



# Plan de projet

Projet de conception

Architecture Logiciel

1.0

Jérôme Blackburn Saucier (BLAJ01019200)

Samuel Harvey (HARS04119907)

Aicha Kaba (KABA14569805)

27-11-2022

## Historique des révisions

Date	Version	Description	Auteur
20-11-2022	1.0	Séparation des tâches, début de la mise en page, planification et calendrier des rencontres	Jérôme B. Saucier
22-11-2022	1.1	Objectifs	
		Ressource matérielle	
25-11-2022	2.2	Stratégie de contrôle des versions	Aicha Kaba
			Aicha kaba

## Définitions

Terme	Définition

## Abréviations/acronymes

Abré./Acro.	Définition

## Table des matières

Historique des révisions .....	2
Définitions .....	2
Abréviations/acronymes .....	2
Table des matières .....	3
1. Introduction .....	4
1.1. Objectifs .....	4
1.2. Références .....	4
1.3. Portée.....	5
1.4. Livrables .....	5
2. Organisation du projet .....	6
2.1. Composition de l'équipe du projet .....	6
2.2. Stratégie de contrôle des versions.....	6
2.3. Directives de livraison .....	6
3. Ressources matérielles, calendrier et budget.....	7
3.1. Ressources matérielles .....	7
3.2. Calendrier des tâches du projet.....	7
3.3. Budget .....	7
4. Gestion des risques .....	8
Annexe A – Calendrier d'équipe .....	9
Annexe B – Diagramme de Gantt.....	11

## 1. Introduction

### 1.1. Objectifs

Dans le cadre du cours comme projet de conception, nous allons mettre en place un logiciel une caméra de surveillance capable de capter les mouvements d'un individu et d'envoyer une alerte par courriel à la personne responsable du système. La caméra devrait être capable de donner les images en temps réel et avoir une numérisation en continu pour identifier les mouvements en négligeant les petits mouvements.

Le but de ce document est de fournir une description complète des exigences logicielles. Afin d'illustrer ces interactions, ce document va contenir les exigences fonctionnelles et non fonctionnelles ainsi que les cas d'utilisation.

### 1.2. Références

**Twilio** : est un outil qui permet aux développeurs de logiciels d'envoyer et de recevoir des messages texte par programmation. Dans le cadre de notre projet, il permettra d'envoyer un message d'alerte via un agent.

Dans notre cas nous avons proposé les deux cas par courriel et par message texte, mais c'est à confirmer avec le client.

**Numpy** : est une bibliothèque de python qui comporte des fonctions permettant de manipuler des matrices ou tableaux multidimensionnels, qui nous permet de faciliter les calculs dans le traitement d'image.

**Tkinter**: Python offre plusieurs options pour développer une interface utilisateur graphique (GUI). De toutes les méthodes GUI. Tkinter est la méthode la plus couramment utilisée. Il s'agit d'une interface python standard pour la boîte à outils Tk GUI livré avec python. Python avec tkinter est le moyen le plus rapide et le plus simple de créer les applications GUI.

**Scipy**: est une bibliothèque d'algorithmes optimisés. Cette librairie permet de posséder notamment une fonction de convolution 2D, ce qui faciliterait grandement le développement d'un filtre gaussien

**Urllib** : est le module de gestion d'URL pour python. Il est utilisé pour récupérer des URL (Uniform Resource Locators). Il utilise la fonction URL open et est capable de récupérer les URL à l'aide d'une variété de protocoles différents.

### 1.3. Portée

Dans le cadre du projet, le programme devant être remis doit être en mesure de faire l'acquisition d'images provenant d'une caméra IP, de traiter l'image afin de pouvoir ensuite y insérer un algorithme de détection de mouvement pour finalement envoyer des notifications à un utilisateur lorsque des mouvements sont détectés. Les limites du projet sont donc constituées d'un côté par la détection de mouvement d'une caméra IP et de l'autre vers l'envoi d'une simple notification sous forme de courriel ou de SMS. Le projet a un lien avec le laboratoire 3 de ce cours puisque nous devons avoir encore une fois une interface graphique créée sous Python, comme c'était le cas dans le laboratoire 3 où nous en avons créé une à l'aide de la librairie Tkinter.

### 1.4. Livrables

Tout d'abord, les documents qui devront être remis dans le cadre du projet sont les suivants :

- Plan de projet
- Spécifications des exigences
- Description des composants
- Architecture et conception du logiciel

Ensuite, en ce qui concerne les livrables du projet lui-même, nous devons inclure un projet étant composé des éléments suivants :

- Acquisition d'images d'une caméra IP
- Traitement des images
- Génération d'alertes lorsque détection de mouvements
- Interface utilisateur

## 2. Organisation du projet

### 2.1. Composition de l'équipe du projet

Aicha Développeur, Gestionnaire de Projet

Responsable de la communication avec le client et s'assure de la qualité des rapports

Jérôme Développeur, Scrum Master

Responsable de la gestion des livrables et doit aussi s'assurer de la validité des « user stories ».

Samuel Développeur Prioritaire

Responsable du code, de la gestion des sources et responsable de la sécurité.

Jérémy, Product owner

### 2.2. Stratégie de contrôle des versions

Pour gérer les différents documents et de garder un historique des versions de notre projet, nous avons pensé à l'utilisation du gestionnaire de version Git. Nous utiliserons le site GitHub pour héberger notre répertoire de projet. Ensuite, nous avons mis en place un Trello afin de partager les tâches et suivre efficacement notre évolution.

