
Dossier de conception

Incrément 1

Responsable du document : Hugo BOUY

État du document : Version finale

AVERTISSEMENT :

Le présent document est un document à but pédagogique. Il a été réalisé sous la direction de Jérôme DELATOUR, en collaboration avec des enseignants et les étudiants de l'option SE, groupe A1 (Hugo BOUY, Bastien CASSAR, Paul CHIRON, Paul JURET, Laurent LETENNEUR, Mathis MOULIN, Romain TROVALLET) du groupe ESEO. Ce document est la propriété de Jérôme DELATOUR du groupe ESEO. En dehors des activités pédagogiques de l'ESEO, ce document ne peut être diffusé ou recopié sans l'autorisation écrite de ses propriétaires.

Table des versions

Date	Actions	Auteur	Version	Révision
19/04/2023	Corrections MAE, diagramme de séquence et architecture candidate suite à l'AC	Tout le groupe	1.1	0
18/04/2023	Relecture et correction finale	Tout le groupe	1.0	0
18/04/2023	Rédaction partie 2.3.1	Romain TROVALLET	0.6	0
17/04/2023	Correction partie 2.3 et ajout MAE Cameraman	Hugo BOUY	0.5	20
17/04/2023	Mise à jour MaE GUI	Bastien CASSAR	0.5	19
17/04/2023	Correction architecture candidate	Hugo BOUY	0.5	18
17/04/2023	Ajout descriptions textuelles des classes cameraman et recognitionAI	Paul JURET	0.5	17
17/04/2023	Ajout de la partie 2.3.3.6 (classe Guard) et 2.3.3.7 (classe Bouncer)	Hugo BOUY	0.5	16
17/04/2023	Description textuelle des classes UISP et doorManager	Mathis MOULIN	0.5	15
16/04/2023	Ajout description textuelle de GUI	Bastien CASSAR	0.5	14
16/04/2023	Ajout classe EmployeeManager et MaE Cameraman	Paul CHIRON	0.5	13
16/04/2023	Ajout diagramme de classe connectionManager et Clock avec leur description	Laurent Letenneur	0.5	12
16/04/2023	Correction diagrammes de séquences	Hugo BOUY	0.5	11
16/04/2023	Ajout diagramme de classe UISS	Romain TROVALLET	0.5	10
15/04/2023	Ajout du diagramme de classe	Hugo BOUY	0.5	9
15/04/2023	Ajout machine à état Bouncer	Paul JURET	0.5	8
15/04/2023	Ajout du diagramme de séquence du CU quitter	Hugo BOUY	0.5	7
15/04/2023	Ajout des description des diagrammes de séquence stratégique et seConnecter	Laurent Letenneur	0.5	6
15/04/2023	Ajout MAE connectionManager	Laurent Letenneur	0.5	5

Date	Actions	Auteur	Version	Révision
15/04/2023	Ajout des MàE de GUI	Bastien CASSAR	0.5	4
13/04/2023	Ajout diagramme de séquence du CU Regarder vidéo et Consulter Calendrier	Bastien CASSAR	0.5	3
12/04/2023	Ajouts diagrammes de séquence	Romain TROVALLET	0.5	2
11/04/2023	Modification diagramme de séquence stratégique et seConnecter	Laurent Letenneur	0.5	1
09/04/2023	Organisation 2.2, 2.3 et Conception Détaillée	Romain TROVALLET	0.5	0
08/04/2023	Rédaction 2.1	Romain TROVALLET	0.4	0
08/04/2023	Rédaction introduction	Romain TROVALLET	0.3	0
05/04/2023	Ajout diagramme de séquence CU initialiser Soft-Sonnette	Paul JURET	0.2	1
04/04/2023	Rédaction 2.2	Romain TROVALLET	0.2	0
03/04/2023	Ajout diagramme de séquence CU Consulter Liste employés	Mathis MOULIN	0.1	8
03/04/2023	Ajout diagramme de séquence Entrer et MaE UISS	Paul CHIRON	0.1	7
03/04/2023	Ajout diagramme de séquence stratégique et seConnecter	Laurent Letenneur	0.1	6
03/04/2023	Création diagramme séquence CU Contrôler Porte À Distance	Romain TROVALLET	0.1	5
30/03/2023	Ajustement de la présentation de l'architecture candidate	Hugo BOUY	0.1	4
30/03/2023	Modification architecture candidate	Paul CHIRON	0.1	3
30/03/2023	Mise à jour de l'architecture candidate vers une v2	Hugo BOUY	0.1	2
21/03/2023	Ajustement mise en page pour correspondre au PAQL	Hugo BOUY	0.1	1
21/03/2023	Rédaction introduction	Paul CHIRON	0.1	0
18/03/2023	Création du document	Paul CHIRON	0.0	0

Table des matières

Table des versions	2
Table des matières	4
1 Introduction	6
1.1 Objet	6
1.2 Portée	6
1.3 Définitions, acronymes et abréviations	7
1.4 Références	8
1.5 Vue d'ensemble	8
1.6 Versions	9
2 Conception générale	10
2.1 Architecture candidate	10
2.2 Grands principes de fonctionnement	11
2.2.1 Présenter les capacités de la STM32MP15 au travers d'une application de Sonnette Connectée	12
2.2.2 Initialiser SoftSonnette	13
2.2.3 Se connecter	13
2.2.4 Demander à entrer	14
2.2.5 Ouvrir porte	14
2.2.6 Regarder vidéo	15
2.2.7 Consulter calendrier	15
2.2.8 Contrôler Porte à distance	16
2.2.9 Consulter liste employés	17
2.2.10 Quitter SaE	17
2.3 Description des composants	18
2.3.1 Description des types manipulés entre composants	18
2.3.2 Diagramme de classe	19
2.3.3 Description des classes	20
2.3.3.1 [Object] ConnectionManager	20
2.3.3.1.1 Philosophie de conception	20
2.3.3.1.2 Description structurelle	20
2.3.3.1.2.1 Attributs	20
2.3.3.1.2.2 Services offerts	20
2.3.3.1.3 Description comportementale	20
2.3.3.2 [IHM] GUI	21
2.3.3.2.1 Philosophie de conception	22
2.3.3.2.2 Description structurelle	22
2.3.3.2.2.1 Attributs	22
2.3.3.2.2.2 Services offerts	23
2.3.3.2.3 Description comportementale	24
2.3.3.3 [Object] EmployeeManager	25
2.3.3.3.1 Philosophie de conception	25
2.3.3.3.2 Description structurelle	25
2.3.3.3.2.1 Attributs	25
2.3.3.3.2.2 Services offerts	25

2.3.3.4	[Object] Clock	26
2.3.3.4.1	Philosophie de conception	26
2.3.3.4.2	Description structurelle	26
2.3.3.4.2.1	Attributs	26
2.3.3.4.2.2	Services offerts	26
2.3.3.5	[Object] Cameraman	26
2.3.3.5.1	Philosophie de conception	26
2.3.3.5.2	Description structurelle	27
2.3.3.5.2.1	Attributs	27
2.3.3.5.2.2	Services offerts	27
2.3.3.6	[Object] Guard	27
2.3.3.6.1	Philosophie de conception	27
2.3.3.6.2	Description structurelle	27
2.3.3.6.2.1	Attributs	27
2.3.3.6.2.2	Services offerts	27
2.3.3.7	[Object] Bouncer	28
2.3.3.7.1	Philosophie de conception	28
2.3.3.7.2	Description structurelle	28
2.3.3.7.2.1	Attributs	28
2.3.3.7.2.2	Services offerts	28
2.3.3.7.3	Description comportementale	28
2.3.3.8	[IHM] UISS	29
2.3.3.8.1	Philosophie de conception	29
2.3.3.8.2	Description structurelle	29
2.3.3.8.2.1	Attributs	29
2.3.3.8.2.2	Services offerts	29
2.3.3.8.3	Description comportementale	30
2.3.3.9	[Object] DoorManager	31
2.3.3.9.1	Philosophie de conception	31
2.3.3.9.2	Description structurelle	31
2.3.3.9.2.1	Attributs	31
2.3.3.9.2.2	Services offerts	31
2.3.3.9.3	Description comportementale	31
2.3.3.10	[IHM] UISP	32
2.3.3.10.1	Philosophie de conception	32
2.3.3.10.2	Description structurelle	32
2.3.3.10.2.1	Attributs	32
2.3.3.10.2.2	Services offerts	32
2.3.3.11	[Object] RecognitionAI	32
2.3.3.11.1	Philosophie de conception	32
2.3.3.11.2	Description structurelle	32
2.3.3.11.2.1	Attributs	32
2.3.3.11.2.2	Services offerts	33

3	Dictionnaire du domaine	34
----------	--------------------------------	-----------

4	Table des figures	37
----------	--------------------------	-----------

1 Introduction

1.1 Objet

Ce dossier de conception a pour objectif de rassembler toute la conception du logiciel PSC. Il permettra à l'équipe A1 de développer le logiciel ainsi qu'élaborer des tests.

Les éléments de conception présentés dans ce document ont été déterminés suite à l'étude du dossier de spécification [SPEC_A1].

Ce dossier de conception suit les recommandations de la norme IEEE 29148 [IEEE 29148 :2018]. Il utilise des schémas et illustrations respectant la norme UML en version 2.5.1 [UML_2.5.1_2017]. Il respecte les exigences du Plan d'Assurance Qualité Logicielle (PAQL) défini par l'équipe A1 [PAQL_A1].

1.2 Portée

Ce document décrit les éléments de conception du Système à l'Étude (SaE). Il est destiné :

- À l'équipe de développement C et celle de développement Android afin de préciser l'implémentation des objets constituant le SaE.
- Aux testeurs, afin qu'ils puissent élaborer les tests adéquats vérifiant la philosophie de conception adaptée par l'équipe.
- Aux auditeurs de la société FORMATO lors de leurs différents suivis du projet.
- Au client pour que le cadre du projet et la direction prise par l'équipe soient claires et dans la continuité des spécifications.

