

Caméra Timelapse

Présentation technique du projet

THIEFFRY Raphaël, HUA Olivier, RIBETTE Baptiste, PUYAUBREAU Ethan et
RUSSAC Antonin

Polytech Paris-Saclay
Université Paris-Saclay

April 27, 2025



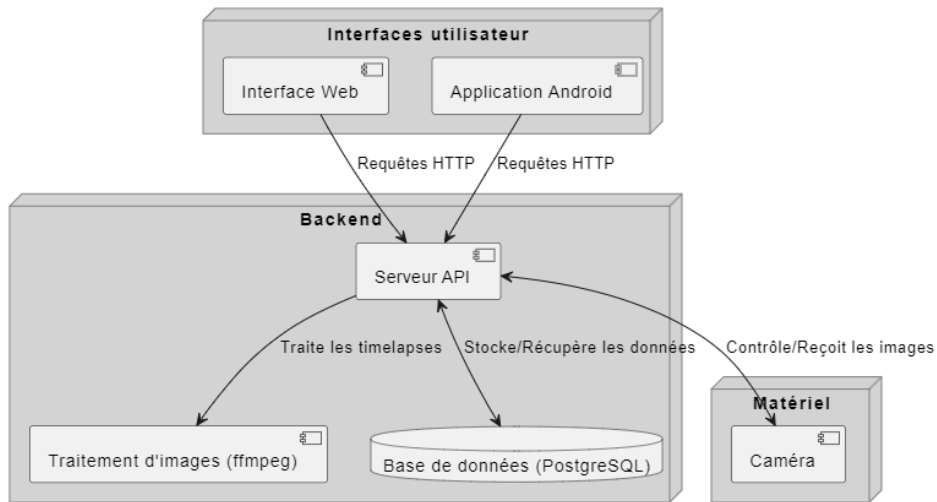
Plan

- ① Introduction
- ② Exemple UML
- ③ Cahier des charges
- ④ Partie mécanique
- ⑤ Partie électronique
- ⑥ Serveur, Front et Application mobile
 - Front end
- ⑦ Démonstration
- ⑧ Conclusion

Contexte et Objectifs

- Présentation du projet : Caméra Timelapse
- Contexte d'utilisation et motivation
- Objectifs principaux du projet

Exemple UML



Spécifications et Contraintes

- Fonctionnalités attendues
- Contraintes techniques (matériel, logiciel)
- Livrables

Partie mécanique

- Présentation de la structure mécanique
- Choix des matériaux et conception
- Schémas ou photos du montage

Partie électronique

- Architecture électronique
- Composants utilisés (capteurs, microcontrôleurs, etc.)
- Schémas électroniques

Serveur, Front et Application mobile

- Présentation de l'architecture logicielle
- Fonctionnalités principales du serveur
- Aperçu du front web et de l'application mobile
- Technologies utilisées

Front end

Choix des technologies :

- HTML, CSS, Javascript
- ChartJS pour la visualisation de données
- Ajax pour la communication asynchrone

Efforts notables :

- Carousel 3D animé, modulable selon le nombre de projets
- Responsive : adaptation à la taille de la fenêtre
- Menu déroulant pour cacher les images en excès
- Affichage d'images compressées pour optimiser la génération
- Style universel pour le confort utilisateur

Front end : Architecture

Organisation du code :

- **HTML** : `index.html`, `pages/projet_detail.html`
- **CSS** : `css/base/style.css`, `css/components/style_menu.css`, `css/pages/style_projet.css`, images dans `css/image/`
- **JS** : `js/core/` (logique principale), `js/libs/` (bibliothèques tierces), `js/pages/` (scripts spécifiques aux pages)

Outils et automatisation :

- Déploiement automatisé avec `.gitea/workflows/deploy.yml`
- Conteneurisation possible via `docker-compose.yml`
- Fichier `deploy.sh` pour le déploiement manuel

Automatisation du déploiement : deploy.sh

Script de déploiement automatique :

- Un script bash (`deploy.sh`) a été développé par Ethan pour automatiser le déploiement du front-end.
- Il permet de mettre à jour rapidement la version en ligne lors du développement continu.
- Ce script facilite l'intégration des modifications et réduit les risques d'erreur humaine lors des mises à jour.
- Il s'intègre dans la chaîne de développement pour accélérer les cycles de test et de validation.

Front end (suite)

Difficultés rencontrées :

- Interface optimale entre le front end, le serveur et l'application mobile
- Évolution constante des routes et de la logique serveur
- Adaptation aux changements du modèle de données

Apprentissages :

- Génération de 3D en HTML et CSS
- Manipulation d'images et de vidéos

Démonstration

- Vidéo ou démonstration en direct du projet
- Points forts à montrer

Bilan et perspectives

Résumé

- Synthèse des apports du projet
- Retour d'expérience

Perspectives

- Améliorations possibles
- Suites envisagées

Remerciements

Merci pour votre attention !