GitHub URL: <https://github.com/samm56784/Projet6GEI311>

### 2.3. Directives de livraison

Le projet doit contenir les éléments suivants :

- Code source Python (fichier .py)
- Fichier pip à installer inclus dans "requirement.txt"
- Configuration de git "git ignore"
- Script d'initialisation de l'environnement tel que "setup.bat"
- Fichier d'instruction "readme.txt"

Les instructions d'installation, configuration et exécutions seront fournies. Les documents du projet expliqueront le fonctionnement du logiciel, son architecture (structure, dépendances, algorithme...) et les résultats obtenus seront fournis.

### 3. Ressources matérielles, calendrier et budget

#### 3.1. Ressources matérielles

Au niveau du matériel utilisé, un ordinateur pour exécuter le logiciel est nécessaire. De plus, il y a une caméra de surveillance qui est connectée sur le réseau local de l'UQAC que l'on peut accéder grâce à l'ordinateur.

#### 3.2. Calendrier des tâches du projet

Voir Annexe B pour le diagramme de Gantt représentant les étapes à suivre du projet.

Le développement se fera de manière agile. Pour suivre le backlog, il suffit d'avoir un compte Trello et de se rendre sur le board suivant :

<https://trello.com/invite/b/mWHhaN7G/ATTI34dd5349a4b56500316165f71bb99c0d35CDE1BC/tableau-agile>

#### 3.3. Budget

Voici comment sera évalué notre budget :

- Dans le cadre de ce projet, nous estimons que nous allons mettre en moyenne 10 heures par personne par semaine.
- Puisque nous avons une période de cinq semaines pour effectuer le projet, nous estimons notre total d'heures à 50 heures par personne et donc 150 heures en comptant toutes les heures mises sur le projet.
- Nos employés sont payés à un taux horaire de 52,82\$, soit le salaire médian au Québec pour un ingénieur informatique<sup>1</sup>.

Au final, le budget alloué au projet sera de 150 heures \* 52,82\$/h, soit 7 923\$.

Salaire ingénieur : salaire médian ingénieur informatique Québec,

<https://www.guichetemplois.gc.ca/rapportmarche/salaire-profession/2866/ca>

---

<sup>1</sup> <https://www.guichetemplois.gc.ca/rapportmarche/salaire-profession/2866/ca>

## 4. Gestion des risques

Risques	Description	Mitigation
Risque de se faire détourner les messages	Quelqu'un de mal intentionné pourrait récupérer les informations transmises par messages textes.	Assure que seul l'agent a accès au système. Mettre une alerte de sécurité complète.
Risque de dévoiler à des gens mal intentionnés les informations sur le lieu capturé par la caméra.	Des personnes pourraient intercepter le « feed » de la caméra ou utiliser l'application elle-même afin d'obtenir des informations dans le but de faire du mal.	Mettre un code de sécurité sur la caméra pour assurer que seulement les personnes concernées puissent accéder au « feed » de la caméra.
Risque d'envoyer des données erronées au client	Détection trop facile à déclencher, alerte constamment le client de la présence de mouvements sur sa caméra.	À chaque sprint il faut faire les tests pour assurer que tout fonctionne bien pour éviter les erreurs d'envoi.
Risque que la caméra ne fonctionne plus.	La caméra elle-même pourrait avoir un problème, rendant l'application inutile.	Vérifier la caméra avant le test pour s'assurer que tout fonctionne bien.

### Norme et code standards

**ISO 29110** : définis les processus de cycle de vie dans son ensemble des profils standardisés et des rapports techniques peuvent être utilisés par un TPE lors du développement, de l'acquisition et de l'utilisation, ainsi que lors de la création et de la fourniture de systèmes comportant des éléments matériels et logiciels. Ils peuvent être appliqués à n'importe quel niveau dans le développement d'un système.

Source: [ISO/CEI 29110 — Wikipédia \(wikipedia.org\)](https://fr.wikipedia.org/wiki/ISO/CEI_29110)

**Simple Courriel Transfer Protocol**: est un protocole de communication standard internet pour la transmission de courrier électronique. Les serveurs de messagerie et autres agents de transfert de message utilisent SMTP pour envoyer et recevoir des messages électroniques.

Source: [Protocole de transfert de courrier simple - FRREF](https://fr.wikipedia.org/wiki/Protocole_de_transfert_de_courriel_simple)

**Modèle OSI**: est un cadre conceptuel qui définit comment les systèmes réseau communiquent et envoient des données d'un expéditeur à un destinataire.



Le modèle est utilisé pour décrire chaque composant de la communication de données pour établir des règles et des normes pour les applications et l'infrastructure du réseau.

Source : [Qu'est-ce que le modèle OSI : définition, couches, et plus | Proofpoint FR](#)

**JPEG:** est une norme qui définit le format d'enregistrement et l'algorithme de décodage pour une représentation numérique compressée d'une image fixe.

Source : [JPEG — Wikipédia \(wikipedia.org\)](#)

## Annexe A – Calendrier d'équipe

### Novembre

L	M	M	J	V	S	D
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20 Première Rencontre
21	22	23 Labo 8-11, Rencontre présentielle	24	25 Rencontre 8h am	26	27 Rencontre Hebdomadaire 19h
28	29	30 Labo 8-11, Rencontre présentielle				

## Décembre

L	M	M	J	V	S	D
			1	2 Rencontre 8h	3	4 Rencontre hebdomadaire 19h
5	6	7 Labo 8-11, Rencontre présentielle	8	9 Rencontre 8h	10	11 Rencontre Hebdomadaire 19h
12	13	14	15	16	17	18 Rencontre Hebdomadaire 19h
19	20	21	22	23	24	25
26	27	28	29	30	31	

## Annexe B – Diagramme de Gantt

