)
        cl.close()

    except OSError:
        # Pas de requête reçue dans le délai, on continue
        pass

    time.sleep(10) # pause avant prochaine annonce

```

Algorithme testled.py

```

from machine import Pin, ADC
from time import sleep

# Capteur analogique sur GP26 (ADC0)
ldr = ADC(26) # ADC0 = GP26

```

```
# LED sur GP15 (change si tu utilises une autre broche)
led = Pin(15, Pin.OUT)

# Seuil de déclenchement (à ajuster selon ton capteur)
SEUIL = 11900 # Luminosité faible si en-dessous

valeur = ldr.read_u16()
print("Lumière mesurée :", valeur)

if valeur < SEUIL:
    led.value(1) # Allume LED
    sleep(4)
    led.value(0)
else:
    led.value(0) # Éteint LED
```

7. Plan de Validation, Procédures de Test et Fiches d'Anomalies

<Product> Verification / Validation Report					
Test <u>id</u> : 1 Raspberry/BdD		Issue: v1.0 Date: 09/06/2025 Author: Océan	System Under Test: Raspberry/BdD Issue SUT: v1.0	at Date, Time: Start: 21/03/2025 End: 14/04/2025	Verification Result: pass
Requirement-Number: 2.1.1.3 — Stockage des photos sur le <u>serveur</u> local (Raspberry Pi) 2.1.1.6 Assurer une <u>latence maximale</u> de 15 <u>secondes</u> entre la capture et le stockage. 2.1.3.10 <u>Éviter</u> les surcharges (pages <u>inutiles</u> , trop de photos <u>chargées</u>).	Open NCRs:			Test Report Executed by: Mattéo Test Report Check & Approved by: Océan	<div><div>FAIL</div><div>PASS</div></div>
Test Description: <u>Insérer</u> la base de données dans la Raspberry					
Test Preparation:					
No.	Activity Description				
i	<u>Démarrer</u> le <u>serveur</u> de base de données				
ii	Tester la connexion avec un script PHP				
iii	<u>Vérifier l'insertion</u> et lecture de données				
Test Execution:					
No.	Activity Description	Pass/Fail Criteria	Status (Pass, Failed)	Remarks	
1	Connexion à la base	<u>Réussie sans erreur</u>	pass	Logs non adaptés ou mal liés à la base de données, affichage erroné de certaines informations	
2	Insertion d' <u>une</u> donnée test	Donnée visible dans la table	pass		
3	Suppression / Modification	Modification prise <u>en compte correctement</u>	pass		
Execution duration: X hours					
Summary:					

<Product> Verification / Validation Report					
Test Id : 2 Raspberry conf/Pico		Issue: v1.0 Date: 09/06/2025 Author: Océan	System Under Test: Raspberry Issue SUT: v1.0	at Date, Time: Start: 29/03/2025 End: 11/05/2025	Verification Result: pass
Requirement-Number:	Open NCRs:			Test Report Executed by: Thibault Test Report Check & Approved by: Mattéo	<div><div>FAIL</div><div>PASS</div></div>
2.1.1.7 Gérer automatiquement un éclairage d'appoint en cas de luminosité insuffisante. 2.1.4.1 Organiser les photos par date, heure, utilisateur et origine (manuelle/automatique). 2.1.4.3 Assurer la persistance des données après redémarrage du Raspberry Pi.					
Test Description:					
Configuration complète de la Raspberry/Liaison Pico					
Test Preparation:					
No.	Activity Description				
i	Etablir la liaison entre le Pico et la Pi				
ii	Charger le script de contrôle LED				
iii	Lancer la commande de test depuis le Pi				
Test Execution:					
No.	Activity Description	Pass/Fail Criteria	Status (Pass, Failed)	Remarks	
1	Envoi de commande ON/OFF	LED s'allume et s'éteint à la demande	pass	Création d'un utilisateur anonyme forcée car User_ID dans la table User était NOT NULL (valeur forcée à anonyme=0)	
2	Lecture de retour de commande	Confirmation reçue	pass		
3	Stabilité de réponse	Réponse immédiate sans bug	pass		
Execution duration: X hours					
Summary:					


<Product> Verification / Validation Report						
Test id : 3 Similitude Maquette (site web)		Issue: v1.0 Date: 09/06/2025 Author: Océan		System Under Test: Site web	at Date, Time: Start: 22/03/2025 End: 30/03/2025	Verification Result: pass
Requirement-Number: 2.1.3.1 Le site doit être accessible localement sans connexion Internet. 2.1.3.8 Le site doit être responsive (adaptatif à tout type d'appareil). 2.1.3.4 Afficher une galerie organisée avec filtres (date, utilisateur, statut, etc.).		Open NCRs:	Issue SUT: v1.0		Test Report Executed by: Dorian Test Report Check & Approved by: Océan	<div>FAIL</div> <div>PASS</div>
Test Description: Réussir à donner un rendu presque similaire que la maquette du site web réalisé lors du premier semestre						
Test Preparation:						
No.	Activity Description					
i	Necessite site web					
ii	Maquette du site web					
iii						
Test Execution:						
No.	Activity Description	Pass/Fail Criteria		Status (Pass, Failed)	Remarks	
1	Comparer site web avec les différents type d'utilisateurs			pass		
2				pass		
3				pass		
Execution duration: X hours						
Summary:						


<Product> Verification / Validation Report					
Test Id : 4 Caméra (Photo)		Issue: v1.0 Date: 09/06/2025 Author: Océan	System Test: Caméra Issue SUT: v1.0	Under at Date, Time: Start: 29/05/2025 End: 05/06/2025 Test Report Executed by: Dorian Test Report Check & Approved by: Thibault	Verification Result: pass <div> <div>FAIL</div> <div>PASS</div> </div>
Requirement-Number: 2.1.1.1 Gérer la prise de photo manuelle. 2.1.1.2 Gérer la prise de photo automatique si aucune photo n'est prise sous 24h. 2.1.4.2 Gérer un historique étendu (capacité importante d'archivage).		Open NCRs:			
Test Description: Tester si la caméra pour les prochaines prises de photos fonctionne correctement					
Test Preparation:					
No.	Activity Description				
i	Maréiels				
ii	Site web				
iii					
Test Execution:					
No.	Activity Description	Pass/Fail Criteria	Status (Pass, Failed)	Remarks	
1	Prendre une photo depuis le terminal	La photo est de bonne qualité	pass		
2			pass		
3	Tester la caméra sur le site web	Voir la caméra sur le site web	pass		
Execution duration: X hours					
Summary:					


<Product> Verification / Validation Report					
Test Id : 5 System		Issue: v1.0 Date: 09/06/2025 Author: Océan	System Test: System (Web + Capteurs + LED) Issue SUT: v1.0	Under at Date, Time: Start: 24/05/2025 End: 02/06/2025 Test Report Executed by: Dorian Test Report Check & Approved by: Mattéo	Verification Result: pass <div> <div>FAIL</div> <div>PASS</div> </div>
Requirement-Number: 2.1.1.7 Gérer automatiquement un éclairage d'appoint en cas de luminosité insuffisante. 2.1.3.2 Intégrer un bouton de prise de photo sur la page de connexion. 2.1.4.1 Organiser les photos par date, heure, utilisateur et origine (manuelle/automatique).		Open NCRs:			
Test Description: Valider le bon fonctionnement global du système complet (web, capteurs, LED).					
Test Preparation:					
No.	Activity Description				
i	Lancer tous les services (web, capteur, LED...)				
ii	Vérifier les connexions				
iii	Lancer un scénario complet de test utilisateur				
Test Execution:					
No.	Activity Description	Pass/Fail Criteria	Status (Pass, Failed)	Remarks	
1	Lancement du site et capteur	Affichage des valeurs de luminosité en direct	pass		
2	Réaction LED selon luminosité	LED activée/désactivée selon le seuil programmé	pass		
3	Vérification de la base de données	Les données sont enregistrées et lisibles	pass		
