

UNIVERSITÉ DE TOULON, I62 GÉNIE LOGICIEL

Rapport Projet I62

*Ivan CROS, Matteo BONAVITA, Clément MAYNADIER -
Groupe A1*

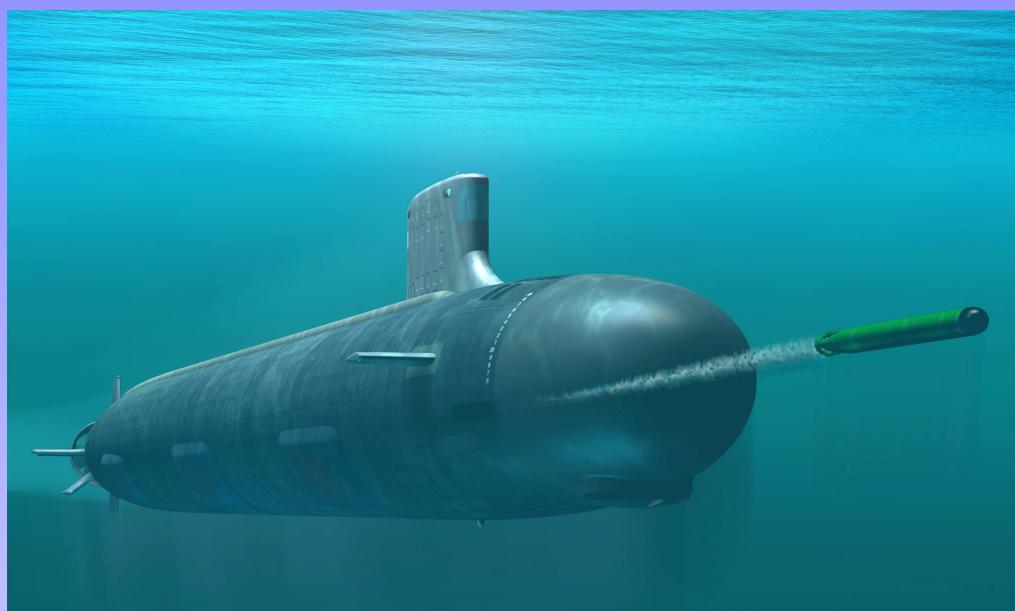


FIGURE 1 – Illustration d'un sous-marin d'attaque, réalisé par Ron Stern

Objectif du rapport

Afin de garantir la viabilité du projet avant d'investir dans des drones réels, notre client a fait l'acquisition d'un simulateur de drones. Le but est d'augmenter la vision des sous-marins d'attaque dans leurs environnement par l'intermédiaire de drones autonomes aériens. Le développement du produit nécessite une collaboration avec nos clients pour permettre notre équipe de s'initier au domaine naval. Pour que notre équipe puisse se familiariser au domaine naval, il est essentiel de travailler en étroite collaboration avec nos clients tout au long du développement du produit. L'objectif de ce rapport est de documenter le produit livrable selon l'analyse des besoins pour faciliter sa maintenance et son déploiement.

Table des matières

1 Présentation des membres du groupe	3
2 Exigences fonctionnelles (listes par catégorie)	3
2.1 Interface	3
2.2 Serveur	3
2.3 Editeur de situation (EdS)	4
2.4 Une situation	4
2.5 Simulation	4
2.6 OPFOR (forces ennemis)	5
2.7 Les unités	5
2.8 La vision	5
2.9 Le client	5
2.10 Les drones	5
2.11 Les sous-marins	5
3 Maquettage du produit	6
3.1 Serveur	6
3.2 Client	11
4 Conception UML	13
4.1 Axe fonctionnel	13
4.1.1 Diagramme de cas d'usage SI Serveur	13
4.1.2 Diagramme de cas d'usage SI Client	14
4.1.3 Structure du système niveau 1	15
4.1.4 Structure du système niveau 2	16
4.2 Axe statique	18
4.2.1 Diagramme des classes partie Serveur	18
4.2.2 Diagramme des classes partie Client	19

2 EXIGENCES FONCTIONNELLES (LISTES PAR CATÉGORIE)

4.3 Axe dynamique	20
4.3.1 Diagramme de séquence Serveur	20
4.3.2 Diagramme de séquence Client	21
4.3.3 Base de donnée : Modèle Entité-Association	22
5 Le manuel d'installation du logiciel	22
5.1 Installation des bibliothèques	22
5.2 Tkinter	22
5.3 Pillow	22
5.4 Comment procéder à l'exécution	22
6 Le manuel utilisateur du logiciel	23
6.1 Application Client	23
6.2 Application Serveur	23
7 Le planning mis en œuvre durant le projet	24
7.1 Sprint 1	24
7.1.1 Sprint Planning 27 février 2024	24
7.1.2 Daily Scrum du 12 Mars 2024	25
7.2 Sprint 2	25
7.2.1 Sprint planning 19 Mars 2024	25
7.2.2 Sprint Review du 26 mars 2024	25
7.2.3 Daily Scrum 8 Avril 2024	25
7.2.4 Sprint Review 16 Avril 2024	26
7.2.5 Sprint Retrospective 1 23 Avril 2024	26
7.2.6 Sprint Retrospective 2 29 Avril 2024	26
8 Diagramme de Gantt	27

1 Présentation des membres du groupe

2 Exigences fonctionnelles (listes par catégorie)

2.1 Interface

- Le Logiciel doit être séparé entre un serveur et des clients

2.2 Serveur

- Le serveur doit posséder un éditeur de situation (EdS).
- Le serveur doit simuler la situation
- Le serveur doit offrir une vue 'replay' des résultat clients

2 EXIGENCES FONCTIONNELLES (LISTES PAR CATÉGORIE)

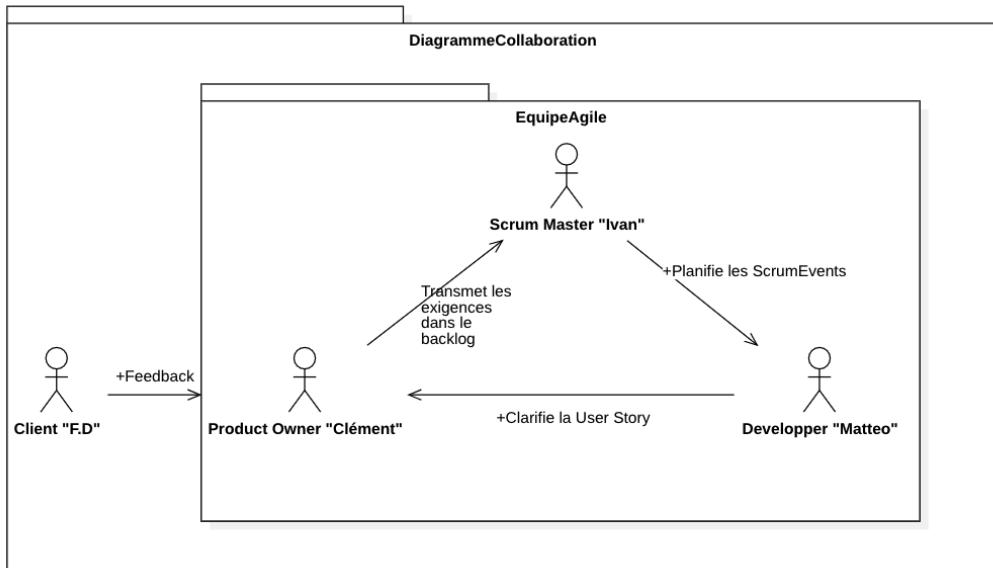


FIGURE 2 – Illustration de l'attribution des rôles principaux au sein de l'équipe.

2.3 Editeur de situation (EdS)

- L'EdS doit permettre de créer ou modifier une situation.
- L'EdS doit permettre d'ajouter des obstacles OPFOR¹ à une situation.
- L'EdS doit permettre de définir une carte topographique de la situation.
- L'EdS doit permettre de définir un ou plusieurs points de départ pour les sous-marins de la situation.
- L'EdS doit permettre de définir un ou plusieurs objectifs à la situation.

2.4 Une situation

- La situation doit pouvoir contenir des unités OPFOR.
- La situation doit contenir une carte topographique.
- La situation doit contenir un ou plusieurs points de départ pour les sous-marins de la situation.
- La situation doit contenir un ou plusieurs objectifs.

2.5 Simulation

Chaque objectif doit être assigné à un sous-marin uniquement.

- La simulation doit être jouée sur le serveur.
- La simulation doit avoir ses sous-marins contrôlés par les clients.

1. Toute force ennemie : drônes, navires

2.6 OPFOR (forces ennemis)

- L'OPFOR doit avoir des unités anti-aériennes.
- L'OPFOR doit avoir des unités navales capables de se déplacer durant la situation.
- L'OPFOR doit avoir des unités autres pour les objectifs de reconnaissance.

2.7 Les unités

- Les unités AA OPFOR doivent pouvoir détecter les drones BLUFOR².
- Les unités AA OPFOR doivent pouvoir détruire les drones BLUFOR.

2.8 La vision

- La vision replay doit offrir une vue BLUFOR de la situation.
- La vision replay doit offrir une vue omnisciente de la situation.