1.3 Définitions, acronymes et abréviations

Terme	Description
CdC	Cahier des charges fourni par le client
Client	Société STMicroelectronics
CU	Cas d'utilisation
Disponibilité	La disponibilité est l'aptitude d'un composant ou d'un système à être en état de marche à un instant donné
Fiabilité	La fiabilité est l'aptitude d'un composant ou d'un système à fonctionner pendant un intervalle de temps
IHM	Interface Homme Machine. Moyens permettant aux utilisateurs d'AOP d'interagir avec AOP
Maintenabilité	La maintenabilité est l'aptitude d'un composant ou d'un système à être maintenu ou remis en état de fonctionnement
N.A	Non Applicable
OMG	(Object Management Group) Association professionnelle internationale définissant entre autres des normes dans le domaine informatique
RTC	Une RTC est une horloge temps-réel, c'est un module présent sur la Board
SàÉ	Système à l'Étude. Il s'agit de l'ensemble des logiciels AOP, SoftPorte et SoftSonnette
UML	Notation graphique normalisée, définie par l'OMG et utilisée en génie logiciel

1.4 Références

Document	Référence
[ISO/IEC/IEEE 29148 :2018]	ISO/IEC/IEEE, “International Standard, Systems and software engineering — Life cycle processes — Requirements engineering”, https://standards.ieee.org/standard/29148-2018.html .
[UML_2.5.1_2017]	OMG, “Unified Modeling Language”, version 2.5.1, , 2017
[PAQL_A1]	« Plan d’Assurance Qualité Logicielle », 2023, Dépôt SE 2024 A1 - ST, chemin d’accès : /qualite/livrables/PAQL_A1.pdf
[SPEC_A1]	« Dossier de spécification », 2023, Dépôt SE 2024 A1 - ST, chemin d’accès : /specification/livrables/SPEC_A1.pdf
[ST_Sonnette_connectée_industrielle_2023]	STMicroelectronics, « Cahier des charges initial pour le projet », 2023, chemin d’accès : /gestion_de_projet/client/ST_Sonnette_connectée_industrielle.pdf
[CR_R2_09_02_2023]	« Compte rendu de la réunion du 09/02/2023 avec les clients », chemin d’accès : /gestion_de_projet/réunions/CR_R2_09_02_2023.pdf

TABLE 1 – Références des documents utilisés dans ce dossier

1.5 Vue d’ensemble

Ce document de conception est structuré en plusieurs parties :

- La partie I définit l’objet et la portée du document ainsi qu’une liste des abréviations utilisées dans ce document et les références des documents cités.
- La partie II concerne la conception générale du prototype PSC. Cette partie présente l’architecture candidate et donne les grands principes de fonctionnement du projet PSC. Elle détaille ensuite chaque composante du système, en présentant pour chacune leur description structurelle. Une description comportementale est présente pour les composantes actives ayant une machine à états.
- La partie III présentera la conception détaillée. Cette partie présente les composantes du système en précisant cette fois-ci la gestion des entrées et des sorties, le multitâche ainsi que la gestion de la persistance.
- Un dictionnaire de domaine constitue la dernière partie du document.

1.6 Versions

Le projet PSC est réalisé en deux incréments. Le présent dossier de conception détaille le système dans sa version globale. Il est important de noter que lors de la réalisation du code source, certaines fonctionnalités ne seront pas implémentées lors de l'incrément 1. La répartition du développement des fonctionnalités est donné à titre indicatif :

- Lors de la version 1, la connexion entre AOP et SoftSonnette est initialisée, et le flux vidéo est affiché sur AOP et SoftSonnette. Le calendrier des horaires peut être affiché, et la fonction reconnaissance faciale est fonctionnelle à partir d'une photo pré-enregistrée.
- Lors de la version 2, le stockage persistant des informations testeurs est implémentée, et les fonctions de contrôle de la porte et de gestion des employés sont développées. Enfin, la communication avec SoftPorte est réalisée.

2 Conception générale

2.1 Architecture candidate

Dans le diagramme suivant est présenté l'architecture candidate du SàE. Les différents objets sont présentes ainsi que leurs méthodes.

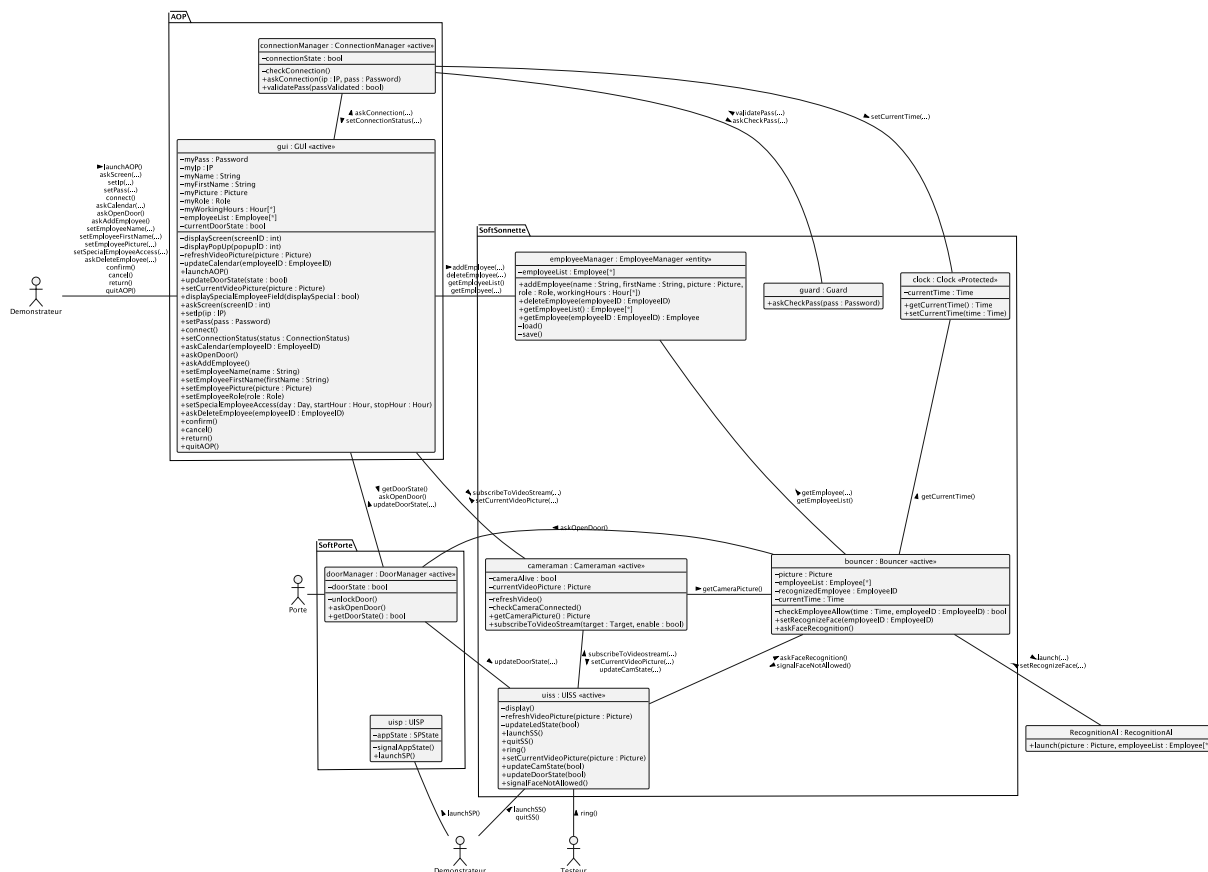


FIGURE 1 – Architecture candidate

Lors de la partie conception générale, est faite l'hypothèse d'un système matériel à ressources infinies. On distingue dans cette représentation 3 grandes entités : AOP, SoftSonnette et Soft-Porte.

- AOP est l'application développée sur le smartphone du Démonstrateur. Elle contient les objets suivants :
 - Connection Manager : Permet de gérer la connexion avec SoftSonnette.
 - GUI : Interface graphique permettant à l'utilisateur d'interagir avec AOP.
- SoftSonnette est l'application exécutée sur le microprocesseur. Elle contient les objets suivants :
 - Employee Manager : Permet de gérer les données des employés.
 - Clock : Permet à SoftSonnette d'avoir accès à une horloge.
 - Cameraman : Permet de gérer le flux vidéo.
 - Guard : Vérifie le mot de passe pour la connexion entre SoftSonnette et AOP.

- Bouncer : Permet de donner l'accès à l'employé.
- UISS : Interface graphique permettant à l'utilisateur d'interagir avec SoftSonnette.
- SoftPorte est l'application exécutée sur le microcontrôleur. Elle contient les objets suivants :
 - Door Manager : Permet d'ouvrir la porte.
 - UISP : Interface physique permettant à l'utilisateur de visualiser l'état de SoftPorte.

On remarque aussi la présence d'un objet externe à ces entités se nommant RecognitionAI. Il s'agit d'un programme externe assurant la reconnaissance faciale à partir d'une photo. Ce programme n'est pas développé par l'équipe projet. Aucune garantie de fonctionnement n'est donc apportée dessus.

2.2 Grands principes de fonctionnement

Cette partie représente le fonctionnement général de SoftSonnette, SoftPorte, AOP et leurs interactions respectives. Les diagrammes de séquences présentés sont :

- Présenter les capacités de la STM32MP15 au travers d'une application de Sonnette Connec-tée
- Initialiser SoftSonnette
- Se connecter
- Demander à entrer
- Regarder vidéo
- Consulter calendrier
- Contrôler Porte à distance
- Consulter liste des employés
- Quitter SaE

Les diagrammes de séquence présentent l'histoire nominal des Cas d'Utilisation du dossier de spécification [SPEC_A1]. Ils ne traitent donc pas les erreurs de fonctionnement.

2.2.1 Présenter les capacités de la STM32MP15 au travers d'une application de Sonnette Connectée

Ce diagramme représente le scénario nominal de l'ensemble des CUs du dossier de spécification [SPEC_A1].

Le démonstrateur lance les diverses applications et peut réaliser plusieurs actions :

Il peut se connecter à AOP pour pouvoir, au choix, ouvrir la porte à distance, consulter le calendrier d'un employé, ajouter ou supprimer un employé ou regarder la video.

Le testeur également demander à entrer.