Execution duration: X hours					
Summary:					


<Product> Verification / Validation Report					
Test Id : 6 Archivage		Issue: v1.0 Date: 09/06/2025 Author: Océan	System Test: Archivage Issue SUT: v1.0	Under at Date, Time: Start: 26/05/2025 End: 29/06/2025 Test Report Executed by: Dorian Test Report Check & Approved by: Océan	Verification Result: pass <div> <div>FAIL</div> <div>PASS</div> </div>
Requirement-Number: 1.4.2 Gérer un historique étendu (capacité importante d'archivage). 1.4.4 Intégrer une stratégie de gestion du stockage (suppression ou archivage). 2.2.9.3 Politique d'archivage (durée, suppression automatique).		Open NCRs:			
Test Description: quand on supprime une photo on est sensé la retrouver dans l'archivage					
Test Preparation:					
No.	Activity Description				
i	Site web				
ii	Base de donnée				
iii					
Test Execution:					
No.	Activity Description	Pass/Fail Criteria	Status (Pass, Failed)	Remarks	
1	Tester le système d'archivage	Retrouver la photo dans l'archivage	pass		
2			pass		
3			pass		
Execution duration: X hours					
Summary:					

<Product> Verification / Validation Report					
Test Id : 7 capteur de luminosité		Issue: v1.0 Date: 09/06/2025 Author: Océan	System Test: Archivage Issue SUT: v1.0	Under at Date, Time: Start: 26/05/2025 End: 29/06/2025 Test Report Executed by: Dorian Test Report Check & Approved by: Océan	Verification Result: pass <div> <div>FAIL</div> <div>PASS</div> </div>
Requirement-Number:		Open NCRs:			
Test Description: quand on supprime une photo on est sensé la retrouver dans l'archivage					
Test Preparation:					
No.	Activity Description				
i	Connecter le capteur				
ii	Charger le script Python de lecture				
iii	Placer le capteur sous différentes luminosités				
Test Execution:					
No.	Activity Description	Pass/Fail Criteria	Status (Pass, Failed)	Remarks	
1	Lecture en faible lumière	Valeur détectée faible	pass		
2	Lecture en forte lumière		pass		
3	Variation progressive testée	Courbe cohérente selon l'intensité lumineuse	pass		
Execution duration: X hours					
Summary:					

<Product> Verification / Validation Report					
Test Id : 8 les images qui <u>s'incrémente</u> correctement dans la bdd/site web		Issue: v1.0 Date: 09/06/2025 Author: Océan	System Test: bdd/site web Issue SUT: v1.0	Under at Date, Time: Start: 26/05/2025 End: 16/06/2025	Verification Result: pass
Requirement-Number: 1.1.3 Les photos <u>doivent être stockées</u> sur le <u>serveur</u> local (Raspberry Pi). 1.1.4 Compresser les photos <u>avant enregistrement</u> pour <u>économiser</u> de l'espace <u>si nécessaire</u> . 1.4.1 <u>Organiser</u> les photos <u>par date, heure, utilisateur</u> et origine.	Open NCRs:				
Test Description: quand on <u>supprime une</u> photo on <u>est censé</u> la <u>retrouver</u> dans l' <u>archivage</u>					
Test Preparation:					
No.	Activity Description				
i	Base de donnée				
ii	Site web				
iii	Matériels				
Test Execution:					
No.	Activity Description	Pass/Fail Criteria		Status (Pass, Failed)	Remarks
1				pass	
2				pass	
3				pass	
Execution duration: X hours					
Summary:					

<Product> Verification / Validation Report					
Test Id : 9 Vérification du login		Issue: v1.0 Date: 09/06/2025 Author: Océan	System Test: login Issue SUT: v1.0	Under at Date, Time: Start: 12/03/2025 End: 2/06/2025	Verification Result: pass
Requirement-Number: 2.2.2.3 Refuser l'accès si les identifiants sont incorrects. 2.2.2.1 Authentification via identifiant et mot de passe.	Open NCRs:				
Test Report Executed by: Dorian Test Report Check & Approved by: Océan					
Test Description: Vérification du Login					
Test Preparation:					
No.	Activity Description				
i	Charger la page de connexion				
ii	Tester des identifiants valides				
iii	Tester des identifiants invalides				
Test Execution:					
No.	Activity Description	Pass/Fail Criteria	Status (Pass, Failed)	Remarks	
1	Connexion avec identifiants valides Redirection vers la page d'accueil		pass	Toutes les vérifications ont été correctement bloquées selon le rôle et la validité des <u>credentials</u> .	
2	Connexion avec mot de passe erroné Accès refusé		pass		
3	Connexion avec compte bloqué Message d'erreur affiché		pass		
Execution duration: X hours					
Summary:					

<Product> Verification / Validation Report						
Test <u>Id</u> : 10 Validation de l'ID User dans les Tables		Issue: v1.0 Date: 09/06/2025 Author: Océan	System Under Test: <u>Id_user</u> dans tables Issue SUT: v2.0	at Date, Time: Start: 26/05/2025 End: 29/05/2025		Verification Result: pass
Requirement-Number: 2.1.4.1 Organiser les photos par utilisateur. 2.1.4.5 Fournir un schéma clair de la base de données.	Open NCRs:			Test Report Executed by: Mattéo Test Report Check & Approved by: Océan		
						
Test Description: Validation de l' <u>id</u> dans les tables						
Test Preparation:						
No.	Activity Description					
i	Insérer un utilisateur test					
ii	Uploader une photo					
iii	Créer un log d'action					
Test Execution:						
No.	Activity Description	Pass/Fail Criteria		Status (Pass, Failed)	Remarks	
1	Vérifier <u>ID_User</u> dans Photo ID correspond bien à l'utilisateur			pass		
2	Vérifier <u>ID_User</u> dans Archive Lien utilisateur conservé			pass	Bonne cohérence des relations entre les tables via l'ID utilisateur.	
3	Vérifier <u>ID_User</u> dans Log Correspondance exacte avec action			pass		
Execution duration: X hours						
Summary:						

<Product> Verification / Validation Report					
Test <u>Id</u> : 11 Évaluation de la performance du site web sur Raspberry Pi		Issue: v1.0 Date: 09/06/2025 Author: Océan	System Under Test: Site Web local Issue SUT: v1.0	at Date, Time: Start: 7/04/2025 End: 12/05/2025	Verification Result: pass
Requirement-Number:	Open NCRs:			Test Report Executed by: Thibault Test Report Check & Approved by: Océan	
1.1.3 Les photos <u>doivent être stockées</u> sur le <u>serveur</u> local (Raspberry Pi). 1.1.4 <u>Compresser</u> les photos <u>avant enregistrement</u> pour <u>économiser</u> de l' <u>espace si nécessaire</u> .					
Test Description: Évaluation de performance site web sur <u>raspberry</u> pico					
Test Preparation:					
No.	Activity Description				
i	Charger plus de 200 photos				
ii	Simuler 3 utilisateurs simultanés				
iii	Surveiller l'utilisation mémoire et CPU				
Test Execution:					
No.	Activity Description	Pass/Fail Criteria	Status (Pass, Failed)	Remarks	
1	Temps de chargement galerie < 15 secondes		pass		
2	Latence de clic entre pages < 500 ms		pass		
3	Stabilité de réponse Aucune erreur serveur		pass	Le site reste stable et fluide jusqu'à ~250 photos, au-delà la mémoire swap commence à être sollicitée.	
Execution duration: X hours					
Summary:					

7.1 Les risques

Registre des Risques					
Projet :	Projet Photo Thales	Groupe :	Thales2-photo-atb	Date de Mise à jour:	26-nov
Id :		Créateur:	Dorian,Thibault,Ocean,Matteo	Date création:	24-nov
Titre :	Mauvaise précision du système de capture d'image.				
