**Projet PILS : Cahier des charges**

BOURAOUI Amir

SONG Lewei

Table des matières

[Introduction 3](#_Toc195823070)

[Contexte : 3](#_Toc195823071)

[Objectifs du projet : 3](#_Toc195823072)

[Description du problème 3](#_Toc195823073)

[Exigences 4](#_Toc195823074)

[Exigences Fonctionnelles 4](#_Toc195823075)

[Exigences Non Fonctionnelles 4](#_Toc195823076)

[Environnement et Modèle 5](#_Toc195823077)

[Guide utilisateur 5](#_Toc195823078)

[Conclusion 6](#_Toc195823079)

# Introduction

## Contexte :

Dans un environnement où l'efficacité énergétique et la qualité de l'air intérieur sont devenues des enjeux majeurs, l'installation d'une VMC double flux se présente comme une solution innovante pour répondre à ces défis. Ce système assure une gestion optimisée de l'air en extrayant l'air vicié des pièces de service tout en insufflant simultanément de l'air neuf dans les pièces de vie. Le projet s’inscrit donc dans une démarche de modernisation des systèmes de ventilation, en offrant aux professionnels et aux particuliers un outil de planification et de modélisation précis pour l'installation d'une VMC double flux.

## Objectifs du projet :

L'objectif principal est de développer une application intuitive permettant de modéliser de manière précise la configuration d'une habitation (étages, murs, fenêtres, et bouches de VMC) et de simuler le bilan aéraulique associé. Ce logiciel devra intégrer une interface graphique détaillée et une vue textuelle synthétisant les débits et les caractéristiques des installations, tout en assurant une communication fluide entre les différentes vues via l'utilisation du bus Ivy. Le projet vise ainsi à fournir un outil d'aide à l'installation qui garantit un dimensionnement optimal et conforme aux exigences techniques de la VMC double flux.

# Description du problème

L'installation d'une VMC double flux présente plusieurs défis techniques et organisationnels. En effet, la réussite d'un tel projet dépend de la capacité à modéliser avec précision la configuration d'une habitation, en tenant compte des particularités architecturales (étages, murs, fenêtres) et de la localisation optimale des bouches de ventilation. Chaque élément doit être dimensionné correctement pour assurer une extraction efficace de l'air vicié et une insufflation adéquate d'air neuf, tout en respectant les contraintes structurelles et les exigences énergétiques.

Le problème se complique par la nécessité de coordonner les différents paramètres aérauliques et de garantir une communication fluide entre les divers modules de l'application, notamment via l'utilisation du bus Ivy. Ainsi, il est essentiel de développer un outil intuitif qui permette non seulement de visualiser de manière graphique et textuelle l'ensemble des installations, mais aussi de simuler et d'optimiser le bilan aéraulique pour garantir des performances conformes aux normes en vigueur.

# Exigences

### Exigences Fonctionnelles

Ce que l’application doit permettre de faire :

Créer et gérer plusieurs étages

L’utilisateur peut ajouter un nouvel étage.

Chaque étage peut avoir un nom et une hauteur personnalisée.

Sélection d’un étage actif

L’utilisateur peut cliquer sur un étage pour le sélectionner et y dessiner.

Outils de dessin

Murs, portes, fenêtres : possibilité de les dessiner à la souris.

Gomme (outil pour effacer des objets).

Bouches de VMC : avec choix du rôle (extraction / insufflation / entrée / sortie).

Interface intuitive

Une barre d’outils en bas avec des icônes claires.

Une zone de dessin centrale.

Une liste d’étages à droite avec défilement (scroll).

Affichage de la hauteur de l’étage

En bas à droite du plan, la hauteur est indiquée (modifiable par l’utilisateur).

Aide visuelle (boussole)

Une petite boussole est affichée pour indiquer l’orientation (Nord, Est, Sud, Ouest).

### Exigences Non Fonctionnelles

Ce que l’application doit respecter en termes de qualité :

Interface simple et lisible

Couleurs douces, police lisible, éléments espacés.

Réactivité de l’interface

L’application réagit immédiatement aux clics ou changements d’outils.

