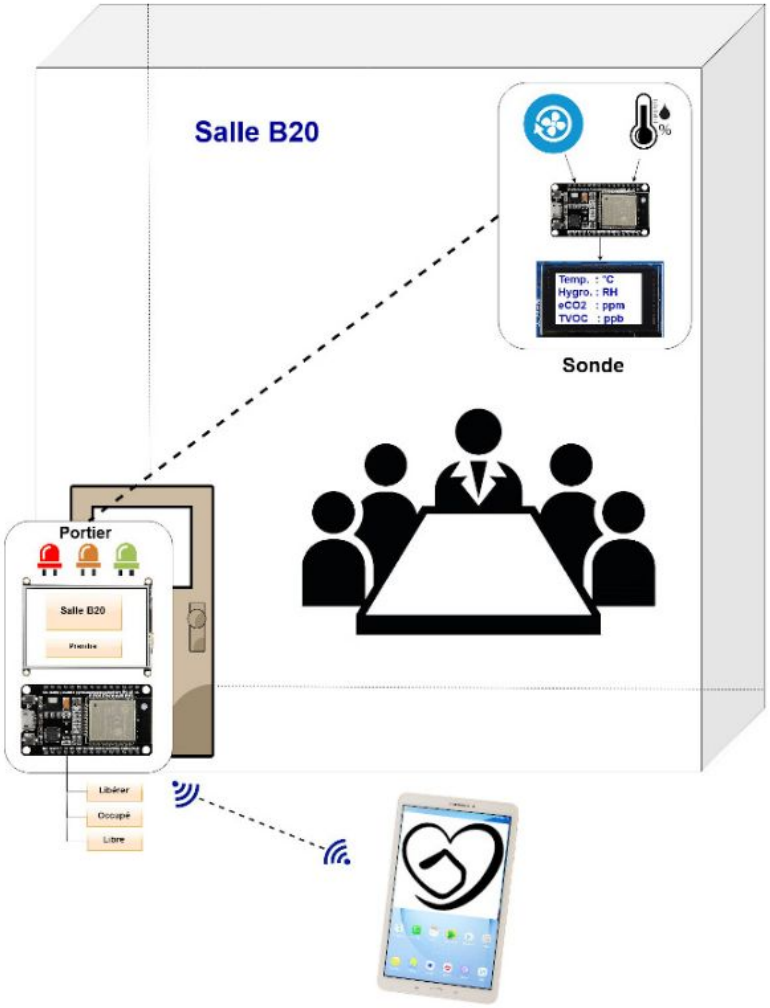
meeting

Projet BTS SN-IR
La Salle Avignon 2020

Dossier v0.1
Revue 2

Présentation générale

Placé à l'extérieur d'une pièce (salle de réunion ou de travail, …), le système permettra d'accéder en temps réel aux informations de l'espace concerné : il affichera la disponibilité et l'état de confort et permettra de réaliser sa réservation.
Tout en s'intégrant facilement à l'environnement, il résout le manque d'interface entre les utilisateurs et les salles de réunion en permettant de travailler plus efficacement.
L'objectif est de proposer une solution simple, alliant flexibilité et ergonomie. Le système affiche de la disponibilité d'accès et le niveau de confort d'une salle de travail ou de réunion.
L'indice de confort est une donnée extrêmement subjective. Elle permet d'apprécier un "ressenti" de confort par rapport à une situation météorologique.



Analyse de l’existant

Des solutions à base d’écran tactile existent mais sont souvent onéreuses car basées sur des modèles de type “tablette”. D’autre part, les systèmes existants sont orientés dans la gestion des salles de réunions et ne permettent pas de gérer les bureaux personnels.
La gestion des salles de réunion est une problématique complexe et inhérente à chaque entreprise. De nombreux paramètres sont à prendre en compte afin d’exploiter au Lycée La Salle Avignon Page 1/10 Session 2020 BTS SN E62 Projet technique : Meeting Aix-Marseille mieux le parc de salles, et faciliter le transfert d’informations liées à l’organisation de réunions de travail.