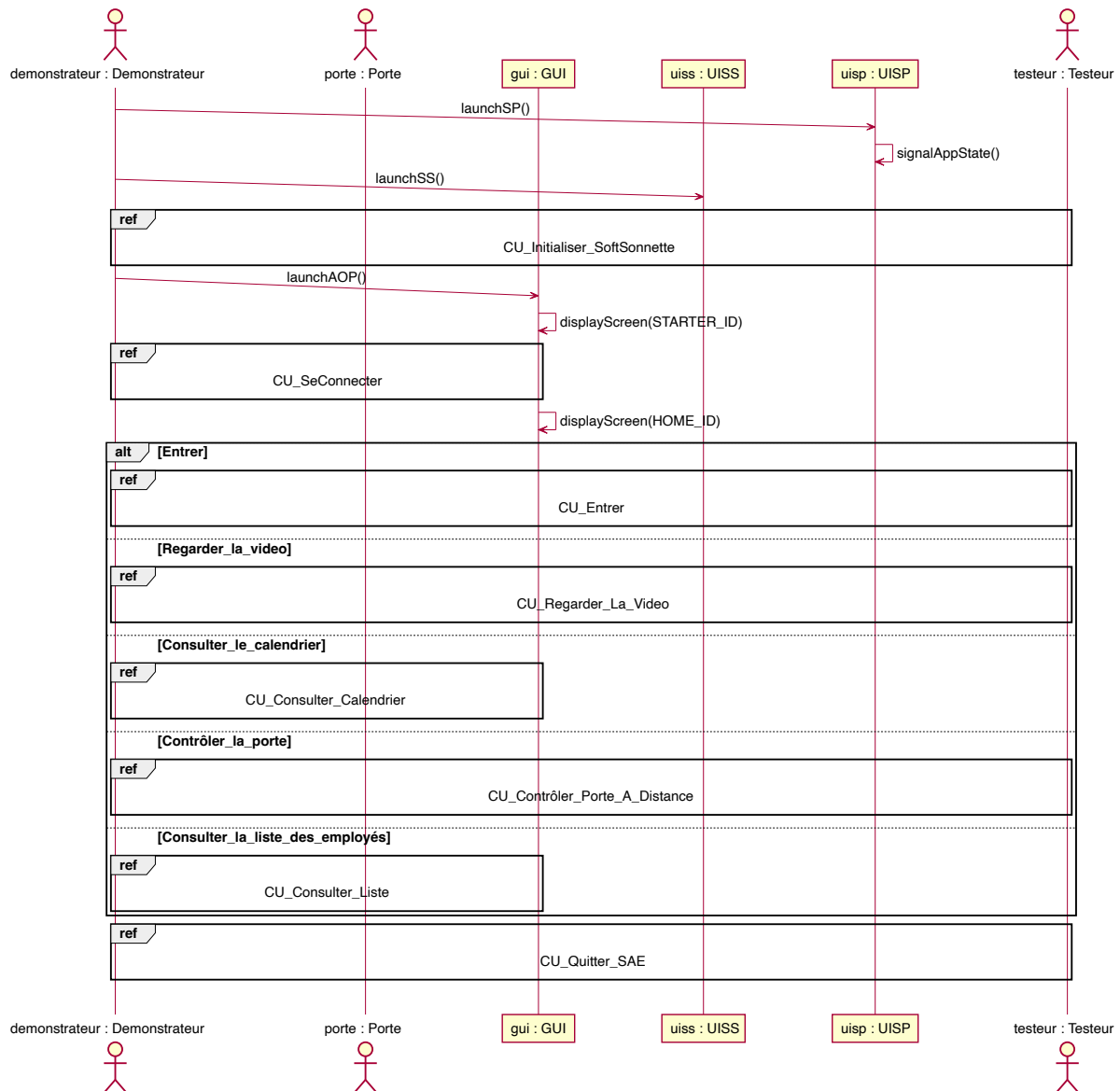


FIGURE 2 – Diagramme de séquence du scénario nominal

2.2.2 Initialiser SoftSonnette

Ce diagramme représente le scénario nominal du CU "Initialiser SoftSonnette" dans le dossier de spécification [SPEC_A1].

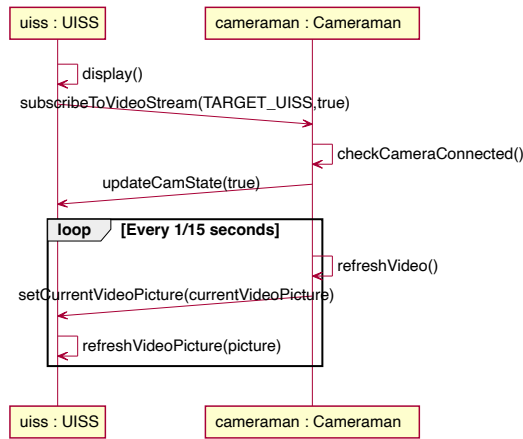


FIGURE 3 – Diagramme de séquence de l'initialisation de SoftSonnette

2.2.3 Se connecter

Ce diagramme représente le scénario nominal du CU "Se connecter" dans le dossier de spécification [SPEC_A1].

Dans ce diagramme de séquence le démonstrateur choisit de réaliser la connexion entre SoftSonnette et AOP.

Pour réaliser cette action le démonstrateur doit renseigner l'adresse IP et le mot de passe de connexion.

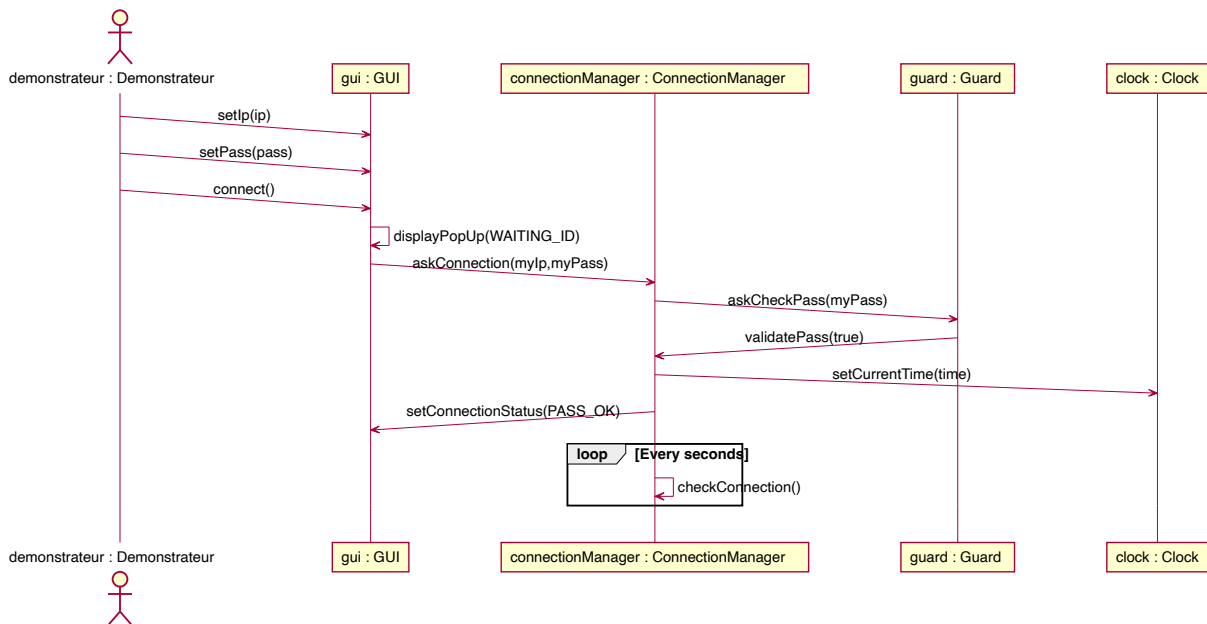


FIGURE 4 – Diagramme de séquence de la connexion entre AOP et SoftSonnette

2.2.4 Demander à entrer

Ce diagramme représente le scénario nominal du CU "Demander à entrer" dans le dossier de spécification [SPEC_A1].

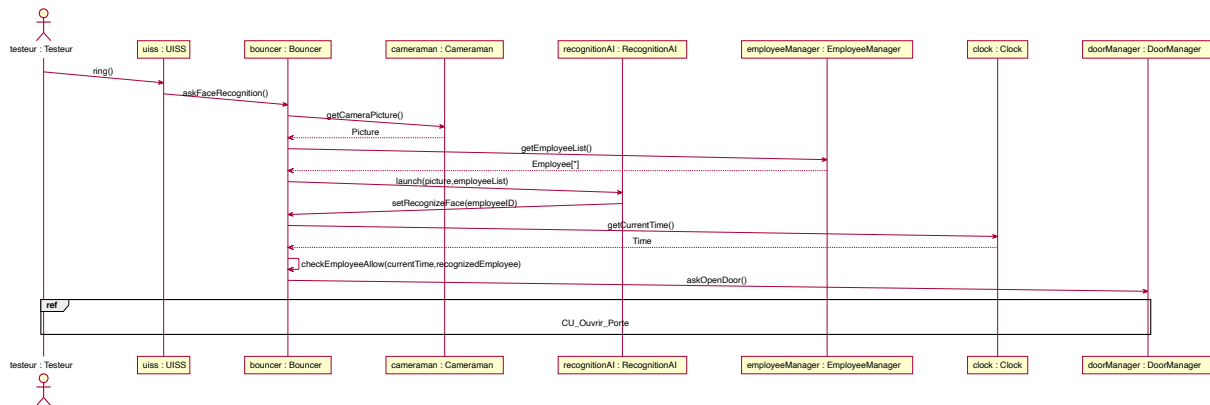


FIGURE 5 – Diagramme de séquence du CU "Demander à entrer"

2.2.5 Ouvrir porte

Ce diagramme représente le scénario nominal du CU "Ouvrir porte" dans le dossier de spécification [SPEC_A1].

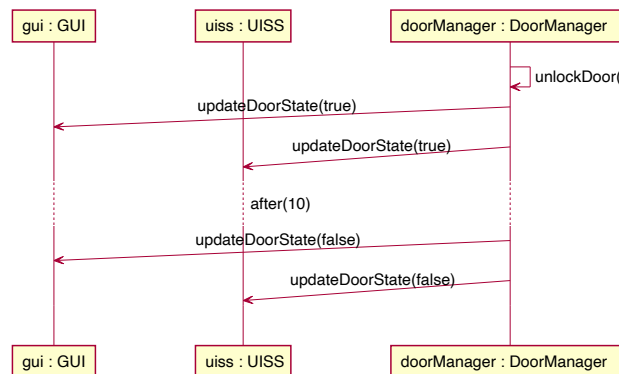


FIGURE 6 – Diagramme de séquence du CU "Ouvrir porte"

2.2.6 Regarder vidéo

Ce diagramme représente le scénario nominal du CU "Regarder vidéo" dans le dossier de spécification [SPEC_A1].

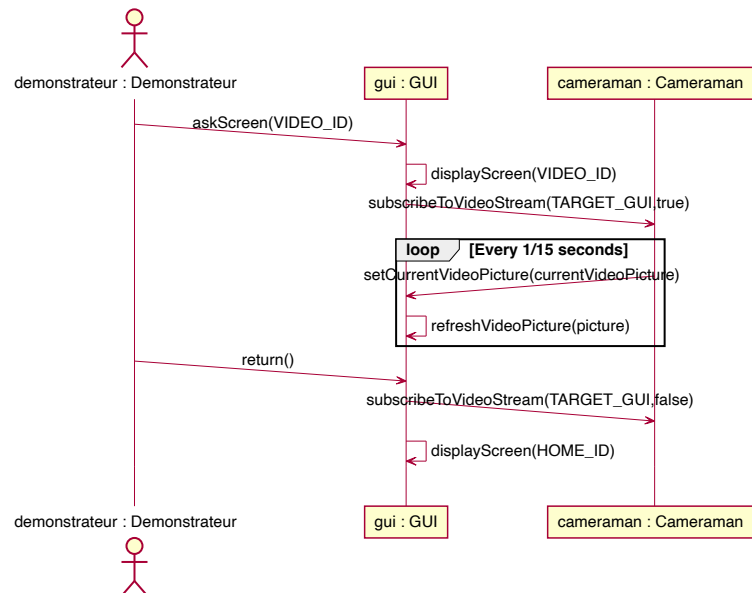


FIGURE 7 – Diagramme de séquence du CU "Regarder vidéo"

2.2.7 Consulter calendrier

Ce diagramme représente le scénario nominal du CU "Consulter calendrier" dans le dossier de spécification [SPEC_A1].