2.9 Le client

Concernant le client :

- Le client doit planifier une résolution de la situation pour son sous-marin.
- Le client doit gérer un sous-marin et sa flotte de drones.
- Le client doit pouvoir contrôler le sous-marin.
- Le client doit afficher ce que son sous-marin voit.
- Le client doit afficher ce que ses drones à portée voient.

2.10 Les drones

- Les drones doivent pouvoir patrouiller autour du sous-marin.
 - Les drones doivent pouvoir faire de la reconnaissance.
 - Les drones doivent pouvoir faire des missions de reconnaissance autonomes.
 - Les drones doivent pouvoir faire garde en autonomie.
 - Les drones doivent pouvoir être contrôlés par le client.
 - Les drones doivent envoyer des rapports de détection à la base de données.
- La base de données doit contenir des informations sur les drones.

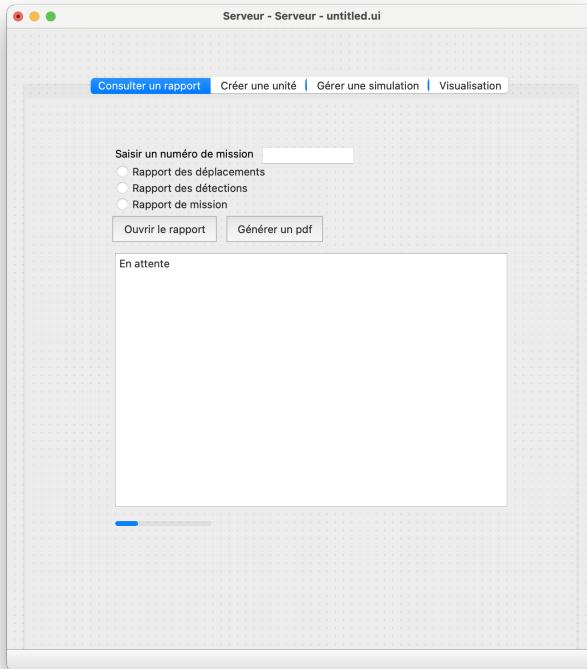
2.11 Les sous-marins

- Le sous-marin doit pouvoir changer de profondeur.
- Le sous-marin doit pouvoir communiquer avec ses drones.
- Le sous-marin doit pouvoir communiquer avec son commandement.

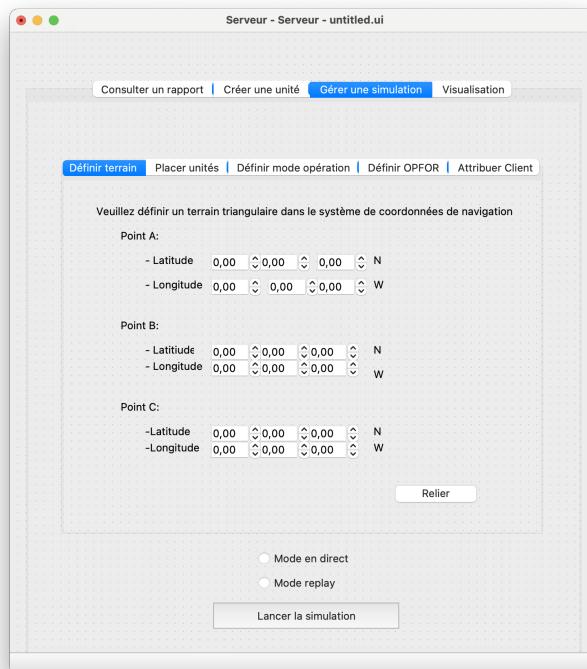
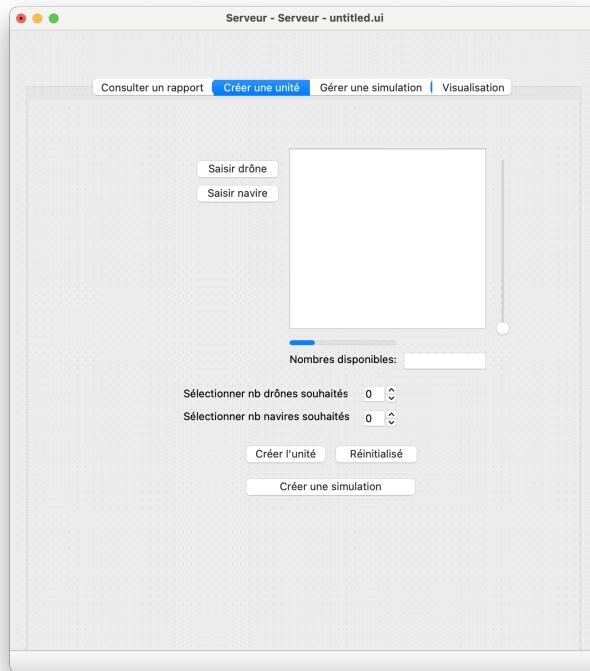
2. Tout ce qui est unité allié : les sous-marins, les drones...

3 Maquettage du produit

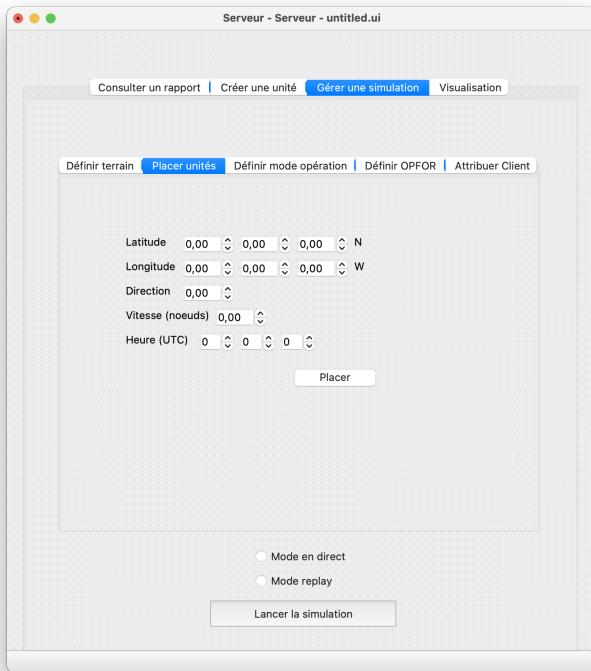
3.1 Serveur



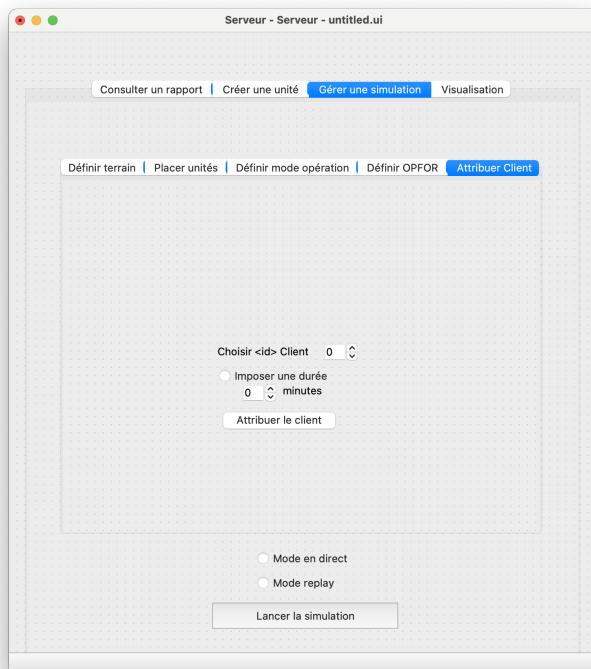
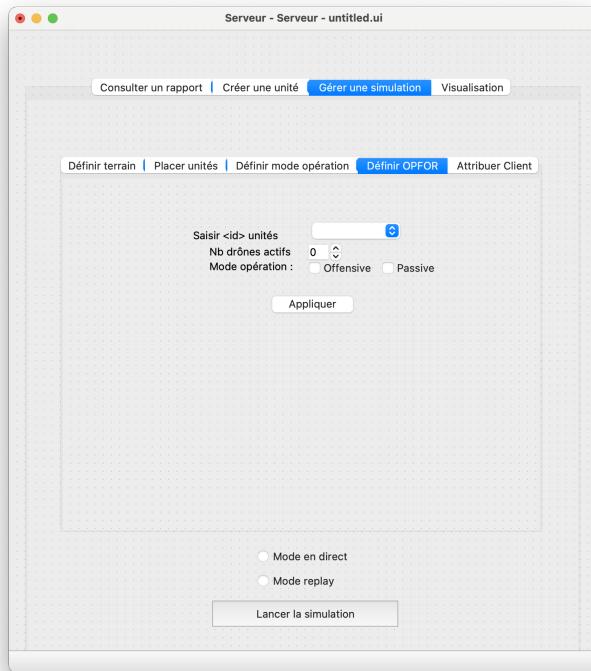
3 MAQUETTAGE DU PRODUIT