- Mettre en avant les caractéristiques des salles mises à disposition : ces informations permettront une réservation d’un espace correspondant au besoin.
- Afficher la disponibilité et le planning d'une salle de réunion : ces informations permettent aux utilisateurs de vérifier d’un coup d’œil quel espace est disponible pour une réunion imprévue, et d’effectuer la réservation sur le moment.
- Lutter efficacement contre les réunions fantôme : ce phénomène fait également partie des problématiques rencontrées par les gestionnaires de salles. Les cas d’espaces réservés, mais finalement non occupés perturbent la gestion du parc de salles.
- Gagner du temps dans la recherche d’une salle ou d’un espace de travail, tout en évitant les interruptions de réunions : permettre aux utilisateurs de trouver et de réserver facilement une salle, directement sur l’écran, ou bien à distance, depuis un poste de travail ou un smartphone.

Expression du besoin

L'objectif est de proposer une solution simple, alliant flexibilité et ergonomie. Le système affiche de la disponibilité d'accès et le niveau de confort d'une salle de travail ou de réunion. L'indice de confort est une donnée extrêmement subjective. Elle permet d'apprécier un "ressenti" de confort par rapport à une situation météorologique. Le confort thermique est traditionnellement lié à 6 paramètres :

- Le métabolisme, qui est la production de chaleur interne au corps humain permettant de maintenir celui-ci autour de 36,7°C. Un métabolisme de travail correspondant à une activité particulière s'ajoute au métabolisme de base du corps au repos.
- L'habillement, qui représente une **résistance thermique** aux échanges de chaleur entre la surface de la peau et l'environnement.
- La **température ambiante** de l'air Ta.
- La température moyenne des parois Tp.
- L'**humidité relative de l'air (HR)**, qui est le rapport exprimé en pourcentage entre la quantité d'eau contenue dans l'air à la température ta et la quantité maximale d'eau contenue à la même température.
- La vitesse de l'air, qui influence les échanges de chaleur par convection. Dans le bâtiment, les vitesses de l'air ne dépassent généralement pas 0,2 m/s.

* Échelle de jugement subjectif de Fanger

Niveau de confort thermique	
+3	Chaud
+2	Tiède
+1	Légèrement tiède
0	Neutre
-1	Légèrement frais
-2	Frais
-3	Froid

Présentation du projet

Le système devra :

- Permettre de voir les salles de réunion (sur l'application mobile)
- Prendre ou libérer une salle de réunion (sur l'application mobile ou sur le portier)
- Visualiser des informations concernant la salle (température / niveau de confort)
- Rechercher une salle de réunion

Le système sera équipé de :

- D'une sonde composée d'un capteur de température, d'un capteur d'humidité et d'un capteur de qualité d'air. La sonde sera dans la salle de réunion donc il y aura autant de sonde que de salle de réunion
- D'un portier composé d'un micro-contrôleur (ESP32), un écran tactile, d'indicateurs lumineux (LED). Le portier sera positionné à côté de la partie donc il y aura autant de portier que de salle de réunion.
- D'une application mobile

Répartition des tâches

Etudiants en charges du projet

Option EC :

- Etudiant 1 : Alexandre Mariette
- Etudiant 2 : Nathan Joubert

Option IR :

- Etudiant 3 : Vincent Devine

Répartition des tâches

Etudiant 3 - DEVINE Vincent

- Rechercher une salle
- Afficher les données (informations sur la salle, disponibilité, mesures, indice de confort) d'une salle
- Prendre/Libérer une salle
- Communiquer avec le portier
- Gérer les favoris
- Éditer les informations associées à une salle

Partie Etudiant 3 : DEVINE Vincent

Objectifs

On doit créer une application mobile permettant de visualiser les salles de réunion, voir les informations détaillées d'une salle (nom, emplacement, description, niveau de confort, température, surface, disponibilité de la salle).
De plus, cette application de permettre de prendre, libérer et configurer une salle de réunion.

Ressources Logicielles

Fonctionnalité	Elément utilisé	Version
Système d'exploitation du poste de développement	GNU/Linux Ubuntu	16.04
Logiciel de planification	Trello + Trello Gantt	(applications web)
Génération des diagrammes	BOUML	7.9.1
Gestion de version	Subversion	1.9.3
Environnement de développement	Android Studio	3.6.2
Génération de la documentation	Doxygen	1.8.11
Tablette test	Samsung	Android 7.0 API 24