Adaptabilité de la fenêtre

L’interface s’adapte automatiquement en cas de redimensionnement (ex : plein écran).

Bonne organisation du code

Code Python structuré en modules (view, controller, etc.), facile à lire et à maintenir.

Gestion des erreurs utilisateur

Messages d’alerte clairs (ex : si l’utilisateur oublie de choisir un type de bouche).

# Environnement et Modèle

**Langage principal :** Python 3.12

**Interface graphique :** Tkinter (bibliothèque standard de Python)

**Structure MVC :**

View : Interface graphique (GraphicalView)

Controller : Réception et gestion des actions utilisateur via ivy\_bus

Model : Données des étages, murs, objets dessinés, etc.

**Autres outils utilisés :**

ivy : Pour la communication entre les composants (bus d’événements léger)

ttk : Pour les widgets modernes (boutons, scrollbars, etc.)

simpledialog, messagebox : Pour les interactions utilisateur (saisie, alertes)

**Modèle de Données**

L’application repose sur un modèle simple :

**Étages (Floors)**

Nom (ex : “Floor 1”)

Hauteur (modifiable, par défaut 2.5m)

**Objets dessinés sur le Canvas**

Murs, fenêtres, portes, bouches de VMC

Chaque objet est lié à des métadonnées (coordonnées, type, infos techniques)

L’outil gomme peut supprimer ces objets individuellement

**Interactions utilisateur**

Chaque clic est intercepté et envoyé au controller via ivy\_bus

Le controller valide l’action et envoie un message de mise à jour à la view

# Guide utilisateur

1. Création des étages

Après avoir lancé `main.py`, l'utilisateur doit cliquer en haut à gauche pour créer un nouvel étage.

Le premier étage est automatiquement nommé "Floor 0".

Un clic droit sur un étage permet de renommer l'étage ou de définir sa hauteur.

2. Ajout de nouveaux étages

Chaque nouvel étage est ajouté au-dessus de l'étage sélectionné.

Actuellement, la suppression des étages n’est pas possible (fonctionnalité non prévue).

3. Outils de dessin

Sur chaque étage, l'utilisateur peut dessiner :

- Des murs (noir, par défaut)

- Des fenêtres (violet clair)

- Des portes (marron)

- Quatre types de gaines de ventilation (avec couleurs différentes)

Après avoir cliqué sur l’outil “ventilation”, un menu permet de choisir le type de gaine (couleur + fonction).

L’utilisateur doit ensuite renseigner le nom, le diamètre et le débit d’air pour chaque gaine (les dialogues s’affichent automatiquement).

4. Affichage des informations

Après la création d’une gaine, un survol avec la souris de plus d’1 seconde affiche les informations suivantes :

- Nom

- Diamètre

- Débit

- Fonction

5. Gomme

Après avoir sélectionné l’outil gomme, un simple clic sur un élément du dessin le supprime.

6. Sécurité d’interaction

Un simple clic sur le canvas sans avoir sélectionné d’outil ne déclenche aucune action.

7. Aide visuelle

En haut à gauche, une boussole et une règle indiquent les directions (N/E/S/O) et une échelle de 2 mètres.

8. Contrôle du canvas

Le canvas peut être agrandi en plein écran.

Après maximisation, il est possible de le faire glisser via les barres de défilement horizontale et verticale.

La molette de la souris sert à changer d’étage lorsqu’il y en a plusieurs.

# Conclusion

Ce projet nous a permis de découvrir et d’explorer l’utilisation du bus de communication Ivy dans une architecture logicielle.

Nous avons appris à mettre en place une communication fluide entre la vue et le contrôleur, en respectant une séparation claire des responsabilités.

En travaillant sur une interface graphique interactive, nous avons également enrichi nos compétences en programmation avec Tkinter, en gestion des événements, ainsi qu’en structuration d’un projet en architecture MVC.

Ce travail nous a offert une expérience complète, allant de la conception à l’implémentation, et nous a permis de mieux comprendre les défis du développement d’applications graphiques interactives.