3 MAQUETTAGE DU PRODUIT



3 MAQUETTAGE DU PRODUIT



3 MAQUETTAGE DU PRODUIT

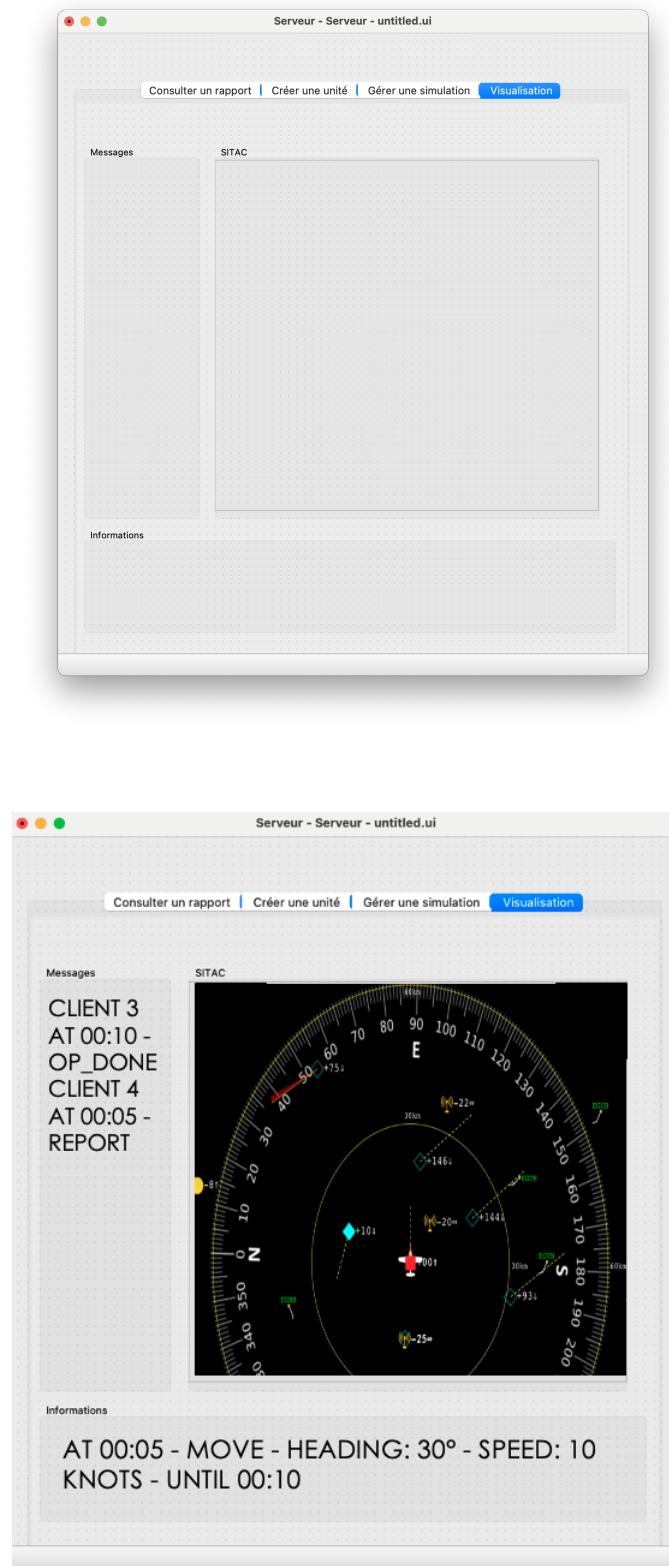
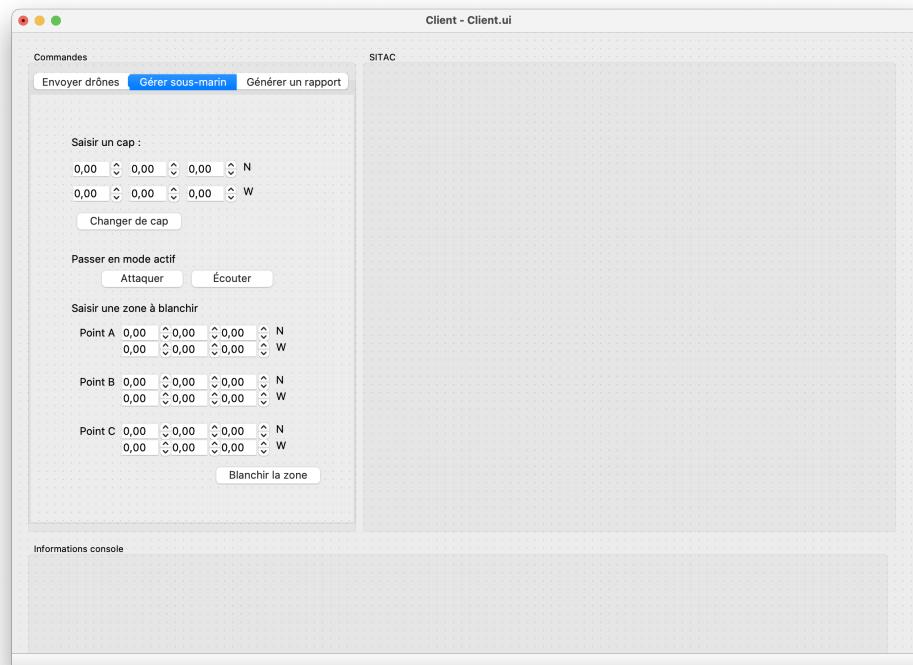
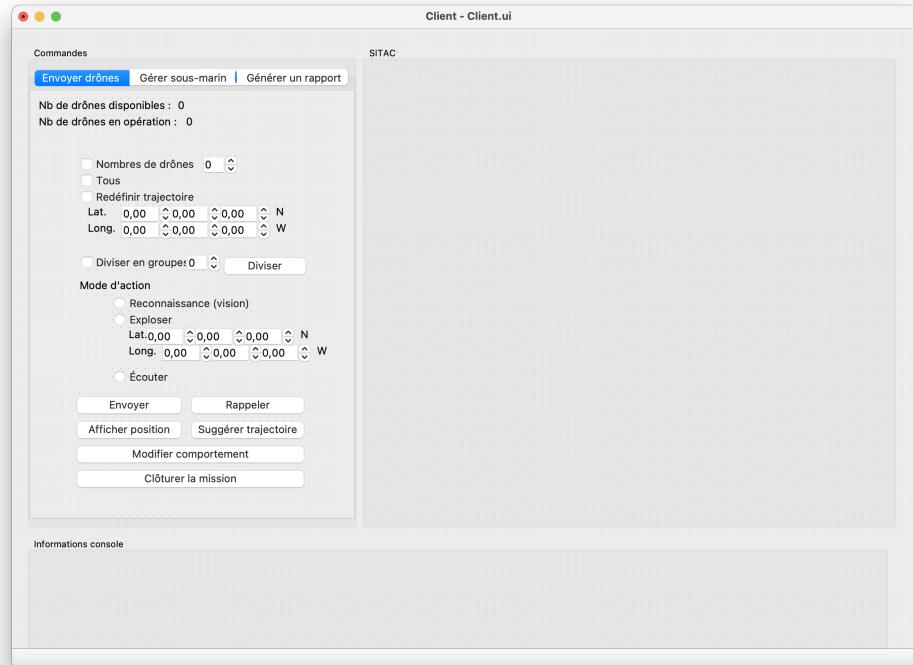


FIGURE 3 – Rendu final attendu (illustration sitac reprise du [PilotAware Rosetta](#))