Diagramme de déploiement

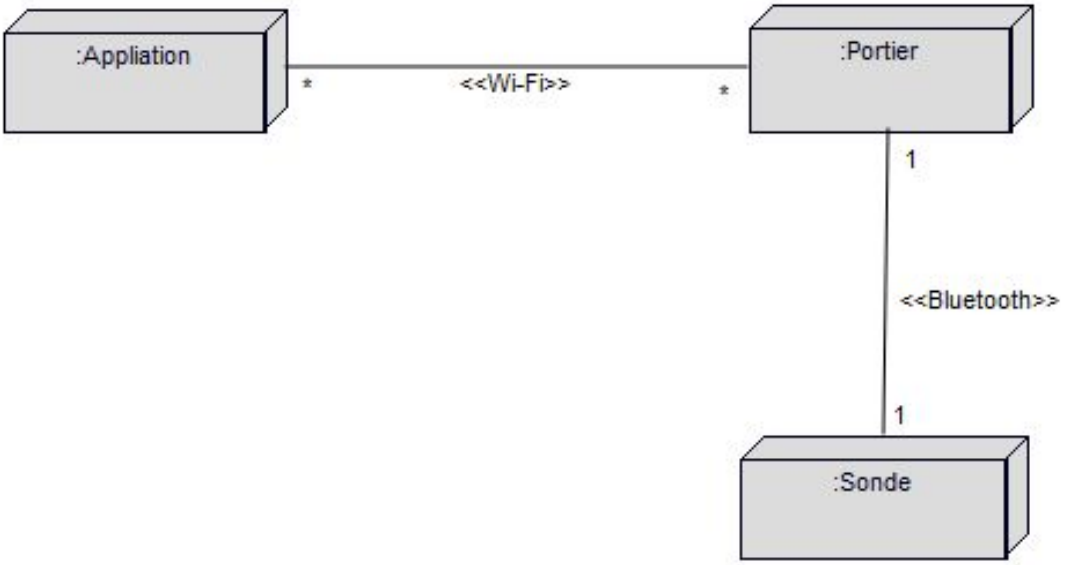


Diagramme des cas d'utilisation

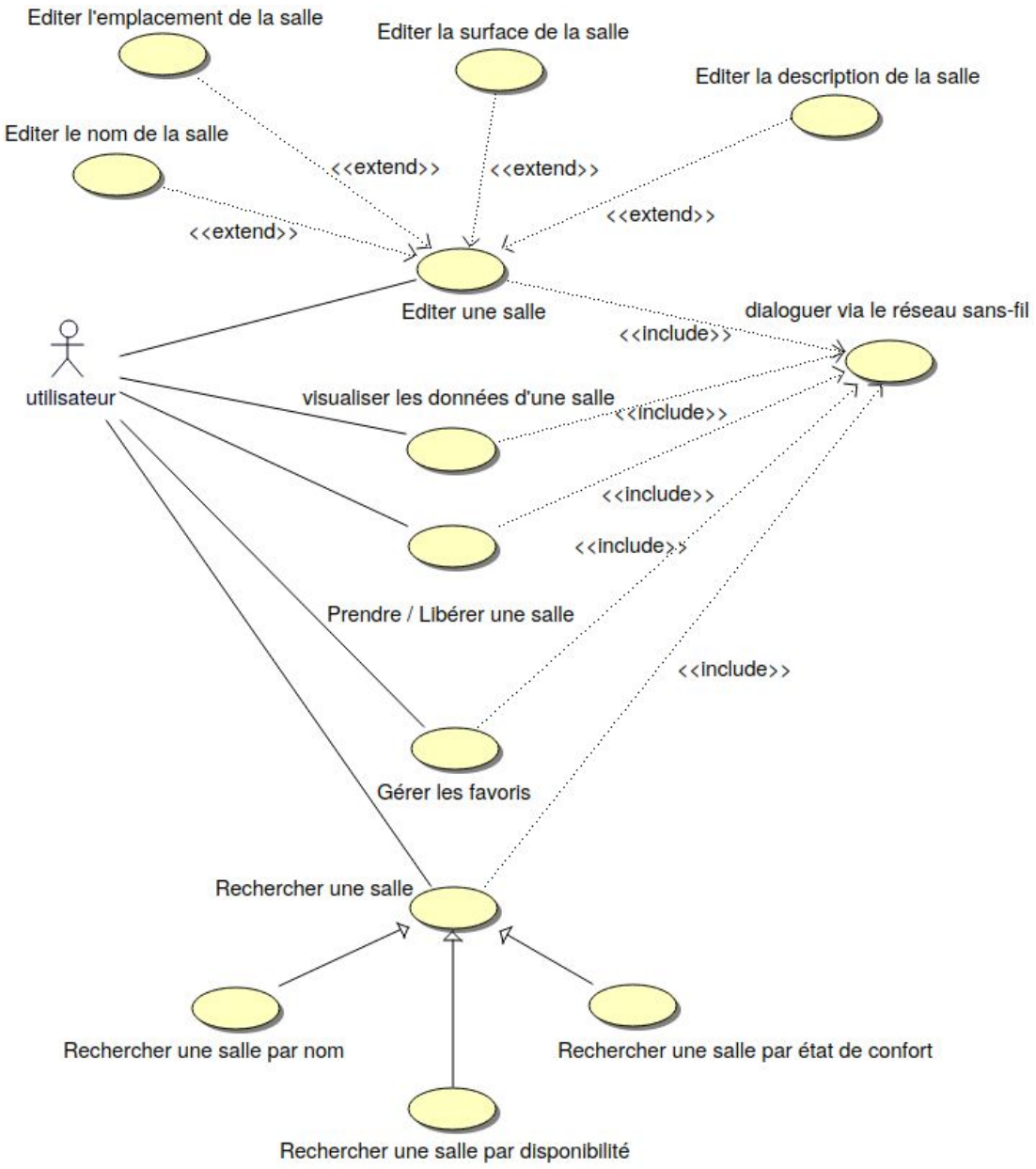


diagramme des cas d'utilisation

Le **diagramme des cas d'utilisation** permet d'avoir une **vue d'ensemble** des **fonctionnalités** du point de vue de l'utilisateur.
L'utilisateur doit pouvoir **visualiser les données d'une salle** en temps réel, tout ceci à distance depuis une interface mobile. Il peut **prendre et libérer une salle**. Il peut