Cause :	Caméra peut rencontrer des dysfonctionnements, comme un mauvais positionnement ou un problème d'autofocus.				
Conséquence :	Images floues ou inutilisables.				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
4	2		X		technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	tester les équipements avant l'utilisation				
	Réaction attendue:	Réaligner ou recalibrer manuellement la caméra si un problème de positionnement est détecté.			
	Statut action(s):	technique			
Actions à mettre en place si risque rencontré :	Changer le materiel si nécessaire				

Titre :	Panne du système d'éclairage automatique				
Cause :	Capteur de luminosité défaillant				
Conséquence :	Images inexploitable car trop sombres				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
3	2		X		technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	tester les équipements avant l'utilisation				
	Réaction attendue:	Ajouter un système d'activation manuelle de l'éclairage			
	Statut action(s):	technique			
Actions à mettre en place si risque rencontré :	Changer le materiel si nécessaire				

Titre :	Prise de retard sur le projet				
Cause :	Procrastination ou sous-estimation du temps nécessaire pour certaines tâches.				
Conséquence :	Charge de travail conséquente à effectuer dans des délais courts				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
3	4		X		planning
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	Travaille sur la partie planning de la gestion de projet				
	Réaction attendue:	Organisation au sein de l'équipe			
	Statut action(s):	planning			
Actions à mettre en place si risque rencontré :	Réaction immédiate de la part de l'équipe				

Titre :	Plusieurs personnes travaillent comme R, A, C ou I sur la même chose				
Cause :	Mauvaise organisation et attribution des rôles				
Conséquence :	Méthode de travail inefficace donc pertes de temps				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
2	1			X	planning
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	Travaille sur la partie RACI de la gestion de projet				
	Réaction attendue:	Organisation au sein de l'équipe			
	Statut action(s):	planning			
Actions à mettre en place si risque rencontré :	Réaction immédiate de la part de l'équipe				

Titre :	Délais trop long entre la requête et la prise de la photo				
Cause :	Mauvaise optimisation du code python				
Conséquence :	Décalage entre la date de capture renseignée sur le site et la date de capture réel				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
1	3		X		technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	oui		Risque refusé (4) :	non	
Actions pour prévenir le risque :	Changer le code python pour permettre un délais moins long				
	Réaction attendue:				
	Statut action(s):				
Actions à mettre en place si risque rencontré :	Réaction immédiate de la part de l'équipe				

Titre :	Mauvaise utilisation de la prise manuelle de photo				
Cause :	Pas d'information concernant le fonctionnement de la prise manuelle				
Conséquence :	Problèmes de sauvegarde				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
2	2			X	technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	Accessibilité de la documentation sur le système.				
	Réaction attendue:	Créations de readme			
	Statut action(s):	technique			
Actions à mettre en place si risque rencontré :	Documentation plus importante				

Titre :	cyberattaque				
Cause :	Mot de passe pas assez sécurisé				
Conséquence :	perte des données et photos				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
5	1	X			technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	Obligation de mot de passe sécurisé (8 caractères, 1 symbole, 1 Majuscule, 1 chiffre)				
	Réaction attendue:	Dépôt de plainte			
	Statut action(s):	technique			
Actions à mettre en place si risque rencontré :	Meilleure sécurisation du site web				

Titre :	Site web inaccessible				
Cause :	Surcharge du site du à un déni de service				
Conséquence :	Inaccessibilité momentanée au site web				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
3	3	X			technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	1 seul utilisateur accède au site à la fois				
	Réaction attendue:	Fermeture du serveur			
	Statut action(s):	technique			
Actions à mettre en place si risque rencontré :	Analyse du code permettant la limite de 1 utilisateur				

Titre :	Site web inaccessible sur certaines plateformes				
Cause :	Site non responsive-design				
Conséquence :	Site web illisible sur smartphone par exemple				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
3	5		X		technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	Site responsive design				
	Réaction attendue:	Site responsive design			
	Statut action(s):	technique			
Actions à mettre en place si risque rencontré :	Site responsive design				

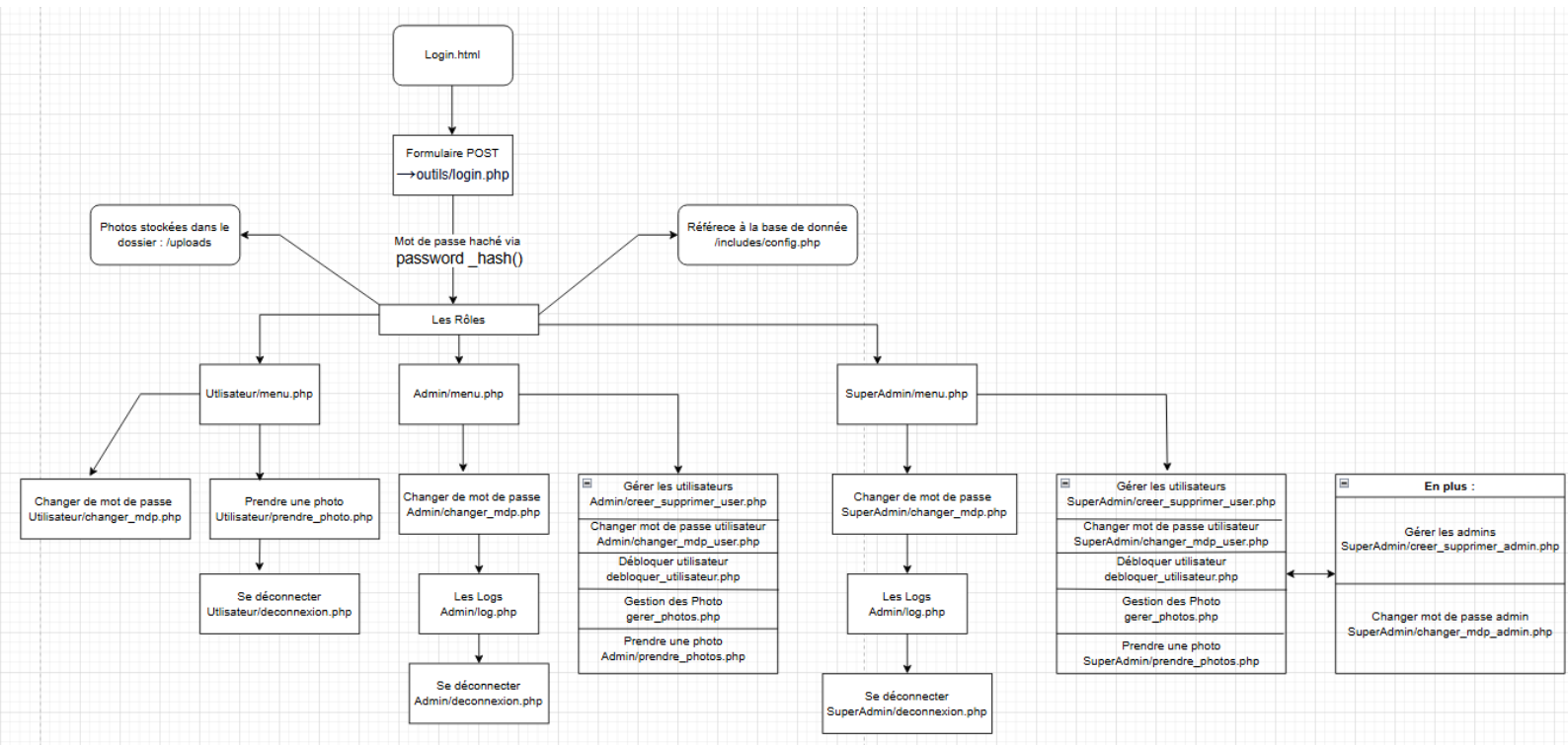
Titre :	Base de donnée surchargée				
Cause :	Photos trop volumineuses				
Conséquence :	Stockage de nouvelles photos impossible				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
5	2	X			technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	Bon formatage des photos effectuées				
	Réaction attendue:	Transfert sur un disque dur externe			
	Statut action(s):	technique			
Actions à mettre en place si risque rencontré :	Utilisation d'un formatage d'image plus léger comme JPEG				

Titre :	Changement dans l'organisation de l'équipe				
Cause :	Suite a retard prit, mauvaise organisation				
Conséquence :	Encore plus de retard, travaille sur la mauvaise chose				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
4	3	X			technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	éviter un retard ou n'importe quelle mauvaise organisation				
	Réaction attendue:	Retard sur le projet			
	Statut action(s):	technique			
Actions à mettre en place si risque rencontré :	Se prendre en mains des que possible sur l'équipe				

Titre :	Avis du client différent				
Cause :	Le cahier des charges du client a changé				
Conséquence :	Encore plus de retard, travaille sur la mauvaise chose				
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
4	3	X			technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	Demander au client si son avis n'a pas changé				
	Réaction attendue:	Retard sur le projet			
	Statut action(s):	technique			
Actions à mettre en place si risque rencontré :	Demander nouveauté du client(ce qu'il a ajouté)				

Titre :					
Cause :	Matériel défaillant ou manquant				
Conséquence :					
Sévérité (1)	Probabilité (1)	Rouge (2)	Jaune (2)	Vert (2)	Type (3)
5	3		X		technique
Décisions sur le risque & action(s) à mettre en œuvre					
Risque accepté (4):	non		Risque refusé (4) :	oui	
Actions pour prévenir le risque :	vérifier la boîte pour voir s'il manque pas de matériel				
	Réaction attendue:	Retard sur le projet			
	Statut action(s):	technique			
Actions à mettre en place si risque rencontré :	Demander au professeur le matériel manquant / défaillant				

8. Échanges Sécurisés entre les Pages du Site Web



Code PHP pour sécurisation pour l'admin