3 MAQUETTAGE DU PRODUIT

3.2 Client



3 MAQUETTAGE DU PRODUIT

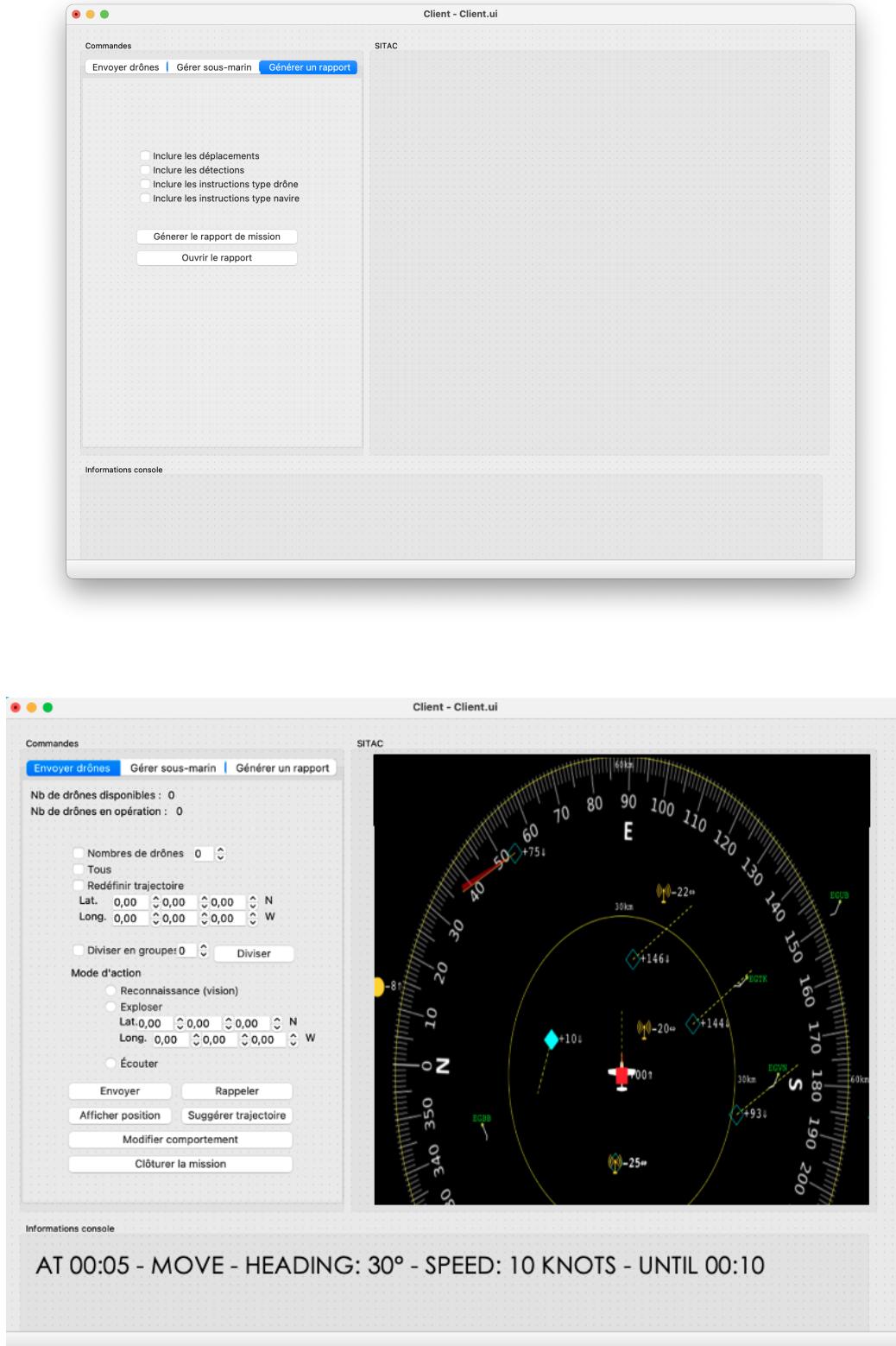


FIGURE 4 – Rendu final attendu (illustration sitac reprise du PilotAware Rosetta)

4 Conception UML

4.1 Axe fonctionnel

4.1.1 Diagramme de cas d'usage SI Serveur

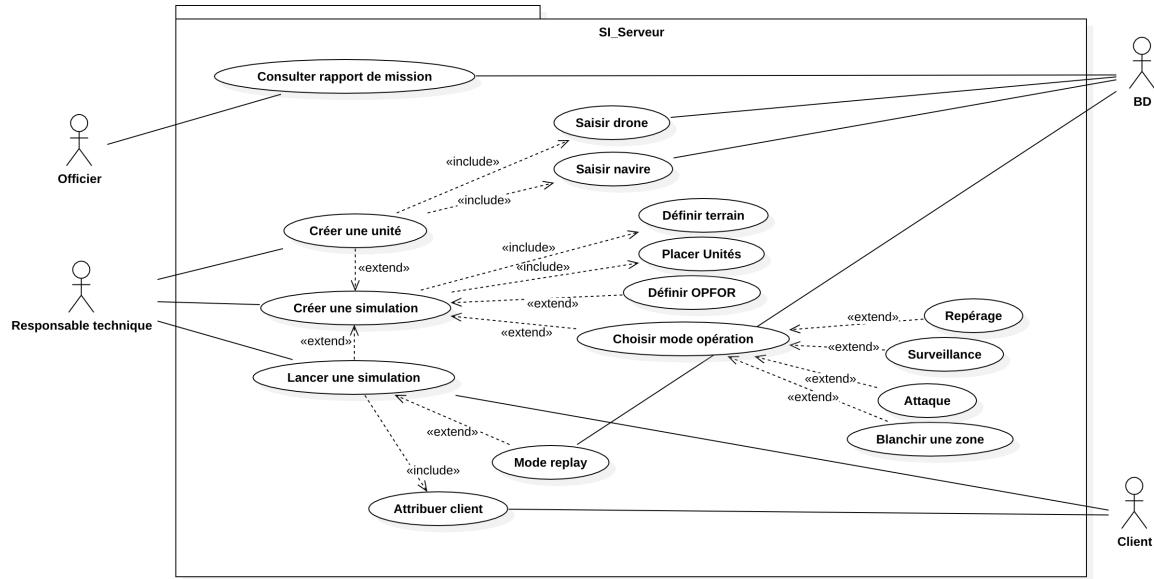


FIGURE 5 – CU SI Serveur.

4.1.2 Diagramme de cas d'usage SI Client

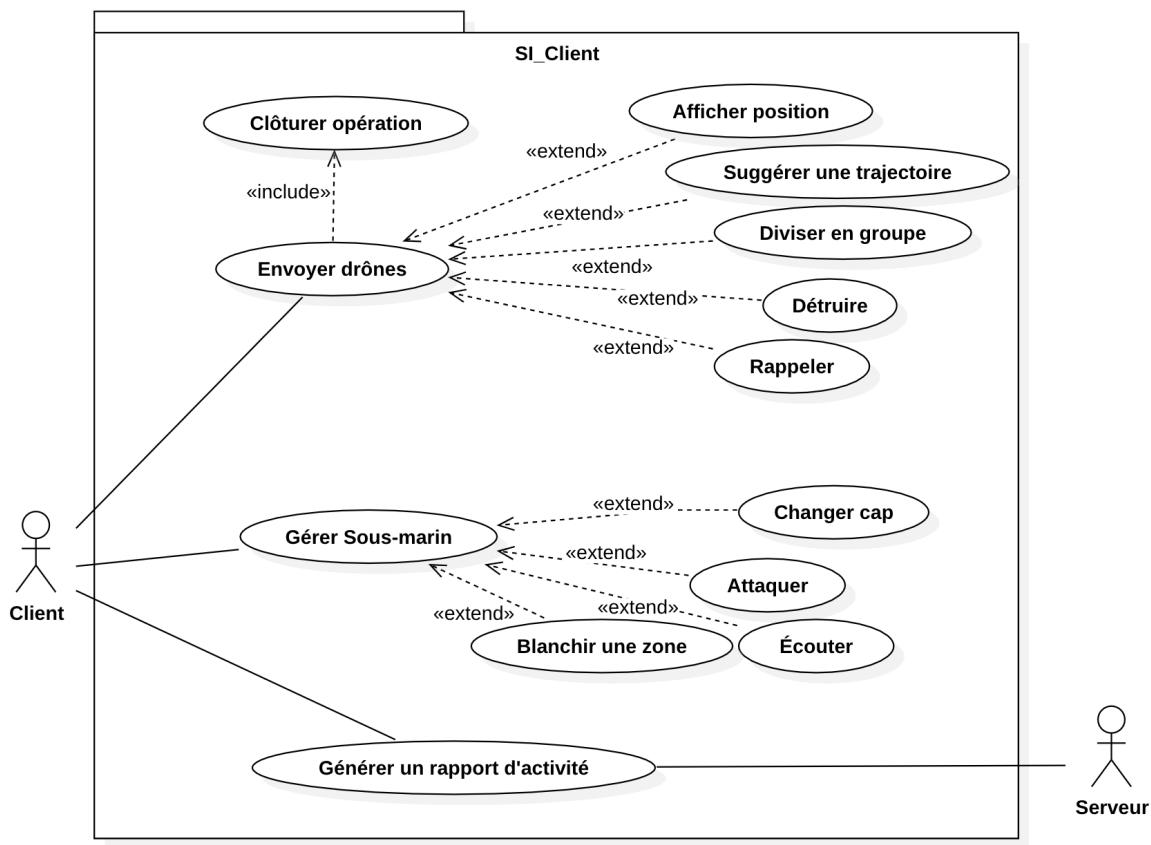


FIGURE 6 – CU SI Client.

4.1.3 Structure du système niveau 1

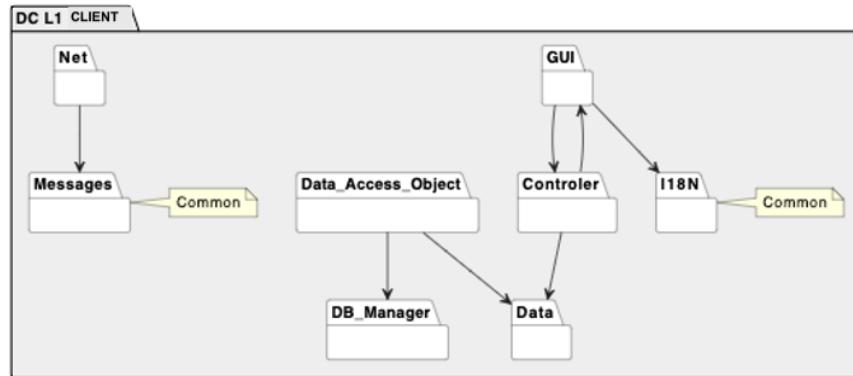


FIGURE 7 – Structure pour le logiciel Client

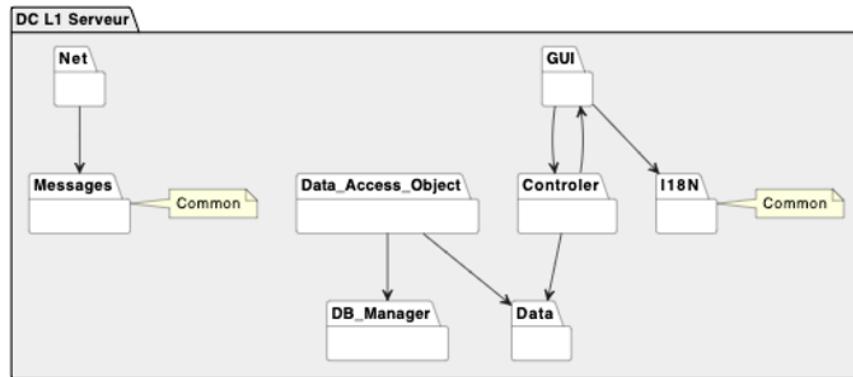


FIGURE 8 – Structure pour le logiciel Serveur.