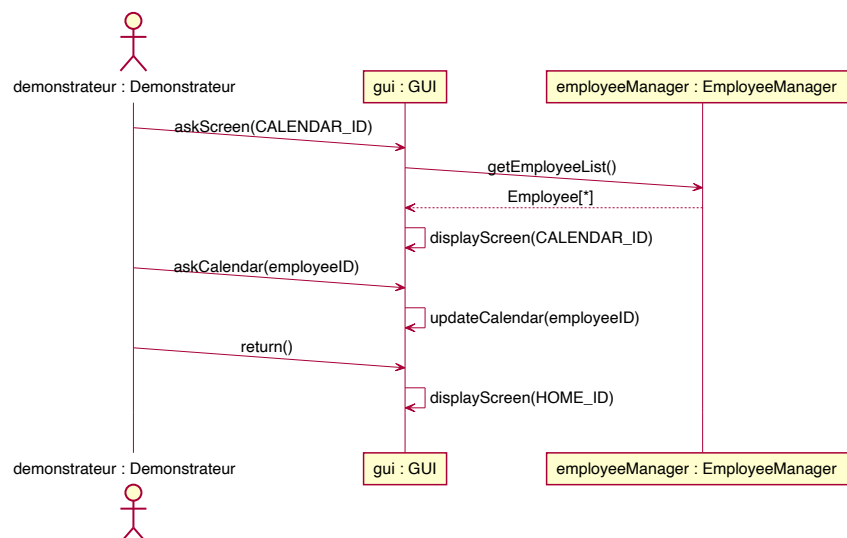


FIGURE 8 – Diagramme de séquence du CU "Consulter calendrier"

2.2.8 Contrôler Porte à distance

Ce diagramme représente le scénario nominal du CU "Contrôle Porte à distance" dans le dossier de spécification [SPEC_A1].

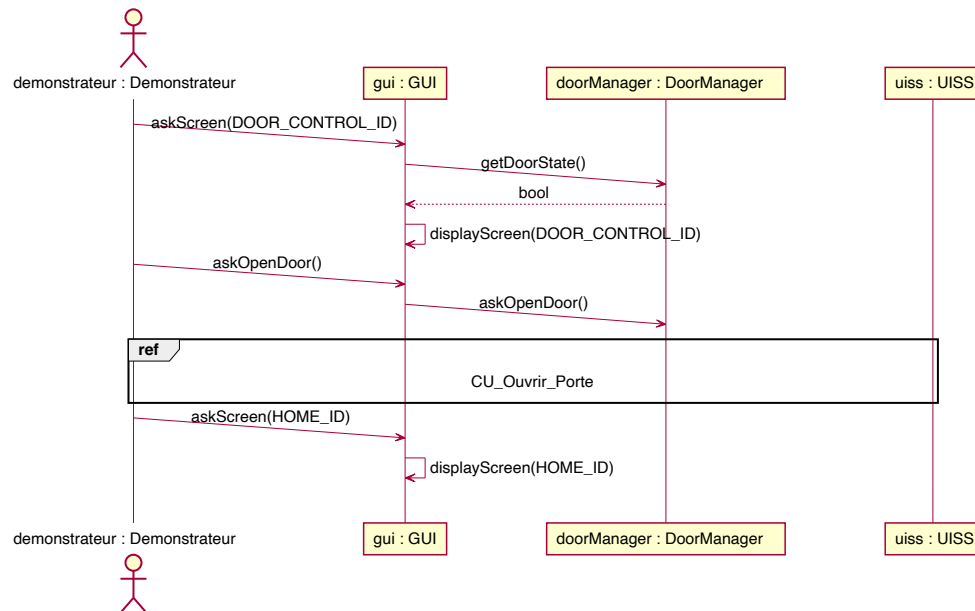


FIGURE 9 – Diagramme de séquence du CU "Contrôle Porte à distance"

2.2.9 Consulter liste employés

Ce diagramme représente le scénario nominal du CU "Consulter liste employés" dans le dossier de spécification [SPEC_A1].

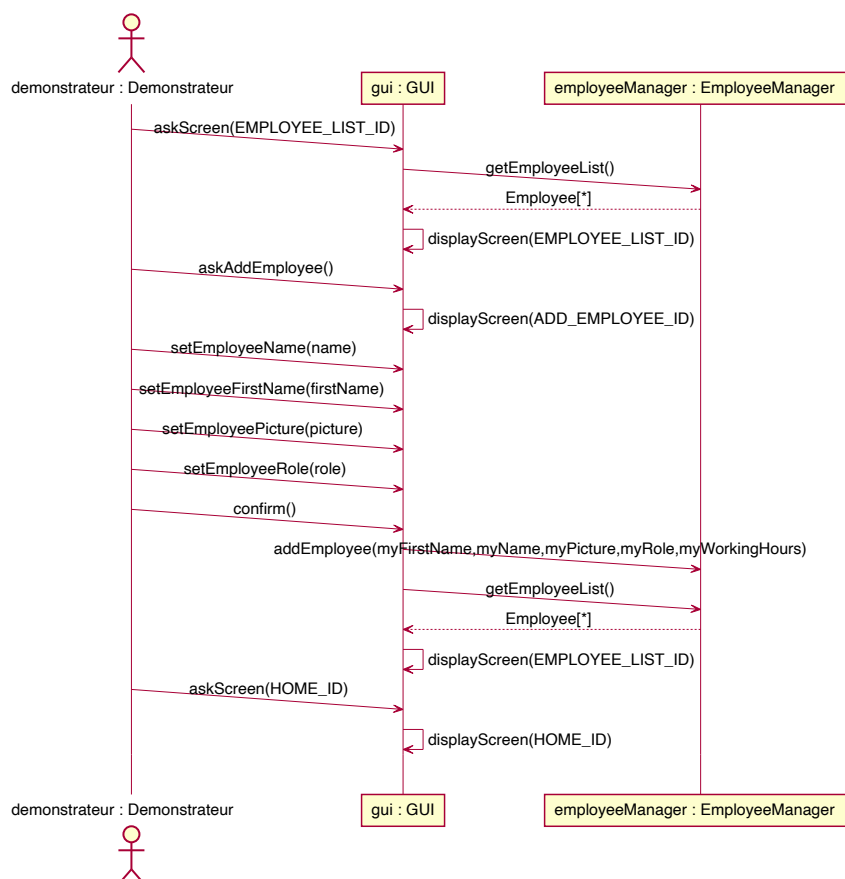


FIGURE 10 – Diagramme de séquence du CU "Consulter liste employés"

2.2.10 Quitter SaE

Ce diagramme représente le scénario nominal du CU "Quitter SaE" dans le dossier de spécification [SPEC_A1].

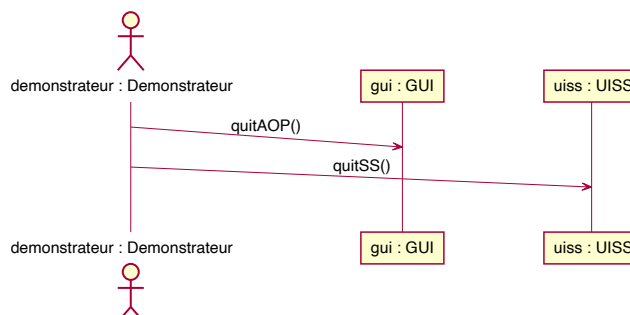


FIGURE 11 – Diagramme de séquence du CU "Quitter SaE"

2.3 Description des composants

2.3.1 Description des types manipulés entre composants

- **ConnectionStatus** : Énumération représentant l'état de la connexion entre AOP et SoftSonnette. ConnectionStatus peut prendre pour valeur : PASS_KO, CONNECT_KO et ALL_OK.
- **Day** : Énumération représentant un jour de la semaine. Day peut prendre pour valeur : MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY et SUNDAY.
- **Employee** : Structure contenant les informations de l'employé. Employee contient : firstName, name, role (Role), id (EmployeeID), workingHours [] [] et picture (chemin local vers l'image). firstName est une chaîne de caractère comprenant entre 2 et 12 caractères alphanumérique encodés en UTF-8. name suit les mêmes règles que firstName à la différence qu'il peut être compris entre 1 et 12 caractères alphanumériques. workingHours [] [] est une matrice de 7 lignes (jours de la semaine) et 2 colonnes (horaires début et fin de journée).
- **EmployeeID** : Un entier qui représente le numéro d'un employé allant jusqu'à MAX_EMPLOYEE qui vaut 10.
- **Hour** : Structure contenant un horaire de début ou de fin de journée où elle est enregistrée au format 24 h avec une précision de 30 minutes. Hour contient heure et minutes.
- **Ip** : Chaîne de caractères de type IPV4 au format X.X.X.X où X est un entier compris entre 0 et 255. Correspond à l'adresse de la board.
- **Password** : Chaîne de caractères composé de 4 chiffres.
- **Picture** : Image du visage d'un employé qui sera utilisée pour la reconnaissance faciale.
- **PopUpID** : Énumération représentant les différentes pop-up affichables sur AOP. PopUpID peut prendre pour valeur : WAITING_ID, ERROR_PASS_ID, ERROR_CONNECT_ID et DELETE_ID.
- **Role** : Énumération représentant les différents types d'employés pour délimiter leurs horaires. Role peut prendre pour valeur : E_MORNING, E_DAY, E_EVENING, E_SECURITY et E_SPECIAL.
- **ScreenID** : Énumération représentant les différents écrans affichables sur AOP. ScreenID peut prendre pour valeur : STARTER_ID, HOME_ID, ADD_EMPLOYEE_ID, ADD_SPECIAL_EMPLOYEE_ID, EMPLOYEE_LIST_ID, DOOR_CONTROL_ID, CALENDAR_ID et VIDEO_ID.
- **Target** : Énumération représentant les différentes cibles pour envoyer la vidéo. Target peut prendre pour valeur : TARGET_GUI et TARGET_UISS.
- **Time** : respecte la norme ISO 8601 : YYYY-MM-DD-HH : MM : SS. Time se met à jour lors de la connexion entre AOP et SoftSonnette.