```

<?php
session_start();
require_once('../outils/nettoyer_logs.php');
require_once('../includes/config.php');

if (!isset($_SESSION["username"]) || $_SESSION["role"] != "Admin") {
    header("Location: ../login.html");
    exit;
}

$username = $_SESSION["username"];

// Retrieve the logged in user ID
$stmt = $pdo->prepare("SELECT ID_User FROM User WHERE Username = ?");
$stmt->execute([$_SESSION["username"]]);
$id_user = $stmt->fetchColumn();
?>

```

Code PHP pour sécurisation pour le superadmin

```

<?php
session_start();
require_once('../outils/nettoyer_logs.php');
require_once('../includes/config.php');

if (!isset($_SESSION["username"]) || $_SESSION["role"] !== "Superadmin") {
    header("Location: ../login.html");
    exit;
}

<?php
session_start();
$username = $_SESSION["username"];
require_once('../includes/config.php');

// Retrieve the logged in user ID
$stmt = $pdo->prepare("SELECT ID_User FROM User WHERE Username = ?");
$stmt->execute([$username]);
$id_user = $stmt->fetchColumn();
?>

```

Code PHP pour sécurisation pour l'utilisateur

Exemples d'échanges de notre site en mode superadmin :

Connexion

Nom d'utilisateur

Mot de passe

Se connecter

Prendre une photo

Pas encore de compte ? Créer un nouveau compte

Bienvenue, Superadmin leboss !

Menu

Galerie photo :

Prendre une photo

Synchroniser les images d'uploads

Utilisateur : Tous Année : Toutes Mois : Tous Jour : Tous Appliquer



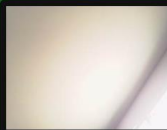
Image 9

Prise par : matteo
Le : 2025-05-19 17:38:24



Auto importée

Prise par : matteo
Le : 2025-05-18 18:14:40



Auto importée

Prise par : matteo
Le : 2025-05-18 18:13:05



Auto importée

Prise par : matteo
Le : 2025-05-16 11:02:43



Auto importée

Prise par : matteo
Le : 2025-05-15 08:43:03

Changer mon mot de passe (Superadmin)

Menu

Ancien mot de passe :

Nouveau mot de passe :

Changer

Changer le mot de passe d'un utilisateur

Menu

Nom d'utilisateur :

Nouveau mot de passe :

Changer

Changer le mot de passe d'un administrateur

Menu

Nom d'administrateur :

dorian ▼

Nouveau mot de passe :

Changer

Gestion des administrateurs

Menu

Créer un administrateur

Nom :

Mot de passe :

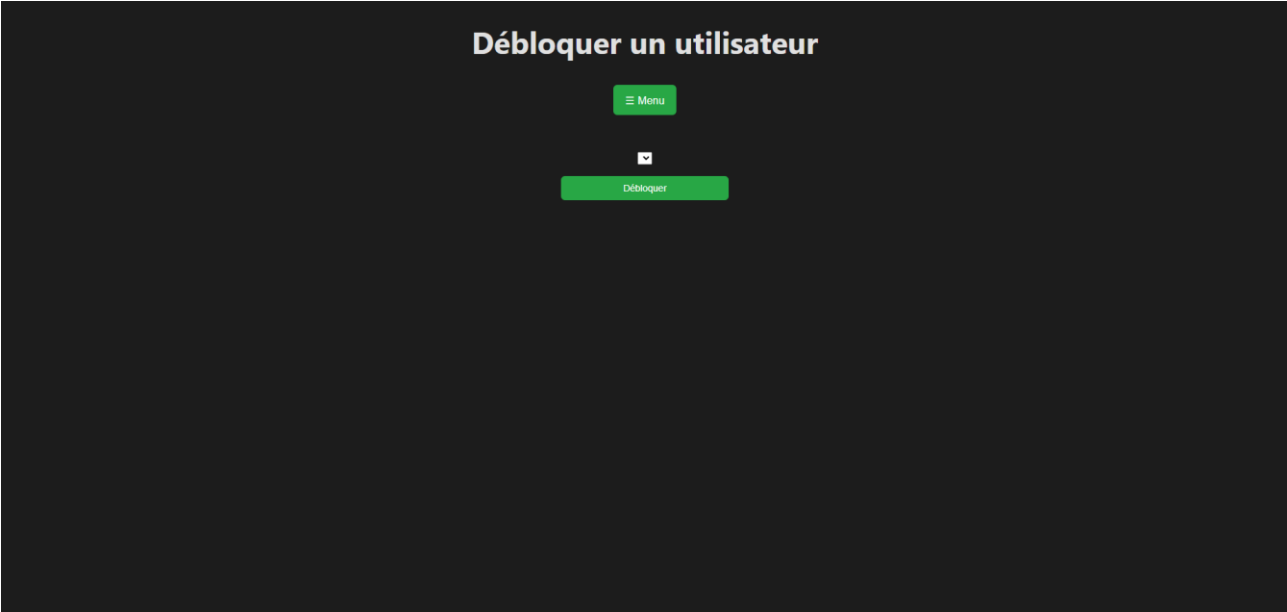
Créer

Supprimer un administrateur

Nom :

dorian ▼

Supprimer



Journal des logs				
<div>Gravité : Toutes Utilisateur : Tous Année : Toutes Mois : Tous Jour : Tous Appliquer</div>				
<div>Menu</div>				
Utilisateur	Action	Date	Gravité	
leboss	Connexion réussie	2025-06-11 22:24:30	Info	
leboss	Connexion réussie	2025-06-11 22:23:48	Info	
leboss	Connexion réussie	2025-06-07 22:15:38	Info	
matteo	Connexion réussie	2025-06-07 22:11:04	Info	
matteo	Connexion réussie	2025-06-07 21:57:47	Info	
User1	Connexion réussie	2025-06-07 21:56:45	Info	
matteo	Connexion réussie	2025-06-07 21:48:38	Info	
matteo	Connexion réussie	2025-06-07 21:43:53	Info	
leboss	Connexion réussie	2025-06-07 21:42:56	Info	
matteo	Connexion réussie	2025-06-07 21:41:14	Info	
matteo	Connexion réussie	2025-06-07 20:42:38	Info	
User3	Changement de mot de passe	2025-06-07 20:35:31	Moyen	
User3	Connexion réussie	2025-06-07 20:35:13	Info	
leboss	Créer un Utilisateur	2025-06-07 20:31:54	Moyen	
leboss	Créer un Admin	2025-06-07 20:25:43	Critique	
leboss	Connexion réussie	2025-06-07 20:25:19	Info	
matteo	Créer un Utilisateur	2025-06-07 20:20:52	Moyen	

Navigation sécurisée entre les pages web

Dans une application web multi-utilisateurs, surtout lorsqu’elle est connectée à des éléments critiques comme un capteur, une caméra ou une base de données contenant des informations personnelles, la sécurité de la navigation entre les pages est un élément central. Elle permet non seulement de protéger l’accès aux données, mais

aussi d'isoler les fonctionnalités sensibles selon les rôles d'utilisateur, tout en garantissant la cohérence du parcours dans l'application.

Dans notre projet, basé sur un système Raspberry Pi (serveur PHP/SQLite) et Raspberry Pico W (capteur de luminosité et LED), nous avons mis en place une architecture sécurisée permettant une navigation fluide mais strictement contrôlée entre toutes les pages. Voici un tour d'horizon des techniques mises en œuvre.

8.1. Authentification par sessions sécurisées

Dès qu'un utilisateur se connecte via le formulaire de login, ses identifiants sont vérifiés dans la base de données. S'ils sont valides, une session PHP est démarrée :

```
session_start();
$_SESSION["username"] = $username;
$_SESSION["role"] = $role;
$_SESSION["ID_User"] = $id_user;
```

Ces variables de session servent de ticket d'accès temporaire à toutes les pages sécurisées. Elles sont stockées côté serveur et associées à un identifiant unique via un cookie PHP sécurisé (identifiant de session).

Ensuite, chaque page vérifie l'existence de la session :

```
if (!isset($_SESSION["username"])) {
    header("Location: ../login.html");
    exit;
}
```

Sans cette vérification, un utilisateur pourrait forcer l'accès à une page en tapant l'URL manuellement.

8.2. Gestion granulaire des rôles (ACL)

Notre site repose sur une logique d'accès contrôlé par rôle (Access Control List - ACL) où chaque niveau (Anonyme, Utilisateur, Admin, Superadmin) a des droits strictement définis.

- Anonyme : peut uniquement consulter la page d'accueil et prendre une photo, sans accès aux galeries ou logs.
- Utilisateur : peut consulter ses propres photos, les archiver, modifier son mot de passe.
- Admin : peut gérer les utilisateurs, prendre et supprimer des photos, consulter les logs.
- Superadmin : dispose de tous les droits, y compris la modification des mots de passe d'autres comptes, et la gestion des comptes Admin.

Chaque page vérifie donc le rôle :

```
if ($_SESSION["role"] !== "Admin") {  
    header("Location: ../login.html");  
    exit;  
}
```

Et dans certains cas, on peut même croiser plusieurs niveaux d'autorisation :