4.1.4 Structure du système niveau 2

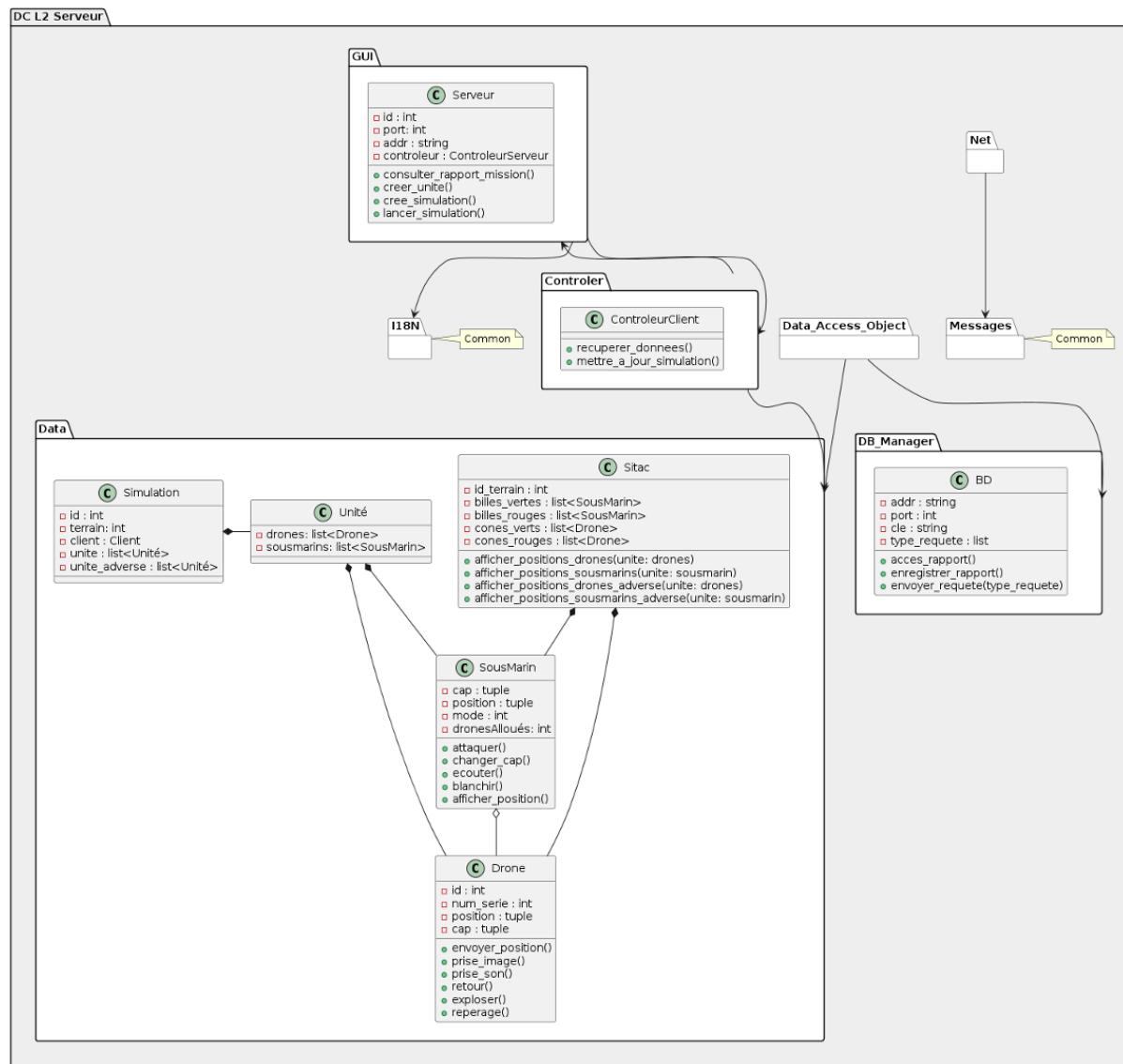


FIGURE 9 – Partie serveur

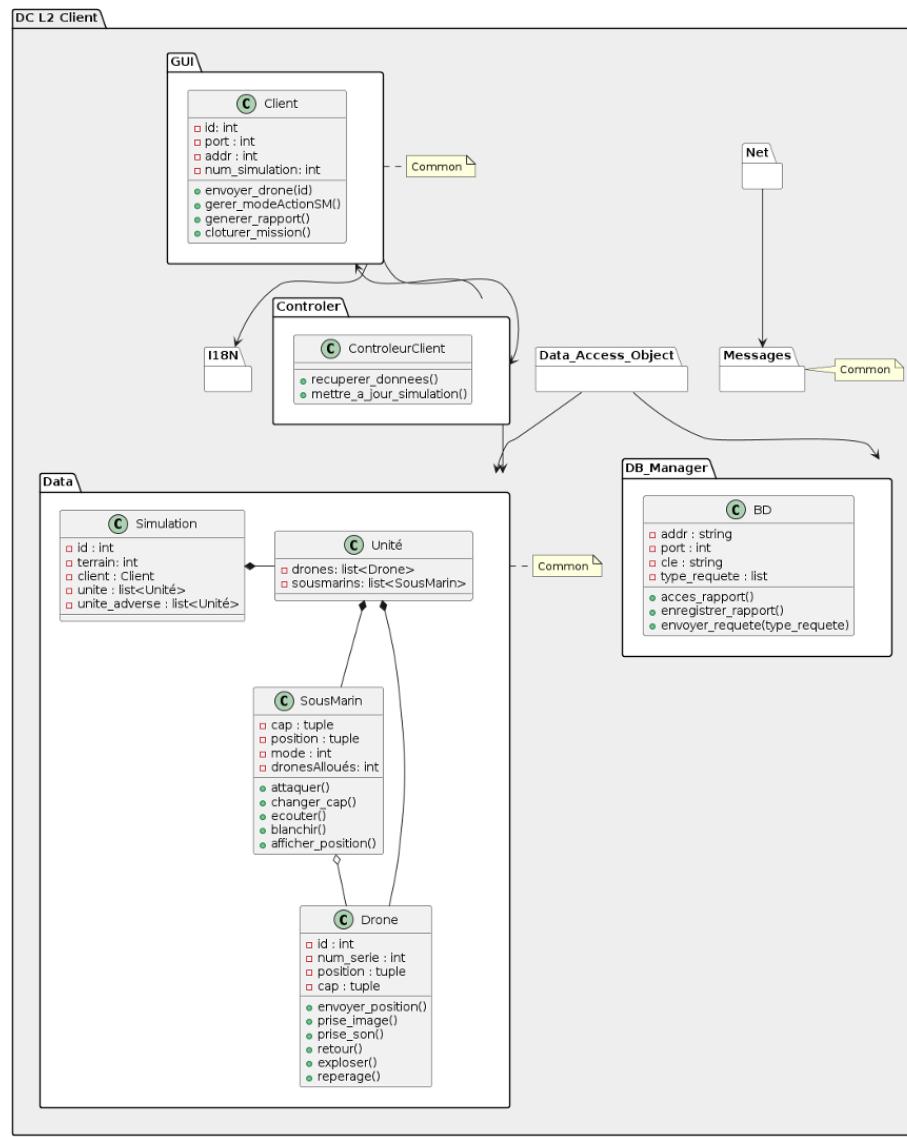


FIGURE 10 – Partie Client

4.2 Axe statique

4.2.1 Diagramme des classes partie Serveur

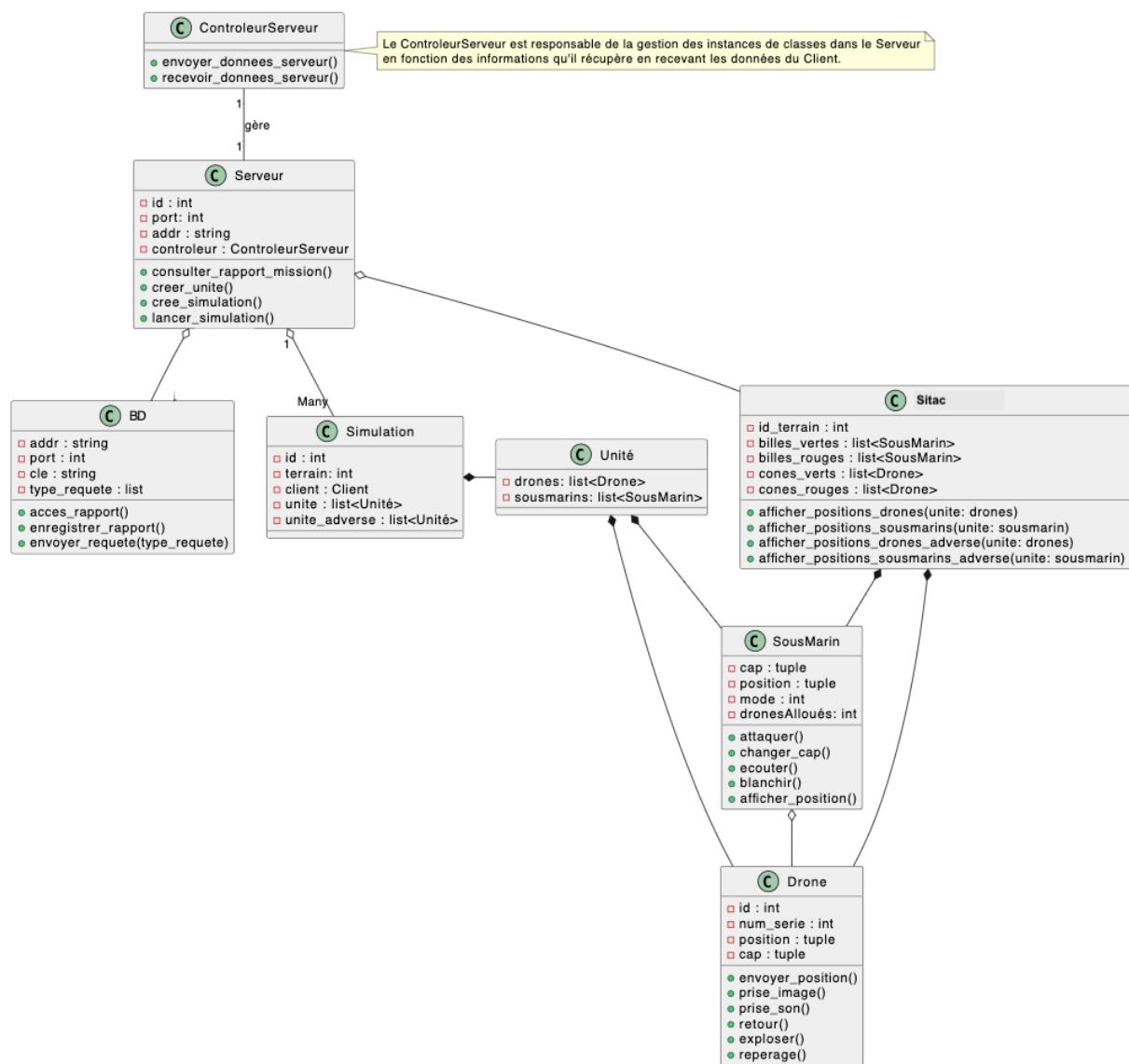


FIGURE 11 – Diagramme des classes pour le simulateur partie Serveur

4.2.2 Diagramme des classes partie Client

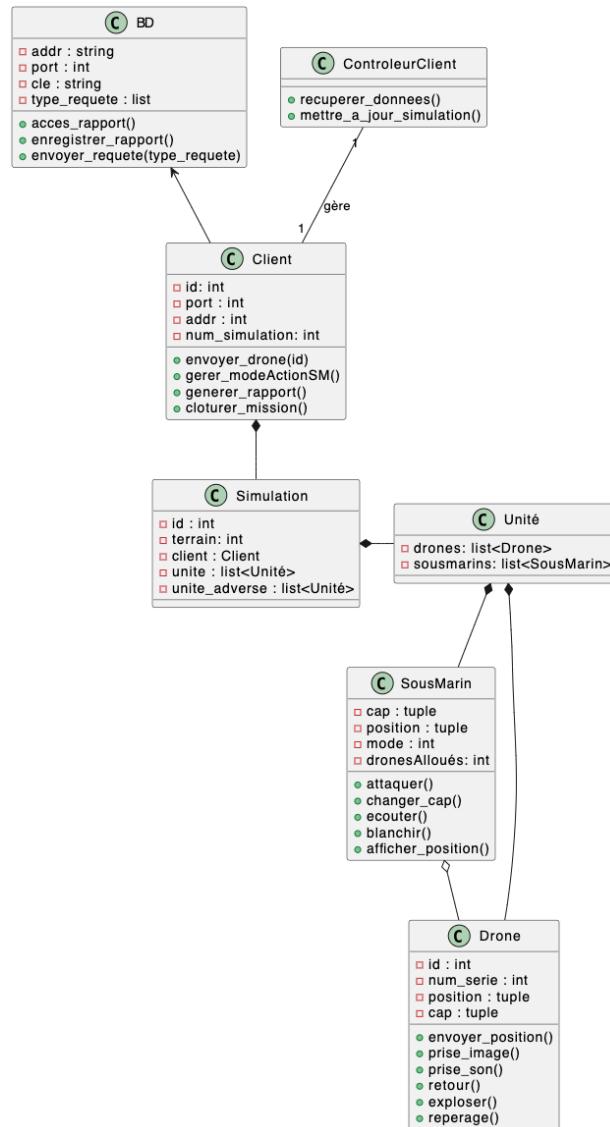