2.3.2 Diagramme de classe

Représenté ci-dessous le diagramme de classe de PSC, composé de 11 classes avec pour chacune leurs attributs et leurs méthodes décrites dans la suite du document.

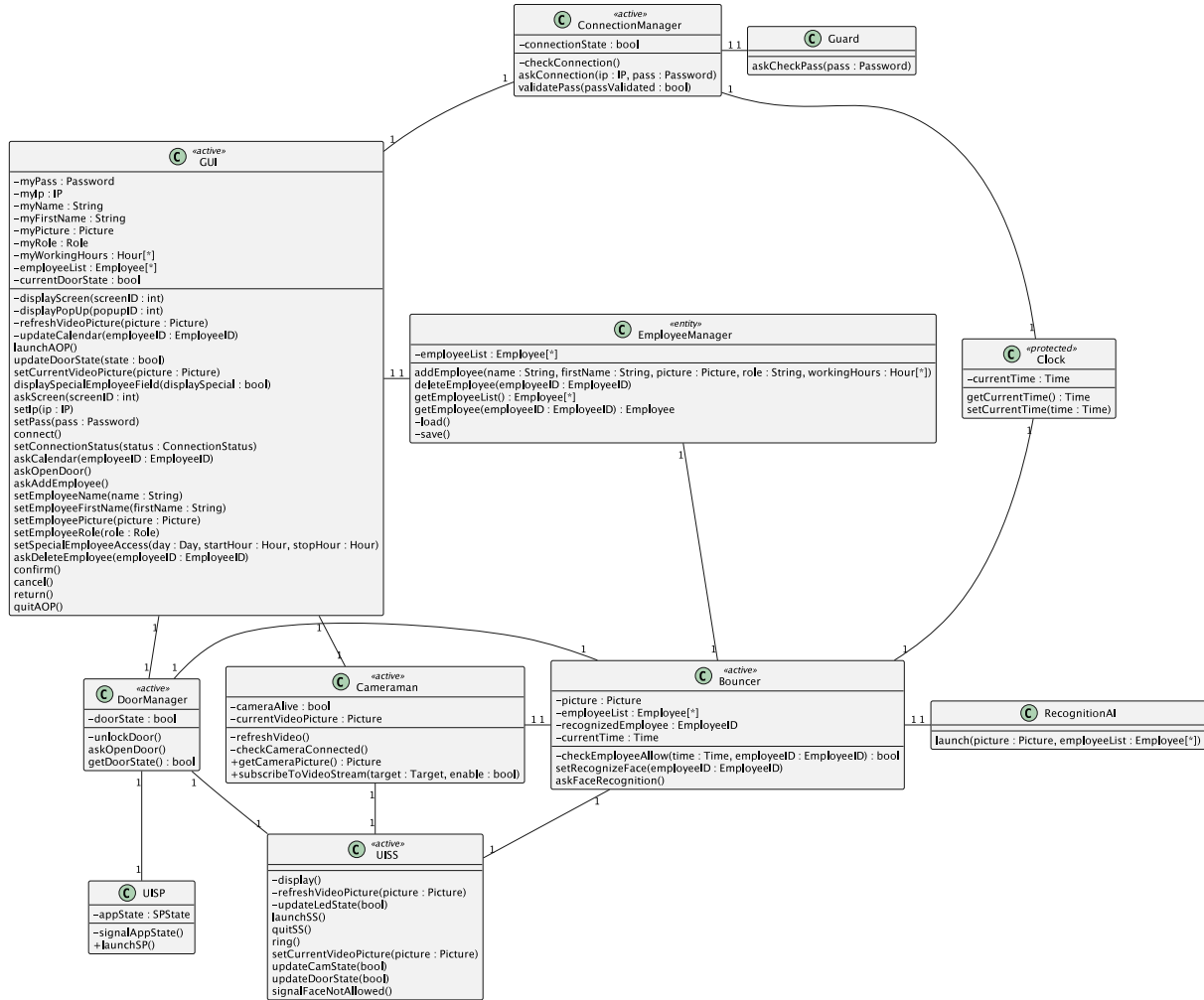


FIGURE 12 – Diagramme de classes du SaE

2.3.3 Description des classes

2.3.3.1 [Object] ConnectionManager

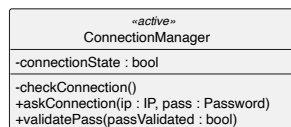


FIGURE 13 – Diagramme de classe représentant ConnectionManager

2.3.3.1.1 Philosophie de conception

La classe ConnectionManager permet de transiter les informations de connexion. Elle permet aussi d'assurer que les applications sont connectées et d'agir en conséquence. Cette classe interagit avec Guard, Clock, et GUI.

2.3.3.1.2 Description structurelle

2.3.3.1.2.1 Attributs

- **connectionState : bool** : Exprime l'état de la connexion actuelle.

2.3.3.1.2.2 Services offerts

- **checkConnection() : void** : Vérifie si la connexion est toujours opérationnelle. Cette fonction envoie un ping toutes les secondes afin de savoir si la connexion entre les appareils est toujours opérationnelle. Dans le cas d'une perte de connexion cette fonction informe les objets intéressés.
- **askConnection(ip : IP, pass : Password) : void** : Envoie les informations de connexion vers le Guard.
- **validatePass(passValidated : bool) : void** : Reçoit l'information de connexion qui valide ou non la connexion.

2.3.3.1.3 Description comportementale

La MAE suivante permet d'expliquer le fonctionnement de l'objet actif ConnectionManager :

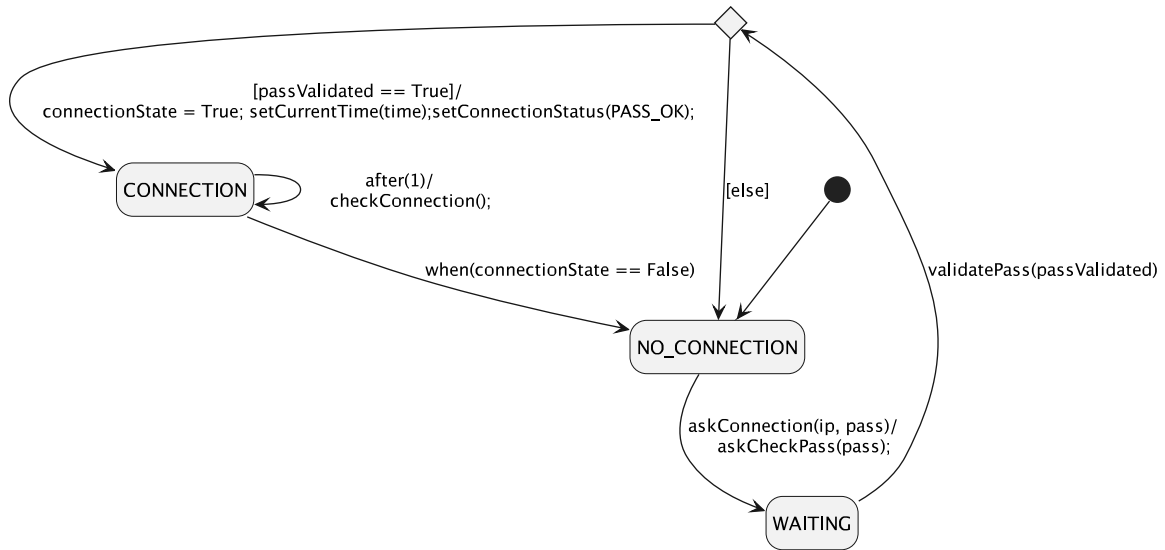


FIGURE 14 – Machine à état représentant ConnectionManager

2.3.3.2 [IHM] GUI



FIGURE 15 – Diagramme de classe représentant GUI

2.3.3.2.1 Philosophie de conception

La classe GUI permet de gérer les interfaces utilisateurs entre le Démonstrateur et AOP en affichant les différentes vues de AOP. Cette classe interagit avec ConnectionManager, DoorManager, EmployeeManager et Cameraman.

2.3.3.2.2 Description structurelle

2.3.3.2.2.1 Attributs

- myPass : Password : Contient le mot de passe entré par le Démonstrateur lors de la connexion.
- myIP : IP : Contient l'adresse IP entrée par le Démonstrateur lors de la connexion.
- myName : String : Contient le nom entré par le Démonstrateur lors de l'ajout d'un employé.
- myFirstName : String : Contient le prénom entré par le Démonstrateur lors de l'ajout d'un employé.
- myPicture : Picture : Contient la photo d'un employé à ajouter.
- myRole : Role : Contient le rôle d'un employé lors de son ajout.
- myWorkingHours : Hour[*] : Contient les horaires de l'employé pour les 7 jours de la semaine.
- employeeList : Employee[*] : liste des employés.
- currentDoorState : bool : état courant de la porte.

2.3.3.2.2 Services offerts

- displayScreen(int screenID) : void : Affiche l'écran entré en paramètre de la fonction.
- displayPopUp(int screenID) : void : Affiche la PopUp entrée en paramètre de la fonction.
- launchAOP() : void : Lance AOP.
- updateCalendar(employeeID : EmployeeID) : void : Met à jour l'affichage du calendrier en fonction de l'ID de l'employé à afficher.
- refreshVideoPicture(picture : Picture) : void : Met à jour la vidéo.
- updateDoorState(state : bool) : void : Modifie l'état de la porte affiché sur AOP grâce au paramètre de la fonction de type booléen.
- displaySpecialEmployeeField(displaySpecial : bool) : void : Affiche les options afin d'ajouter les horaires d'un employé spécial.
- askScreen(screenID : int) : void : Démonstrateur demande à afficher l'écran entré en paramètre de la fonction.
- setIP(ip : IP) : void : Modifie l'attribut myIP de type IP avec le paramètre de la fonction de type IP.
- setPass(pass : Password) : void : Modifie l'attribut myPass de type Password avec le paramètre de la fonction de type Password.
- connect() : void : Demande la connexion.
- setConnectionStatus(status : ConnectionStatus) : void : Vérifie si la connexion est établie via le paramètre de la fonction status de type ConnectionStatus.
- askCalendar(employeeID : EmployeeID) : void : Demande d'afficher le calendrier correspondant au paramètre de la fonction employeeID de type EmployeeID.
- askOpenDoor() : void : Demande l'ouverture de la porte.
- askAddEmployee() : void : Demande a ajouter un employé.
- setEmployeeName(name : String) : void : Modifie l'attribut myName de type String avec le paramètre de la fonction de type String.
- setEmployeeFirstName(firstName : String) : void : Modifie l'attribut myFirstName de type String avec le paramètre de la fonction de type String.
- setEmployeePicture(picture : Picture) : void : Modifie l'attribut myPicture de type Picture avec le paramètre de la fonction de type Picture.
- setEmployeeRole(role : Role) : void : Modifie l'attribut myRole de type Role avec le paramètre de la fonction de type Role.
- setSpecialEmployeeAccess(day : Day, startHour : Hour, stopHour : Hour) : void : Entre les horaires de l'employé spécial. Avec comme paramètres day, startHour et stopHour qui prend en compte le jour, le début et de la fin de la journée de l'employé. Ces paramètres permettent la mise à jour de myWorkingHours.
- setCurrentVideoPicture(picture : Picture) : void : crée la vidéo à partir du flux.
- askDeleteEmployee(employeeID : EmployeeID) : void : Demande la suppression d'un employé. L'employé est déterminé par le paramètre de la fonction.
- confirm() : void : Confirme les modifications faites.
- cancel() : void : Annule les modifications effectuées.
- return() : void : Retourne à l'écran précédent.
- quitAOP() : void : Quitte AOP.