```
if (!in_array($_SESSION["role"], ["Admin", "Superadmin"])) {  
    header("Location: ../login.html");  
    exit;  
}
```

8.3. Protection contre les accès directs

Chaque page PHP sensible commence obligatoirement par un bloc de protection. Même si quelqu'un connaît l'URL (par exemple `supprimer_photo.php?id=42`), il ne pourra pas y accéder si la session n'est pas valide, ou si son rôle ne le permet pas.

De plus, lorsqu'on clique sur des boutons d'action (supprimer, archiver, etc.), on passe généralement par une requête POST ou GET sécurisée par une vérification serveur, et non en JS uniquement.

8.4. Nettoyage des sessions et sécurité de déconnexion

Pour éviter qu'un utilisateur laisse une session active sur un poste partagé, un bouton Déconnexion est toujours présent. Lorsqu'il est cliqué, le script `logout.php` détruit la session proprement :

```
session_start();  
session_unset();  
session_destroy();  
header("Location: ../login.html");
```

En plus, un log de type 'Info' est automatiquement généré dans la base pour garder une trace de l'action :

```
INSERT INTO Log (ID_User, Action, Date, Gravite)  
VALUES (?, 'Déconnexion réussie', datetime('now', 'localtime'), 'Info');
```

8.5. Chiffrement des communications (HTTPS)

Dans un contexte de développement local, HTTP est parfois toléré, mais dès que le site est accessible sur un réseau externe, HTTPS devient indispensable. Il permet de :

- Chiffrer les données échangées entre le navigateur et le serveur.
- Protéger les identifiants contre les interceptions.

- Renforcer la confiance des utilisateurs.

8.6. Sécurité des données transmises (fichiers, requêtes)

Toutes les interactions utilisateurs passent par des requêtes paramétrées pour prévenir les injections SQL :

```
$stmt = $pdo->prepare("SELECT * FROM User WHERE Username = ?");  
$stmt->execute([$username]);
```

Les fichiers photo uploadés ou archivés sont aussi vérifiés côté serveur pour éviter toute tentative de fichier malveillant.

8.7. Journalisation des actions critiques

Pour chaque action importante (connexion, suppression de photo, changement de mot de passe...), un log est enregistré dans la base :