FIGURE 12 – Diagramme des classes pour le simulateur partie Client.

4.3 Axe dynamique

4.3.1 Diagramme de séquence Serveur

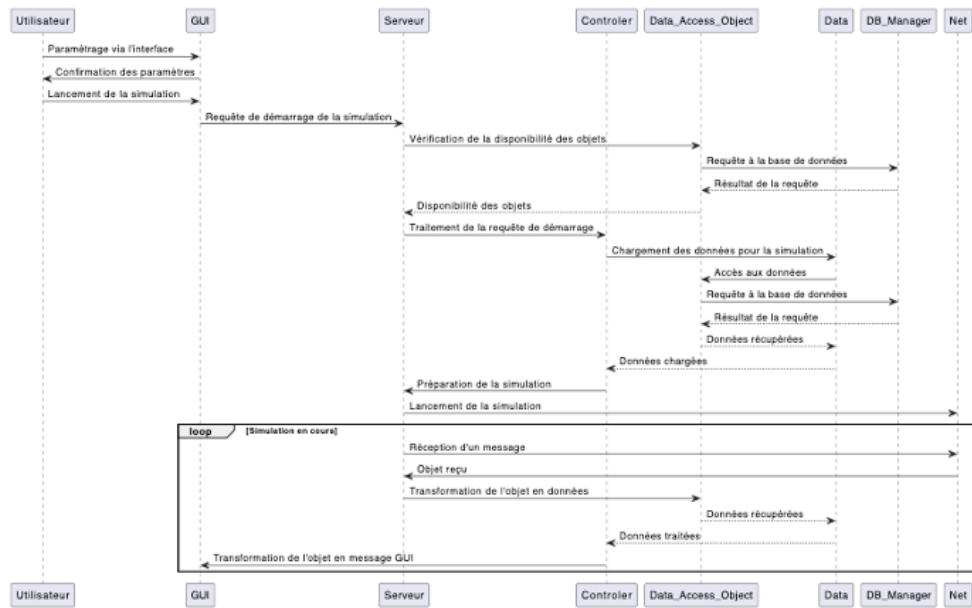


FIGURE 13 – Diag. de séq. sur l’interaction entre le serveur et ses différents composants lors de la configuration d’une simulation.

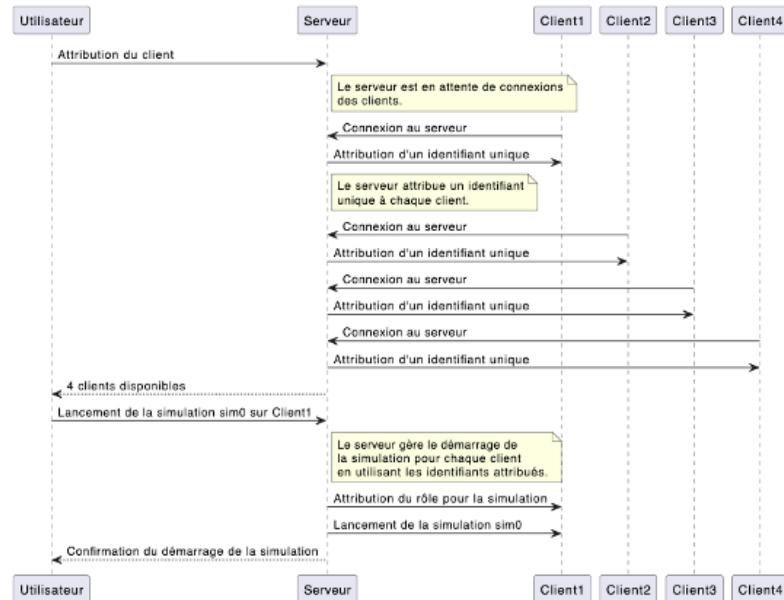


FIGURE 14 – Diag. de séq. sur l’attribution des rôles aux clients par le serveur.