2.3.3.2.3 Description comportementale

Les MAE suivantes permettent d'expliquer le fonctionnement de l'objet actif GUI :

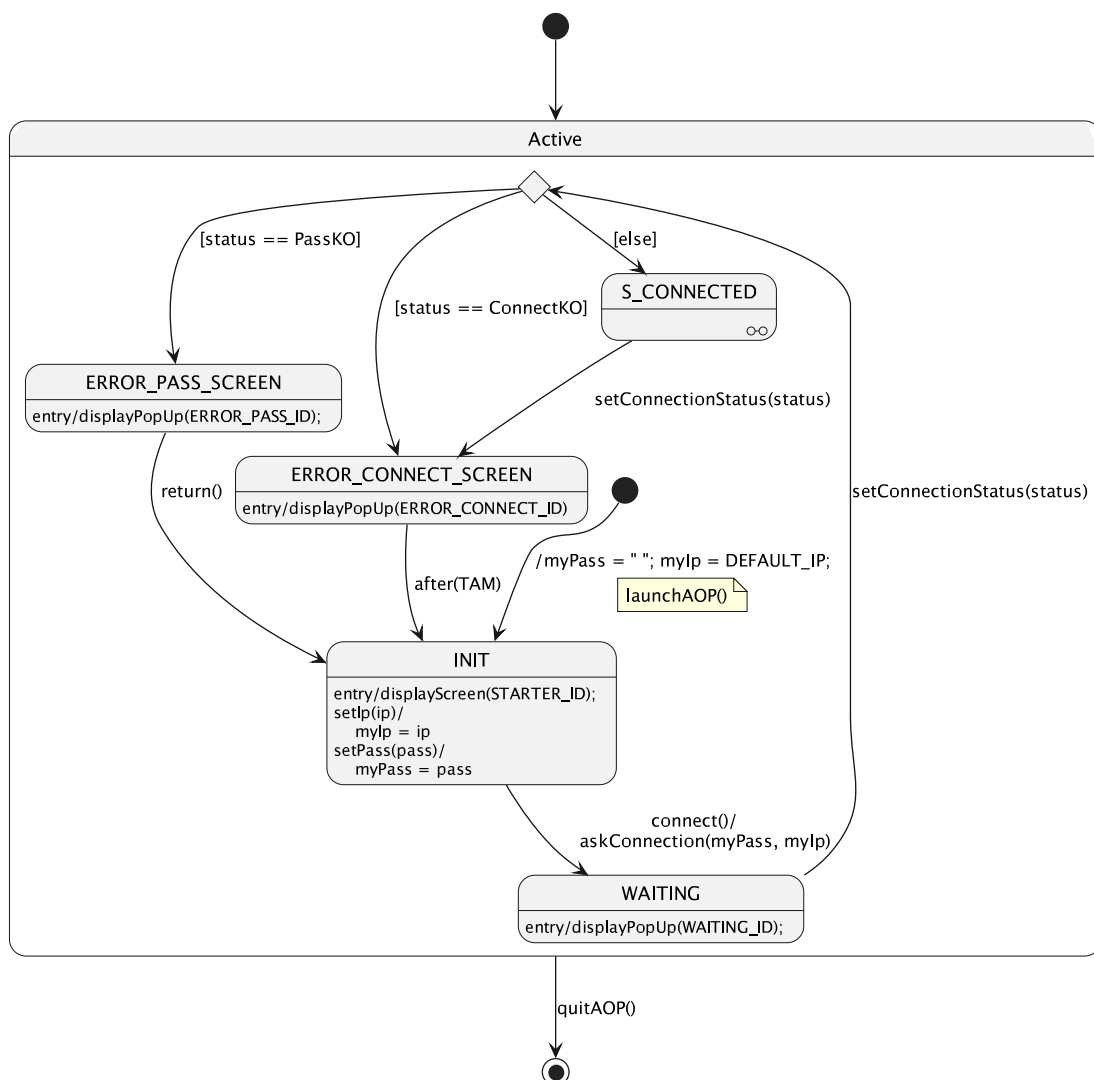


FIGURE 16 – Machine à état représentant GUI

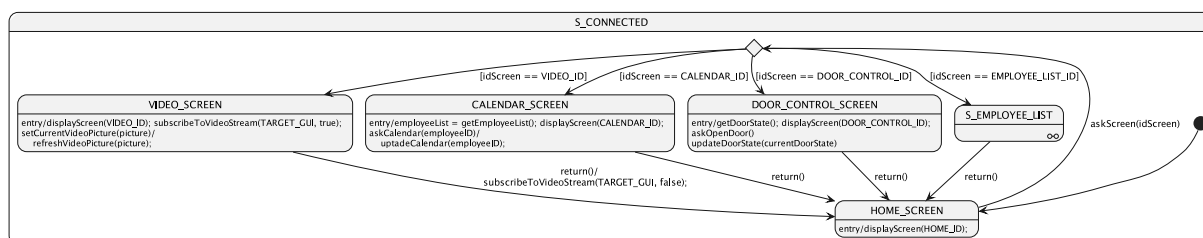


FIGURE 17 – Machine à état représentant GUI

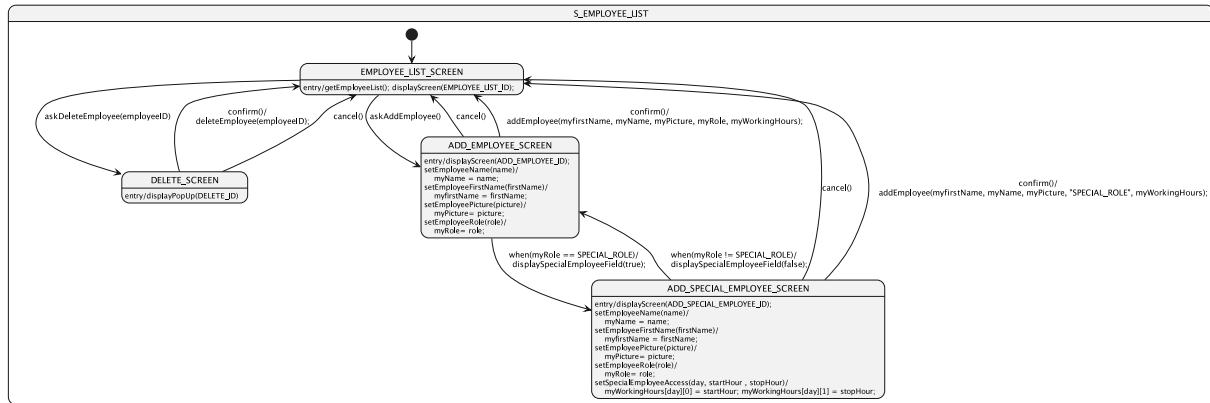


FIGURE 18 – Machine à état représentant employé liste

2.3.3.3 [Object] EmployeeManager

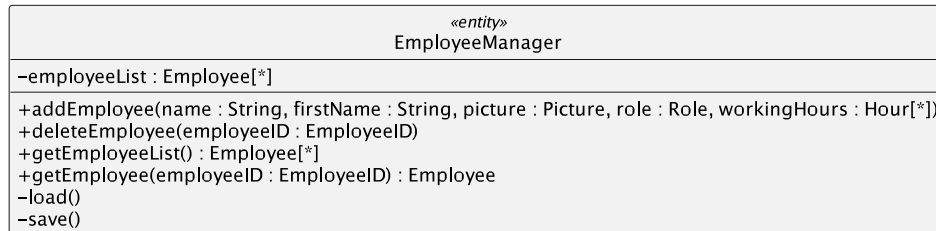


FIGURE 19 – Diagramme de classe représentant EmployeeManager

2.3.3.3.1 Philosophie de conception

EmployeeManager est un objet qui répertorie la liste des employés enregistrés dans l'application. Cet objet gère l'ajout et la suppression d'employés et interagit avec GUI, Bouncer.

2.3.3.3.2 Description structurelle

2.3.3.3.2.1 Attributs

- employeeList : Employee[*] : Contient la liste des employés.

2.3.3.3.2.2 Services offerts

- addEmployee(name : String, firstName : String, picture : Picture, role : String, workingHours : Hour[*]) : void : Ajoute un employé à la liste à partir de son nom, prénom, sa photo, son rôle et les horaires de travail qui lui sont liés.
- deleteEmployee(employeeID : EmployeeID) : void : Supprime l'employé lié à l'identifiant employeeID passé en paramètre.
- getEmployeeList() : Employee[*] : Renvoie la liste des employés.
- getEmployee(employeeID : EmployeeID) : Employee : Retourne l'objet employé lié à l'identifiant employeeID passé en paramètre.
- load() : void : Charge les données persistantes des informations employés.
- save() : void : Sauvegarde les données persistantes après un changement.

2.3.3.4 [Object] Clock

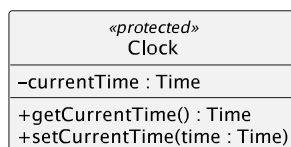


FIGURE 20 – Diagramme de classe représentant Clock

2.3.3.4.1 Philosophie de conception

Clock est l'objet représentant l'heure. Cet objet est utilisé afin de synchroniser les heures entre l'AOP et SoftSonnette. Il est aussi utilisé pour savoir si un employé peut entrer en fonction de ses horaires attribués.

2.3.3.4.2 Description structurelle

2.3.3.4.2.1 Attributs

- currentTime : Time : Contient le temps courant.

2.3.3.4.2.2 Services offerts

- getCurrentTime() : Time : Permet de donner l'heure courante pour la comparaison des heures dans le bouncer.
- setCurrentTime(time : Time) : void : Permet de modifier l'heure courante, ce qui est utilisé lors de la connexion avec l'AOP.