```
CREATE TABLE Log (  
    ID_Log INTEGER PRIMARY KEY AUTOINCREMENT,  
    ID_User INTEGER,  
    Action TEXT,  
    Date DATETIME,  
    Gravite TEXT  
);
```

Cela permet de suivre l'historique complet des actions.

8.8. Redirections contrôlées

Après chaque action, une redirection automatique est effectuée vers une page contrôlée comme menu.php. Cela évite les rechargements multiples et garde une cohérence de navigation.

8.9. Accès aux fichiers sécurisés









Les dossiers comme /uploads/ sont protégés pour empêcher l'indexation publique, via un fichier .htaccess et un contrôle des accès via PHP.

Conclusion













La navigation sécurisée entre les pages est le socle de confiance de toute application web. Dans notre projet, nous avons appliqué une stratégie complète combinant sessions, rôles, logs, redirections, et contrôle d'accès strict.

Cela garantit une expérience utilisateur fluide, mais surtout une sécurité rigoureuse, indispensable dès lors qu'on gère des données utilisateurs, des fichiers ou des capteurs connectés.












9. Gestion GitHub

 Semestre_1	Delete Semestre_1/Planning.xlsx	last month
 Semestre_2	Add files via upload	yesterday
 dorian	Update README	last month
 mattéo	Update readme	last month
 océan	Update README	5 months ago
 thibault	Add files via upload	3 weeks ago
 Planning.xlsx	Add files via upload	2 weeks ago
 README.md	Update README.md	2 weeks ago

S1

 ..		
 Algorithme_python.drawio	Add files via upload	2 months ago
 Base de données.odt	Add files via upload	2 months ago
 Cahier Des Charges.odt	Add files via upload	2 months ago
 Diagramme algo PICO.drawio	Add files via upload	2 months ago
 Diagramme algo raspberry.drawio	Add files via upload	2 months ago
 La Maquette du Diagramme.drawio	Add files via upload	2 months ago
 La Maquette du Site Web.png	Add files via upload	2 months ago
 Les risques.pdf	Add files via upload	last month
 Projet en anglais	Create Projet en anglais	2 months ago
 RACI S1.xlsx	Rename RACI-S1.xlsx to RACI S1.xlsx	2 months ago
 Thales02_SAE15_rapport_S1.docx	Add files via upload	2 months ago

S2

 ..		
 Algorithme	Delete Semestre_2/Algorithme/test pour algo	3 days ago
 Base de données	Add files via upload	2 months ago
 Rapport Final	Create test pour rapport final	3 days ago
 Site Web	Delete Semestre_2/Site Web/mon_site_structured2.zip	last week
 Arborescence du Rapport de Validation-Vérification.png	Add files via upload	last month
 Base de données V2.drawio	Add files via upload	2 months ago
 RACI S2.xlsx	Add files via upload	last month
 Rapport de Validation-Vérification V2 - Modifié.docx	Add files via upload	yesterday
 Rapport de Validation-Vérification.docx	Rename Rapport de Validation-Vérification (Test)(1).docx to Rapport d...	2 months ago
 prendre_photo.py	Add files via upload	3 weeks ago

Organisation du dépôt GitHub du projet Thales

L'organisation du dépôt GitHub a été pensée pour garantir **clarté, accessibilité et traçabilité** tout au long des deux semestres du projet. Il reflète fidèlement la répartition du travail, la progression par étapes, et la séparation logique des livrables et scripts.

Arborescence principale

- **Semestre_1**
Dossier contenant tous les éléments de la phase S1 : cahier des charges, premières maquettes, scripts initiaux et fichiers de conception.
- **Semestre_2**
Répertoire dédié à la phase S2, avec les documents de validation, la base de données finale, les scripts MicroPython, ainsi que le site web complet.
- **dorian / matéo / océan / thibault**
Chaque membre de l'équipe dispose d'un dossier personnel pour y déposer ses travaux individuels, tests locaux ou fichiers intermédiaires. Cela permet de **tracer les contributions personnelles** et de suivre les évolutions de chacun.
- **Planning.xlsx**
Document de référence pour le suivi des tâches, les deadlines et la coordination entre membres.
- **README.md**
Fichier de présentation du projet, régulièrement mis à jour, résumant les objectifs, la structure, et le contenu du dépôt.

Contenu du dossier S1

On retrouve les livrables produits au premier semestre :

- Diagrammes de conception (UML, architecture matérielle)
- Algorithmes Python de départ
- Maquettes du site web
- Premier cahier des charges et fiche RACI
- Présentation en anglais
- Rapport S1 (Thales1_SAE23_rapport_S1.docx)

Ce dossier témoigne de la **phase de cadrage du projet**, avec un effort important sur la préparation, la compréhension des besoins et la planification.

Contenu du dossier S2

Ce dossier regroupe l'ensemble des **fichiers finaux et techniques** :

- Scripts MicroPython (`main.py`, `testled.py`)
- Base de données SQLite complète
- Rapport de validation (Vérification/Validation)
- Site web complet zippé (`structureV2_site_web.zip`)
- Documents finaux : rapport S2, fiches de tests, fiches d'anomalies
- README spécifique au semestre 2

Les livrables y sont rangés de manière rigoureuse, ce qui facilite le **rejeu du projet**, les démonstrations ou les tests indépendants.

Points forts de l'organisation

- **Lisibilité** : noms clairs, pas de surcharge de fichiers racine.
- **Versionnement logique** : S1 vs S2.
- **Traçabilité des membres** via dossiers nominatifs.
- **Centralisation des livrables clés** dans chaque dossier.

10. Gestion de Projet

S1

	Dorian	Thibault	Mattéo	Océan
Planning	A	R	C	R
Exigences	R	R	R	A
Risques	C	R	A	C
RACI	A	C	R	R
Algorithme python	A	R	R	I
Présentation projet(anglais)	R	I	A	R
Maquettes	R	I	I	A
Base de données	R	C	A	R
Github	R	A	C	C
Expérience individuel	R	R	R	R

S2

RACI				
R= Réalisateur				
C=Consultant				
I=informer				
A=approbateur				
	Dorian	Thibault	Mattéo	Océan
RACI	C	I	A	R
Base de données	R	C	R	A
Planning	R	C	I	A
Liaison Pico et Python	A	R	C	I
Site Web	R	A	R	C
Risques	A	C	I	R
Plan de test	R	A	C	R
Fiche d'anomalies	R	C	A	R
Schéma électrique	C	R	R	A
Configuration Raspberry	R	R	C	A

RACI : Répartition des rôles et responsabilités dans le projet