4.3.2 Diagramme de séquence Client

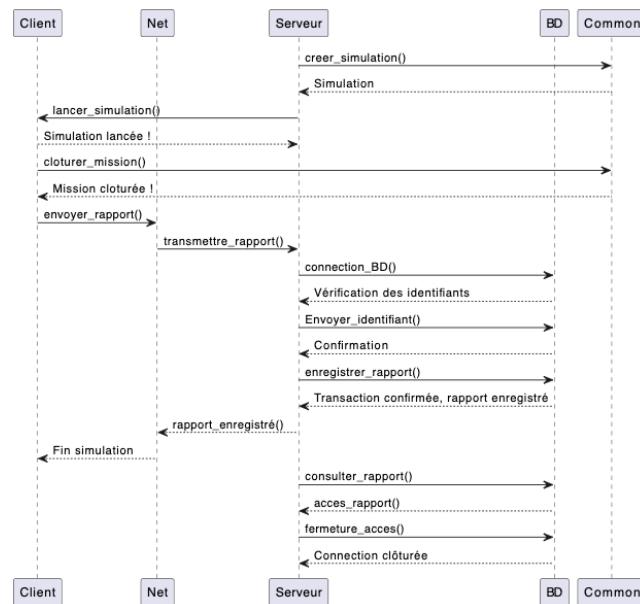


FIGURE 15 – Diag. de séq. pour l'envoi de rapport par le Client

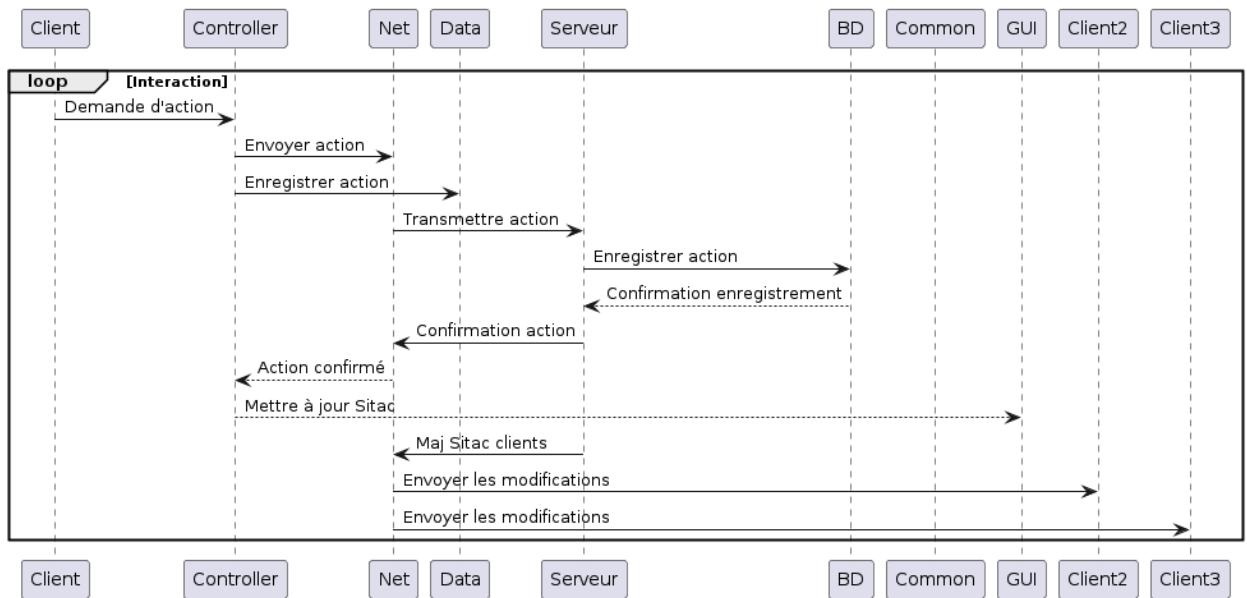


FIGURE 16 – Diag. de séq. pour l'envoi des modifications

4.3.3 Base de donnée : Modèle Entité-Association

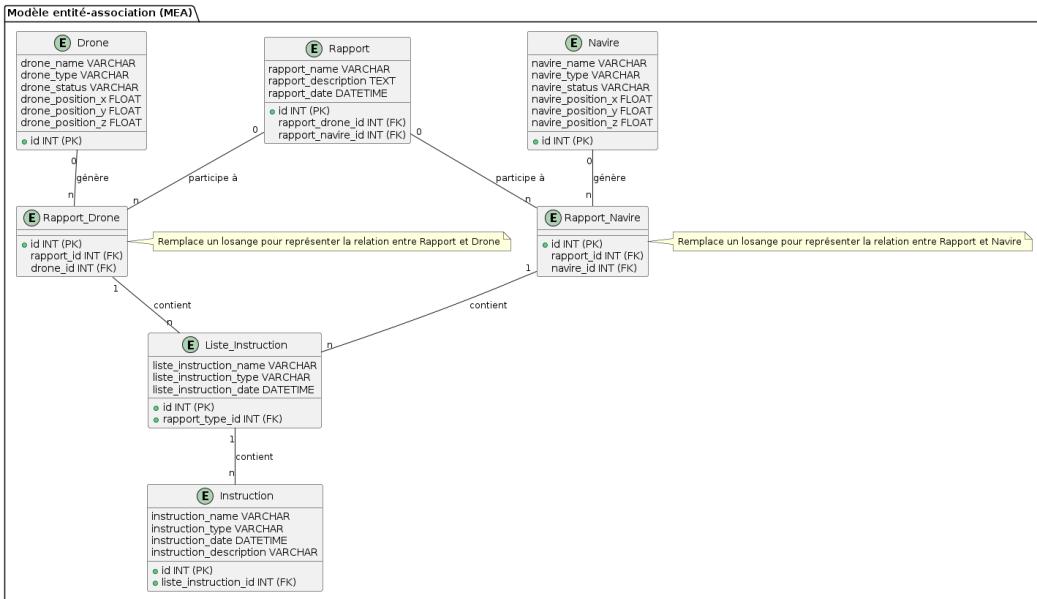


FIGURE 17 – MEA pour la BD de Serveur

5 Le manuel d'installation du logiciel

5.1 Installation des bibliothèques

5.2 Tkinter

Tkinter est une bibliothèque pour créer des interfaces graphiques en Python. Si vous utilisez Python 3, il est probablement déjà installé. Sinon, vous pouvez l'installer en utilisant pip :

```
pip install tk
```

5.3 Pillow

Pillow est une bibliothèque de traitement d'images en Python. Vous pouvez l'installer via pip :

```
pip install Pillow
```

5.4 Comment procéder à l'exécution

Pour le logiciel de configuration de simulation côté serveur :

1. Placez-vous dans le répertoire Code/server.
2. Ouvrez un terminal.
3. Exécutez le fichier client.py en utilisant la commande suivante :

```
python3 client.py
```

Pour le logiciel de simulation côté client :

1. Placez-vous dans le répertoire Code/client.
2. Ouvrez un terminal.
3. Exécutez le fichier serveur.py en utilisant la commande suivante :

```
python3 serveur.py
```

6 Le manuel utilisateur du logiciel

6.1 Application Client

L'application côté client permet trois actions différentes : "Envoyer drônes", "Gérer sous-marin" et "Générer un rapport".

- "Envoyer drônes" ordonne à un de vos drônes de commencer une opération. Vous pouvez sélectionner plusieurs drônes à la fois. L'application indiquera combien de drônes sont en opération et combien sont disponibles. Lorsque vous envoyez un drone ou plusieurs en opération, vous pouvez définir leur trajectoire, les diviser en plusieurs groupes et choisir un mode d'action entre faire de la reconnaissance, explorer ou faire de l'écoute.
- "Gérer sous-marin" permet de donner un cap au sous-marin. Vous pouvez aussi donner l'ordre d'écouter ou d'attaquer un autre sous-marin, ou de blanchir (sécuriser) une zone. Pour blanchir une zone, choisissez trois points de l'espace, ces trois points définiront la zone triangulaire à blanchir.
- "Générer un rapport" permet d'envoyer un rapport au serveur en choisissant les données à fournir. Vous pouvez consulter le rapport à tout moment.

La partie droite de l'application représente la carte avec au centre le sous-marin contrôlé. Les drônes sous le contrôle du sous-marin sont représentés par des cônes, les unités alliées en vert et les OPFOR en rouge.

6.2 Application Serveur

L'application côté serveur permet de faire les actions suivantes :

- Consulter un rapport : entrez le numéro d'une mission passée pour pouvoir consulter son rapport.
- Créer une unité : vous permet de créer un drone ou un navire qui sera l'acteur principal d'une simulation. Après avoir créé l'unité, cliquez sur "Créer une simulation".

- Gérer une simulation : après avoir créé l'acteur principal, vous permet de définir une simulation et de la lancer. Avant de pouvoir lancer la simulation, il est nécessaire de compléter toutes les étapes de la création, à savoir placer les unités de la simulation, définir le terrain, le type de simulation, l'OPFOR et définir un client pour cette simulation.
- Visualisation : affiche le déroulement de la simulation une fois lancée.