2.3.3.5 [Object] Cameraman

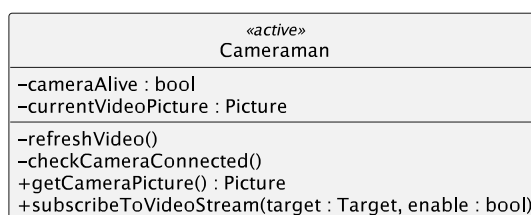


FIGURE 21 – Diagramme de classe de Cameraman

2.3.3.5.1 Philosophie de conception

Cameraman est l'objet représentant la webcam. Celui-ci sera utilisé pour prendre une image du testeur afin qu'il se fasse reconnaître, ou pour prendre un flux vidéo qui sera retransmis sur l'écran de SoftSonnette ou sur l'AOP.

2.3.3.5.2 Description structurelle

2.3.3.5.2.1 Attributs

- cameraAlive : bool : Indique si la caméra est connectée ou non.
- currentVideoPicture : Picture : Image courante qui permet de créer la vidéo.

2.3.3.5.2.2 Services offerts

- refreshVideo() : void : Met à jour la vidéo sur les target ayant fait un subscribe. La mise à jour se fait image par image.
- checkCameraConnected() : void : Vérifie la présence de la caméra.
- getCameraPicture() : Picture : Envoie une photo de ce que voit Cameraman. Cette photo pourra être utilisée pour reconnaître le visage d'une personne.
- subscribeToVideoStream(target : Target, enable : bool) : void : Permet de demander à recevoir le flux vidéo en provenance de Cameraman.

2.3.3.6 [Object] Guard

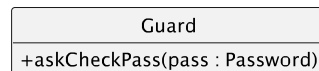


FIGURE 22 – Diagramme de classe de Guard

2.3.3.6.1 Philosophie de conception

La classe Guard est le gardien du mot de passe permettant à AOP d'interagir avec SoftSonnette. Il vérifie que le mot de passe reçu est celui codé en dur dans l'application puis autorise la connexion si ce dernier est correct.

2.3.3.6.2 Description structurelle

2.3.3.6.2.1 Attributs

N.A

2.3.3.6.2.2 Services offerts

- askCheckPass(pass : Password) : void : Permet de demander à Guard de vérifier le mot de passe passé en paramètre

2.3.3.7 [Object] Bouncer

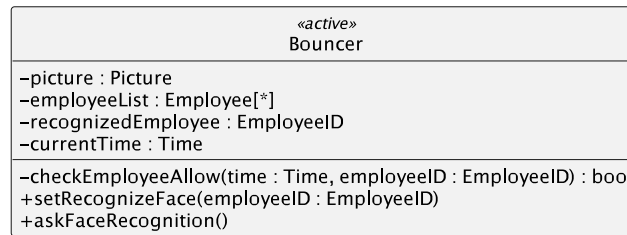


FIGURE 23 – Diagramme de classe représentant Bouncer

2.3.3.7.1 Philosophie de conception

La classe Bouncer permet au SàE d'autoriser ou non une personne à entrer. En interaction avec Cameraman elle déclenche la prise d'une photo de la personne se présentant à la sonnette. Elle interagit également avec EmployeeManager pour récupérer la liste des employés. Elle déclenche la reconnaissance faciale avec la classe RecognitionAI à qui elle envoie toutes les données nécessaires. Elle vérifie qu'un employé reconnu ait l'autorisation d'entrer sur l'horaire courant qu'elle récupère auprès de la classe Clock. Elle demande enfin l'ouverture de la porte à DoorManager ou signal à UISS que la personne n'est pas reconnu.

2.3.3.7.2 Description structurelle

2.3.3.7.2.1 Attributs

- picture : Picture : photo de la personne à reconnaître.
- employeeList : Employee[*] : liste courante des employés.
- currentTime : Time : temps utilisé pour vérifier que l'employé est autorisé à entrer.
- recognizedEmployee : EmployeeID : id de l'employé reconnu par RecognitionAI.

2.3.3.7.2.2 Services offerts

- checkEmployeeAllow(time : Time, employeeID : EmployeeID) : bool : À partir du temps courant et de l'ID de l'employé, elle vérifie si l'employé est autorisé à entrer à l'horaire courant.
- setRecognizeFace(employeeID : EmployeeID) : void : Retourne l'employé reconnu par l'IA, ou NULL s'il n'y a pas de correspondance.
- askFaceRecognition() : void : Trigger permettant de déclencher le processus reconnaissance facial et d'autorisation d'entrée.

2.3.3.7.3 Description comportementale

La MAE suivante permet d'expliquer le fonctionnement de l'objet actif Bouncer :

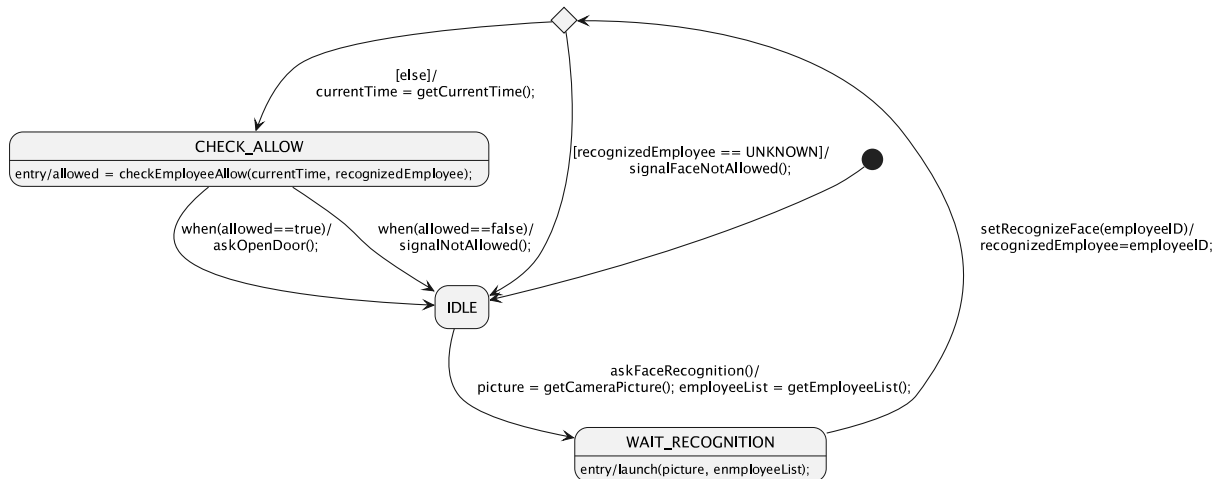


FIGURE 24 – Machine à États de Bouncer

2.3.3.8 [IHM] UISS

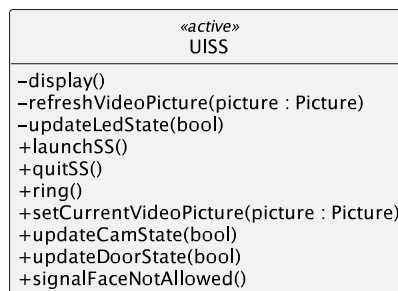


FIGURE 25 – Diagramme de classe d'UISS

2.3.3.8.1 Philosophie de conception

La classe UISS s'occupe de la gestion de l'interface utilisateur entre le Testeur et SoftSonnette. Cela permet l'affichage de la vidéo, l'état de la porte mais aussi au Testeur de sonner.

2.3.3.8.2 Description structurelle

2.3.3.8.2.1 Attributs

N.A

2.3.3.8.2.2 Services offerts

- display() : void : Affiche l'écran de SoftSonnette.
- refreshVideoPicture(picture : Picture) : void : Met à jour la vidéo de SoftSonnette.
- updateLedState(bool) : void : Met à jour sur la Board la LED signalant un visage non reconnu.
- launchSS() : void : Démarre l'application SoftSonnette.

- quitSS() : void : Ferme l'application SoftSonnette.
- ring() : void : Testeur sonne.
- setCurrentVideoPicture(Picture) : void : Met à jour l'image de la vidéo courante.
- updateCamState(bool) : void : Met à jour la présence de la caméra sur l'écran.
- updateDoorState(bool) : void : DoorManager met à jour l'état de la Porte sur l'écran.
- signalFaceNotAllowed() : void : Bouncer signale que le visage est non reconnu.

2.3.3.8.3 Description comportementale

La MAE suivante permet d'expliquer le fonctionnement de l'objet actif UISS.