Dans le cadre de notre projet, nous avons adopté le modèle RACI pour organiser et structurer la répartition des responsabilités au sein de notre équipe composée de quatre membres : Dorian, Thibault, Mattéo et Océan. Ce modèle permet de clarifier qui est Responsable (R), Approbateur (A), Consulté (C) et Informé (I) pour chaque tâche critique du projet. L'objectif est d'assurer une bonne collaboration, une répartition équilibrée des charges et une traçabilité claire des décisions et des contributions de chacun.

Dorian:

Dorian a été responsable de la réalisation directe des tâches suivantes : Base de données , Planning, Site Web, Plan de test, Fiche d'anomalies, Configuration Raspberry . Être responsable signifie qu'il/elle a pris en charge l'exécution complète de ces livrables, du développement à la mise en place concrète, en assurant leur bon fonctionnement.

Dorian a également assumé un rôle d'approbateur pour : Liaison Pico et Python, Risques. En tant qu'approbateur, Dorian a validé les livrables finaux, s'est assuré de leur conformité aux attentes et a parfois tranché sur les choix techniques ou organisationnels.

En tant que consultant, Dorian a été sollicité pour donner son avis ou partager son expertise sur : Schéma électrique. Ce rôle a permis d'enrichir les décisions grâce à un regard extérieur ou complémentaire.

Thibault:

Thibault a été responsable de la réalisation directe des tâches suivantes : Liaison Pico et Python, Schéma électrique, Configuration Raspberry . Être responsable signifie qu'il/elle a pris en charge l'exécution complète de ces livrables, du développement à la mise en place concrète, en assurant leur bon fonctionnement.

Thibault a également assumé un rôle d'approbateur pour : Site Web, Plan de test. En tant qu'approbateur, Thibault a validé les livrables finaux, s'est assuré de leur conformité aux attentes et a parfois tranché sur les choix techniques ou organisationnels.

En tant que consultant, Thibault a été sollicité pour donner son avis ou partager son expertise sur : Base de données , Planning, Risques, Fiche d'anomalies. Ce rôle a permis d'enrichir les décisions grâce à un regard extérieur ou complémentaire.

Mattéo:

Mattéo a été responsable de la réalisation directe des tâches suivantes : Base de données , Site Web, Schéma électrique. Être responsable signifie qu'il/elle a pris en charge l'exécution complète de ces livrables, du développement à la mise en place concrète, en assurant leur bon fonctionnement.

Mattéo a également assumé un rôle d'approbateur pour : Fiche d'anomalies. En tant qu'approbateur, Mattéo a validé les livrables finaux, s'est assuré de leur conformité aux attentes et a parfois tranché sur les choix techniques ou organisationnels.

En tant que consultant, Mattéo a été sollicité pour donner son avis ou partager son expertise sur : Liaison Pico et Python, Plan de test, Configuration Raspberry . Ce rôle a permis d'enrichir les décisions grâce à un regard extérieur ou complémentaire.

Enfin, Mattéo a été informé des avancées sur : Planning, Risques, ce qui lui a permis de rester au courant du déroulement du projet, sans y participer directement.

Océan:

Océan a été responsable de la réalisation directe des tâches suivantes : Risques, Plan de test, Fiche d'anomalies. Être responsable signifie qu'il/elle a pris en charge l'exécution complète de ces livrables, du développement à la mise en place concrète, en assurant leur bon fonctionnement.

Océan a également assumé un rôle d'approbateur pour : Base de données , Planning, Schéma électrique, Configuration Raspberry . En tant qu'approbateur, Océan a validé les livrables finaux, s'est assuré de leur conformité aux attentes et a parfois tranché sur les choix techniques ou organisationnels.

En tant que consultant, Océan a été sollicité pour donner son avis ou partager son expertise sur : Site Web. Ce rôle a permis d'enrichir les décisions grâce à un regard extérieur ou complémentaire.

Enfin, Océan a été informé des avancées sur : Liaison Pico et Python, ce qui lui a permis de rester au courant du déroulement du projet, sans y participer directement.

Planning durant le projet :

Numéro	Titre de la Tâche	Responsable Tâche	Date Début	Date Limite	Durée (jour)	Status
Projet			01/10/24	19/06/25	188	
1,0	Appropriation du projet		01/10/24	12/10/24	9	
1,1	Brainstorming de départ	Groupe	01/10/24	10/10/24	8	Terminé
1,2	Compréhension de la problématique	Groupe	01/10/24	12/10/24	9	Terminé
1,4	GitHub	Groupe	10/10/24	29/10/24	14	Terminé
2,0	Définition et planification du projet (Semestre 1)		16/10/24	15/12/24	43	
2,1	Planning	Groupe	16/10/24	15/12/24	43	Terminé
2,2	Exigences	Groupe	30/10/24	05/11/24	5	Terminé
2,3	Raci	Groupe	20/11/24	01/12/24	8	Terminé
2,4	Analyse des risques	Groupe	20/11/24	03/12/24	10	Terminé
2,5	Présentation et objectifs du projet en anglais	Groupe	26/11/24	08/12/24	9	Terminé
2,7	Algorithme du programme python	Groupe	30/11/24	15/12/24	10	Terminé
2,8	Schéma de la Base de Données	Groupe	27/11/24	14/12/24	13	Terminé
2,9	Maquette du site web	Groupe	12/11/24	29/11/24	14	Terminé
2,A	Expérience individuel	Groupe	01/10/24	20/12/24	59	Terminé
3,0	Réalisation du projet (Semestre 2)		16/03/25	19/06/25	95	
3,1	Schéma électrique du montage	Groupe	06/05/25	20/05/25	11	Terminé
3,2	Conception des échanges entre pages du site web	Groupe	22/03/25	30/03/25	5	Terminé
3,3	Configuration de la Raspberry (système + applications + configuration)	Groupe	21/03/25	26/03/25	5	Terminé
3,4	Mise a jour de la base de donnée	Groupe	21/03/25	14/04/25	17	Terminé
3,5	Liaison Pico/pi	Groupe	29/03/25	11/05/25	30	Terminé
3,6	Planning (semestre 2)	Groupe	16/03/25	22/03/25	5	Terminé
3,7	Réalisation du RACI (semestre 2)	Groupe	16/03/25	21/03/25	5	Terminé
3,8	Canevas fiche d'anomalie	Groupe	07/04/25	28/04/25	16	Terminé
3,9	Rapport de Validation/ Vérification	Groupe	16/04/25	27/05/25	11	Terminé
3,A	Rapport Final du Projet	Groupe	20/05/25	11/06/25	21	Terminé

Analyse et Description du Planning du Projet

Le planning du projet a été conçu comme un outil central de coordination, permettant à l'équipe de structurer et d'échelonner les différentes tâches du semestre en tenant compte des contraintes pédagogiques, des compétences de chacun, et des jalons importants à respecter. Ce document, planifié dès le début du projet, a joué un rôle fondamental dans la réussite globale en assurant une visibilité claire sur l'avancement des livrables et la répartition des responsabilités.