7 Le planning mis en œuvre durant le projet

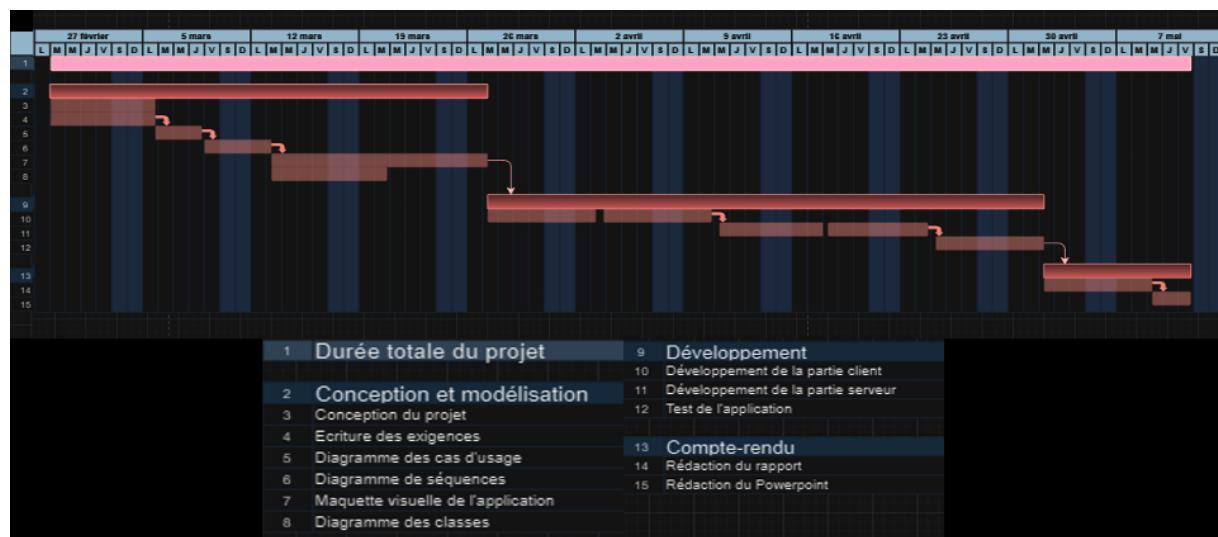


FIGURE 18 – Diagramme de Gantt pour la planification du projet.

7.1 Sprint 1

7.1.1 Sprint Planning | 27 février 2024

1. Liste de l'ensemble des demandes du client.
2. Cadrage initial du projet et définition des objectifs ; pour se faire : réflexion avec l'équipe et le représentant du client pour se familiariser avec le domaine du produit.

Tâches à réaliser

- Définition des besoins : surveillance, renseignement, reconnaissance.
- Schématisation du navires et de ses drones.
- Familiarisation avec la doctrine detection passive.
- Prise en compte de la dimension distribution serveur/client pour simuler les échanges entre les systèmes.

- Prototype papier d'une base de données pour contenir les rapports de détection, les caractéristiques des drones et l'état des missions.

7.1.2 Daily Scrum du 12 Mars 2024

Objectif(s)

- Liste de l'ensemble des demandes et prise en compte des retours du client.

Tâches réalisées

- Listes des exigences fonctionnelles.
- Échange entre le client, son représentant pour éclaircir les zones de flou sur le fonctionnement du système et lever des barrières et des interrogations autour des différents usages du système à travers d'un schéma et d'échanges.

7.2 Sprint 2

7.2.1 Sprint planning | 19 Mars 2024

Objectif(s)

- Définitions des objectifs à réaliser entre l'équipe et les représentants du client.
- Planification des tâches.

Tâches réalisées

- Réalisation du diagramme de cas d'usage.
- Liste des différentes fonctions pour chaque mode du simulateur.
- Définition des contraintes du système et de son articulation.

7.2.2 Sprint Review du 26 mars 2024

Tâches réalisées

- Réalisation du diagramme statique (objet).
- Assimilation des rôles pour la méthode Agile Scrum.
- Recherche de développement du code : PyQt ou Tkinter.

7.2.3 Daily Scrum | 8 Avril 2024

Objectif(s)

- Clarification de ce qui a été réalisé, des différentes tâches à finaliser et démarrage d'un cycle extreme programming.

Tâches réalisées

- Échange au tableau sur la conception IHM de notre simulateur.
- Développement en binôme : Clément pour la partie serveur et Mattéo pour la maquette de l'interface. Programmation modulaire avec Tkinter en Python3.
- Réalisation et reprise des diagrammes UML.

7.2.4 Sprint Review | 16 Avril 2024

Objectif(s)

- Vérification du code et remise en question du développement.

Tâches réalisées

- Ajustement de l'IHM à partir de la maquette.
- Échange avec le client pour réviser l'approche client-serveur.

7.2.5 Sprint Retrospective 1 | 23 Avril 2024

Objectif(s)

- Clarification de ce qui a été réalisé.
- Reprise des diagrammes UML.
- Développement des classes selon le MVC.

7.2.6 Sprint Retrospective 2 | 29 Avril 2024

Clarifications

- Briefing sur ce qui a été réalisé.
- Réalisation d'un diagramme de Gantt.
- Réalisation d'un diagramme de séquence dynamique
- Vérification du code selon le modèle UML.

8 Diagramme de Gantt

29 Avril 2024

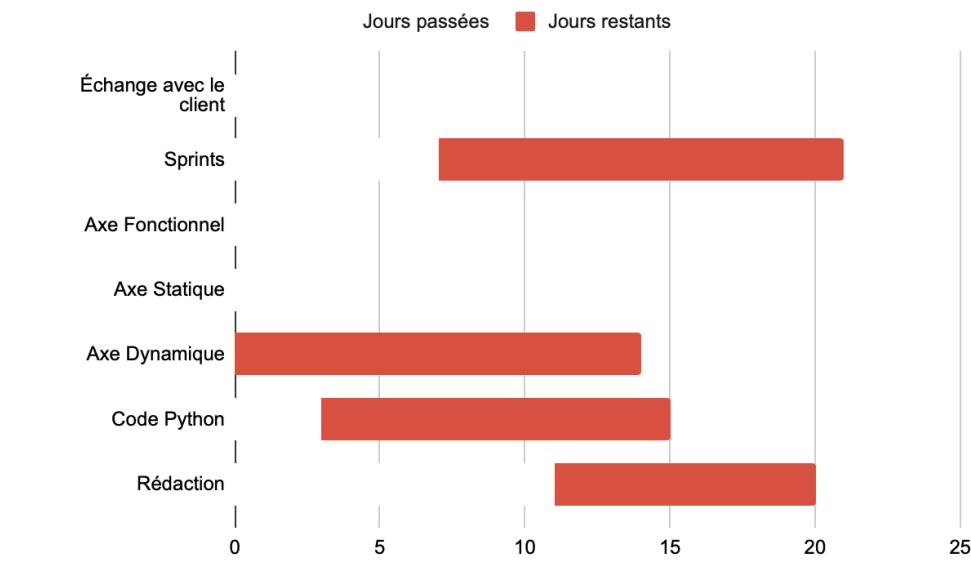


FIGURE 19 – Diagramme de Gantt le 29 Avril 2024

TABLE 1 – Planning du Projet

	Date de début	Jours passées	Jours restants	TOTAL
Échange avec le client	27/02/2024	3	0	01/03/2024
Sprints	27/02/2024	7	14	19/03/2024
Axe Fonctionnel	23/04/2024	4	0	27/04/2024
Axe Statique	19/03/2024	3	0	22/03/2024
Axe Dynamique	29/04/2024	0	14	13/05/2024
Code Python	29/04/2024	3	12	14/05/2024
Rédaction	05/03/2024	11	9	15/05/2024

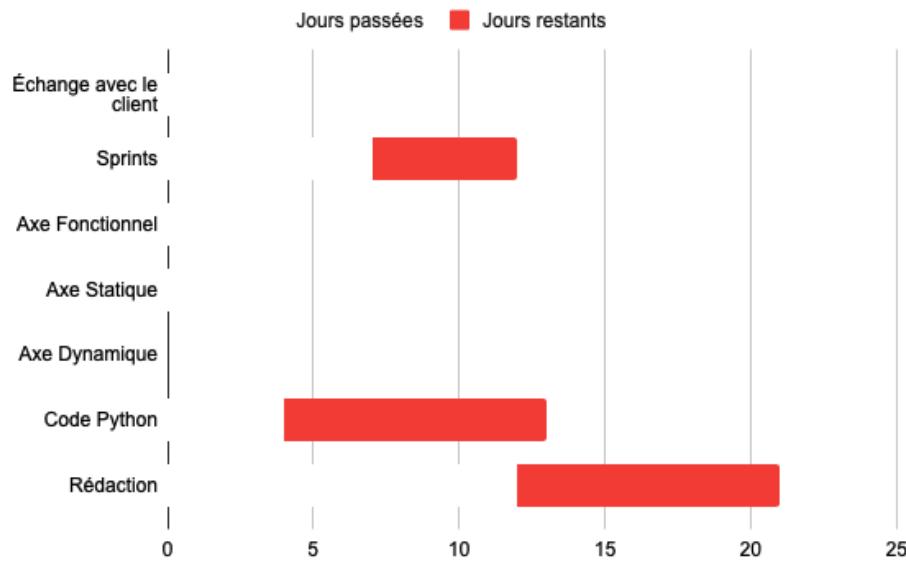
8 Mai 2024

FIGURE 20 – Diagramme de Gant le 8 Mai 2024

	Date de début	Jours passées	Jours restants	TOTAL
Échange avec le client	27/02/2024	4	0	01/03/2024
Sprints	27/02/2024	7	5	19/03/2024
Axe Fonctionnel	23/04/2024	4	0	27/04/2024
Axe Statique	19/03/2024	3	0	22/03/2024
Axe Dynamique	29/04/2024	0	0	13/05/2024
Code Python	29/04/2024	4	9	17/05/2024
Rédaction	05/03/2024	12	9	17/05/2024

Bilan de la méthode Agile

Nous avons pu rectifier le livrable que nous allons fournir en un seul logiciel, en deux logiciels distincts (client et serveur). Ces améliorations ont été discutées lors des échanges, tout comme la création d'une interface graphique pour la partie client alors que nous pensions plutôt à une interaction par commande via un terminal.