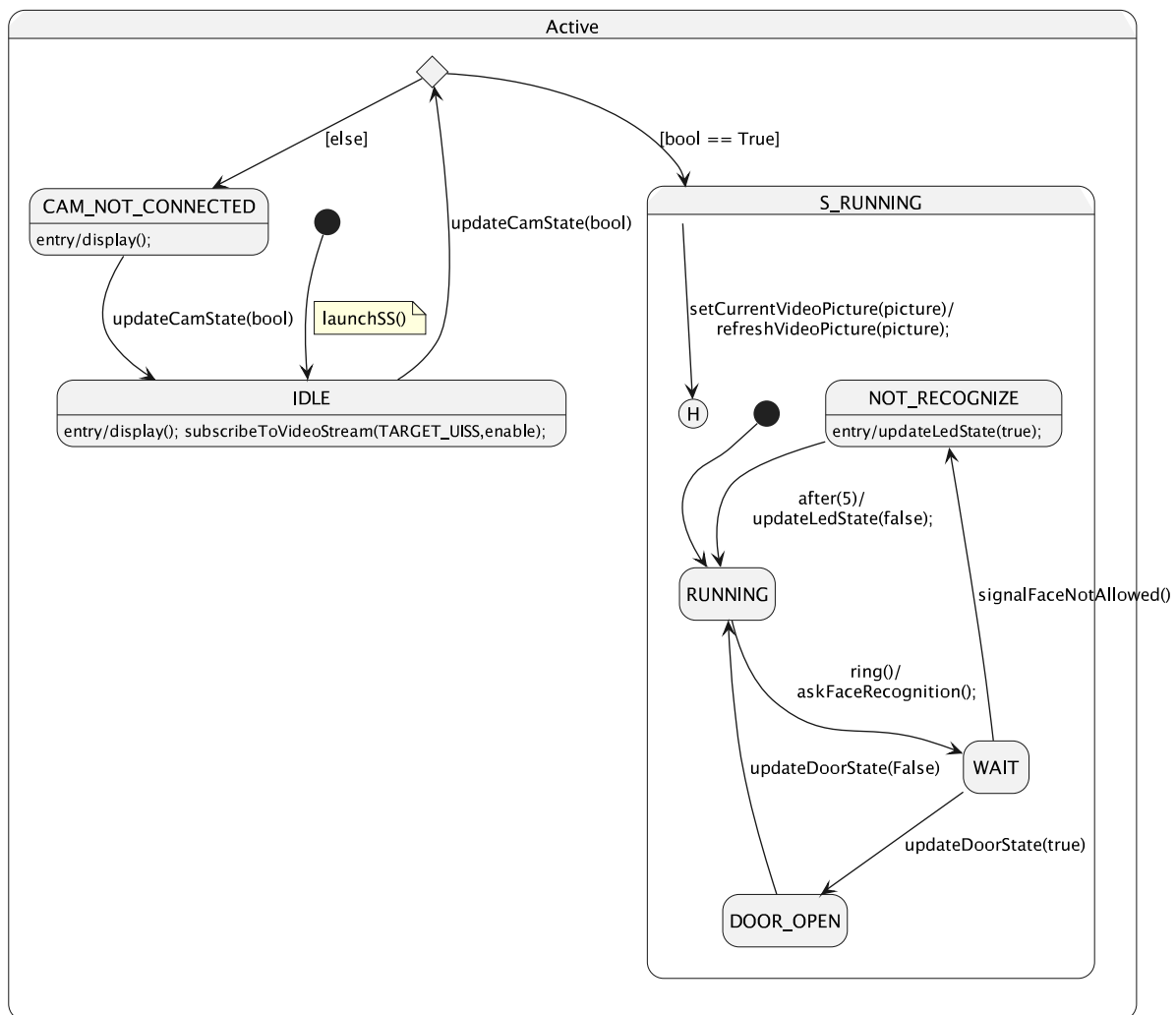


FIGURE 26 – Machine à États d'UISS

2.3.3.9 [Object] DoorManager

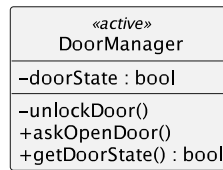


FIGURE 27 – Diagramme de classe représentant DoorManager

2.3.3.9.1 Philosophie de conception

La classe DoorManager est l'objet frontière entre GUI, UISS et UISP. Elle permet de commander le déverrouillage de la porte, et de fournir l'état de la porte.

2.3.3.9.2 Description structurelle

2.3.3.9.2.1 Attributs

- doorState : bool : Indique l'état de la porte (ouvert/fermé).

2.3.3.9.2.2 Services offerts

- unlockDoor() : void : Déverrouille la porte pendant 10 secondes.
- askOpenDoor() : void : Envoie la demande d'ouverture de la porte vers UISP.
- getDoorState() : bool : Retourne l'état actuel de la porte (ouverte ou fermée).

2.3.3.9.3 Description comportementale

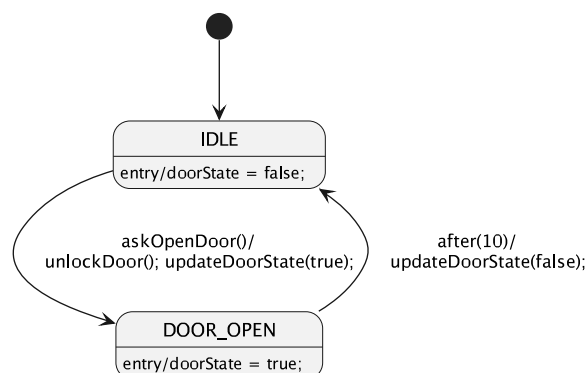


FIGURE 28 – Machine à état représentant DoorManager

2.3.3.10 [IHM] UISP

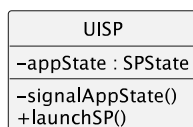


FIGURE 29 – Diagramme de classe d'UISP

2.3.3.10.1 Philosophie de conception

La classe UISP permet de mettre à jour la Led de la Board indiquant au Démonstrateur l'état de l'application.

2.3.3.10.2 Description structurelle

2.3.3.10.2.1 Attributs

- appState : SPState : Indique l'état de SoftPorte (on/off).

2.3.3.10.2.2 Services offerts

- signalAppState() : void : Signale le lancement de l'application en allumant la Led.
- launchSP() : void : Lance SoftPorte.

2.3.3.11 [Object] RecognitionAI

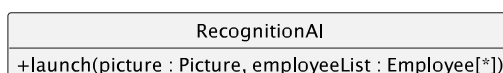


FIGURE 30 – Diagramme de classe de RecognitionAI

2.3.3.11.1 Philosophie de conception

RecognitionAI est l'objet contenant l'algorithme de reconnaissance faciale. En lui fournissant la photo d'une personne, celui-ci pourra déterminer si la personne en question est autorisée à entrer. Cet objet sera appelé par Bouncer lorsqu'une demande d'entrée est annoncée. Une fois la reconnaissance faciale terminée, RecognitionAI envoie à Bouncer le résultat.

2.3.3.11.2 Description structurelle

2.3.3.11.2.1 Attributs

N.A.

2.3.3.11.2.2 Services offerts

- launch(picture : Picture, employeeList : Employee[*]) : void : Lance le processus de reconnaissance faciale.

3 Dictionnaire du domaine

A

AOP (Application Ouverture Porte)

AOP est l'application Android utilisée par le Démonstrateur comme interface utilisateur pour gérer les différentes fonctionnalités du PSC. Elle comprend 4 menus distincts :

- Employés : Consulter la liste des employés, ajouter ou supprimer des employés.
- Calendrier : Consulter le calendrier d'un employé sélectionné.
- Vidéo : Afficher le flux vidéo provenant de la webcam.
- Contrôle de la porte : Contrôler la porte à distance et consulter l'historique des activités.

B

Board

Est désigné par le terme Board la STM32MP15 formant l'ensemble Microcontrôleur + Microprocesseur.

E

employeID

Correspond à l'identifiant de l'employé dans la base de données. Il s'agit d'un entier compris entre 1 et MAX_EMPLOYE qui vaut 10.

Employé

Un employé est équivalent à un Testeur dans le cadre du prototype PSC. Il s'agit d'une personne susceptible de se présenter devant le PSC pour entrer, et dont les informations peuvent être entrées dans l'application AOP.

F

Flux vidéo

Le flux vidéo est généré par la webcam (E_Caméra), qui peut filmer avec une résolution de 1280x720p. Une résolution plus faible sera préférée pour l'envoi du flux, cette résolution sera définie en conception.

H

Horaires

Les horaires d'accès pour déclencher l'ouverture de la porte sont sous la forme d'un bloc horaire unique pour chaque journée. Les horaires sont regroupés en créneaux de 30 min et le format d'affichage est le format 24h. Ces horaires se basent sur l'heure de la Board, qui est équipée d'une RTC.

I

Informations de connexion

Les informations de connexion correspondent à l'adresse IP de la Board et au mot de passe requis pour entrer dans AOP.

Informations testeurs

Les informations liées aux testeurs sont : le prénom (celui-ci comprend entre 2 et 12 caractères alphanumériques), le nom (celui-ci comprend entre 1 et 12 caractères alphanumériques), une photo, le rôle et pour les employés spéciaux les horaires d'accès. La convention d'encodage est l'UTF-8.

IP

L'adresse IP permettant la connexion entre AOP et SoftSonnette. Cette adresse est représentée sous la forme IPv4 "x.x.x.x" où x représente un entier allant de 0 à 255

L**Linux**

Fait référence à l'OS OpenSTLinux (OSTL) version 4.1 (Kernel Linux : 5.15) déployé sur le Microprocesseur.

P**Paramètres photo**

Les paramètres de la photo seront définis en conception.

Password

Le mot de passe permettant au Démonstrateur de se connecter à AOP. Celui-ci est composé de 4 chiffres.

popupID

Correspond à l'identifiant de la PopUP sur l'application AOP. Il s'agit d'un entier compris entre 1 et 4.

PSC (Prototype Sonnette Connectée)

PSC est le nom donné au projet qui comprend SoftSonnette, SoftPorte et AOP.

R**Rôle**

Le rôle d'un employé définit sa plage horaire d'accès. Il existe cinq rôles différents :

- Employé matin : 3h-13h, du lundi au vendredi.
- Employé journée : 8h-20h, du lundi au vendredi.
- Employé soir : 13h-23h, du lundi au vendredi.
- Employé sécurité : Accès illimité (24/7).
- Employé spécial : Les horaires sont définis manuellement pour chaque jour lors de l'ajout de l'employé dans l'application.

S**screenID**

Correspond à l'identifiant de l'écran sur l'application AOP. Il s'agit d'un entier compris entre 1 et 8.

Synchronisation

La synchronisation correspond au processus dans lequel AOP importe la base de données présente sur SoftSonnette, sur le Téléphone.

SàE correctement installé

Pour une installation correcte il faut les éléments suivants :

- AOP installé sur le téléphone.
- Linux installé sur le Microprocesseur.
- SoftSonnette installé sur Linux.
- SoftPorte déployé sur le Microcontrôleur.

T**TAM**

Le Temps d’Affichage d’un Message correspond à la durée pendant laquelle la fenêtre PopUP_Erreur_Connexion s’affiche avant de se fermer. Cette durée sera fixée en conception.

4 Table des figures

1	Architecture candidate	10
2	Diagramme de séquence du scénario nominal	12
3	Diagramme de séquence de l'initialisation de SoftSonnette	13
4	Diagramme de séquence de la connexion entre AOP et SoftSonnette	13
5	Diagramme de séquence du CU "Demander à entrer"	14
6	Diagramme de séquence du CU "Ouvrir porte"	14
7	Diagramme de séquence du CU "Regarder vidéo"	15
8	Diagramme de séquence du CU "Consulter calendrier"	15
9	Diagramme de séquence du CU "Contrôle Porte à distance"	16
10	Diagramme de séquence du CU "Consulter liste employés"	17
11	Diagramme de séquence du CU "Quitter SaE"	17
12	Diagramme de classes du SaE	19
13	Diagramme de classe représentant ConnectionManager	20
14	Machine à état représentant ConnectionManager	21
15	Diagramme de classe représentant GUI	21
16	Machine à état représentant GUI	24
17	Machine à état représentant GUI	24
18	Machine à état représentant employé liste	25
19	Diagramme de classe représentant EmployeeManager	25
20	Diagramme de classe représentant Clock	26
21	Diagramme de classe de Cameraman	26
22	Diagramme de classe de Guard	27
23	Diagramme de classe représentant Bouncer	28
24	Machine à États de Bouncer	29
25	Diagramme de classe d'UISS	29
26	Machine à États d'UISS	30
27	Diagramme de classe représentant DoorManager	31
28	Machine à état représentant DoorManager	31
29	Diagramme de classe d'UIISP	32
30	Diagramme de classe de RecognitionAI	32