1. Structure générale du planning

Le planning a été articulé autour de grandes phases de travail correspondant aux moments clés du projet :

- Appropriation du sujet
- Planification stratégique
- Réalisation technique (web, capteurs, caméra, base de données)
- Tests, validation, documentation et soutenance

Chaque tâche a été numérotée de manière hiérarchique pour indiquer sa position dans le projet global (exemple : 1.1 pour une sous-tâche de la tâche 1). La durée prévue, le responsable et le statut sont également précisés, ce qui permet un suivi précis semaine après semaine.

2. Logique temporelle et séquençement

Le planning commence avec une phase d'appropriation du projet courant octobre 2024. Le brainstorming et la compréhension de la problématique ont permis à l'équipe de définir les objectifs clairs, en lien avec le cahier des charges. Cette base solide a ensuite débouché sur la création du dépôt GitHub, et une organisation de l'équipe autour des rôles définis dans le RACI.

À partir de mi-octobre, la planification stratégique du projet est lancée avec des dates limites réalistes et réparties sur plusieurs semaines. Le choix de prévoir de longues plages pour certaines tâches (ex. : planification, développement) traduit une volonté de laisser de la flexibilité pour les ajustements et les itérations.

3. Adaptabilité et robustesse du planning

Le planning montre une capacité d'adaptation grâce à la présence de jalons intermédiaires (réunions de suivi, étapes de validation) qui permettent d'ajuster la suite en fonction de l'avancement réel.

Les tâches sont planifiées de manière réaliste, sans chevauchement excessif, ce qui évite la surcharge de travail et permet une spécialisation des membres sur leurs domaines respectifs (web, base de données, capteurs, tests). Les retards éventuels sont anticipés par des marges de sécurité entre certaines tâches.

4. Répartition et coordination

La répartition des tâches se fait majoritairement en groupe au début du projet, ce qui permet de poser les bases communes (brainstorming, compréhension du sujet, GitHub, maquette). Ensuite, chaque membre prend la responsabilité de modules spécifiques selon ses compétences. Cette stratégie permet à la fois l'implication de tous dans la phase initiale, et une efficacité maximale dans la réalisation technique.

Conclusion

Ce planning, bien que perfectible comme tout outil de prévision, a été pensé de manière structurée, évolutive et réaliste. Il a permis à l'équipe de respecter les délais critiques, de répartir équitablement la charge de travail, et de maintenir une cohérence globale dans la conduite du projet. Son utilité a été essentielle dans la réussite de cette SAE, tant sur le plan technique que méthodologique.

11. Retour d'Expérience Individuel

Chaque membre de l'équipe exprime ici son retour d'expérience :

Dorian :

Implication Semestre 1:

Interface du Github terminé, (03/11)

Travail sur la maquette et un diagramme de la maquette terminé, (15/12)

base de données terminé, (21/12)

projet en anglais terminé, (18/12)

Cahier des charges, (08/01)

Rapport, (12/01)

Implication Semestre 2:

Planification

RACI

Site web en php, html, css

Base de données en sqlite

Arborescence des plans de test/validation

Retour expérience individuel pour la S1 :

Projet intéressant et complet pour le moment, quelques difficultés avec les membres du groupe en termes de coordination de celui-ci,

Mais notre groupe arrive quand même à suivre le rythme, le déroulé du projet avec chaque tâche à accomplir dans un temps imparti est captivant.

Retour expérience individuel pour la S2 :

La fin de cette expérience est proche, il reste beaucoup de choses à finaliser ce qui est je pense la chose la plus dure de ce semestre 2.

Ce qui a été intéressant est l'évolution et l'investissement de chacun du groupe, que ce soient des points faibles comme des forts.

En tant que chef de projet j'ai un peu touché à tout en donnant l'aide nécessaire.

Mattéo :

Semestre 1:

Futur rapport, travail sur les exigences (cahier des charges).

Réalisateur du R.A.C.I, du cahier des charges et de l'algorithme python.

Approbateur pour les risques et la bases de données.

Consultant pour le projet en anglais et planning.

Informé pour les maquettes.

Semestre 2:

Réalisation de la base de données codée en SQLite.

Réalisation du site web en php, html et css.

Liaison entre la base de données et le site web.

Dans l'ensemble le projet est assez intéressant c'est à dire de comprendre comment organiser la structuration du projet.

Le fait de mener à bien la réalisation des différentes tâches pour en voir le résultat à la fin est très réjouissant.

Océan :

travail sur le planning, exigences et RACI

travail sur github

recherche sur la partie en anglais et exigences, base de données

travail sur base de donnée et projet anglais

travail rapport de test/validation

travail rapport final

j'ai aimé travailler sur le projet, il est très intéressant.

j'aime les travaux de groupe mais au début c'était difficile de se répartir les différentes tâches cependant je trouve qu'on a bien travaillé.

Thibault :

Semestre 1 :

Implication en tant que :

Réalisateur dans la réalisation du planning, du cahier des charges (exigences), des risques et de l'algorithme python.

Consulté pour la structuration de la base de données et pour l'attribution des rôles dans le RACI.

Informé de l'avancement de la maquette du site web et de la présentation du projet en anglais.

Retour d'expérience personnelle :

Cette expérience m'a permis de mieux visualiser les différentes tâches à réaliser en amont dans un projet,

c'est à dire toute la partie gestion de projet, mais également les éléments qui seront essentiels à l'aboutissement du projet par la suite.

Semestre 2 :

Installation de l'OS sur le Raspberry PI

Ouverture de l'accès au Raspberry en connexion SSH depuis l'extérieur

Etablissement de la communication entre le Raspberry Pi et le Pico avant la prise de photo via la webcam