aussi **éditer une salle** pour les données les informations suivantes : nom, emplacement, surface, description de la salle.

Objectifs

Pour la première itération, l'application se concentre sur l'essentiel attendu de l'application qui est :

- Afficher les données (informations sur la salle, disponibilité, mesures, indice de confort) d'une salle
- Prendre/libérer une salle
- Communiquer avec le portier
- Éditer les informations associées à une salle

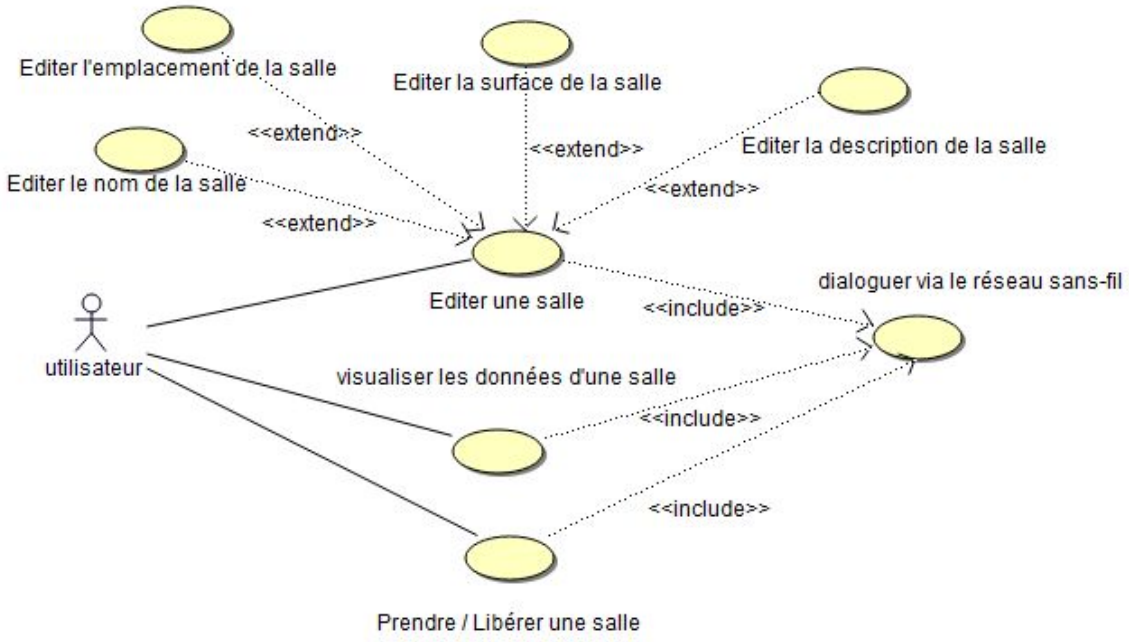
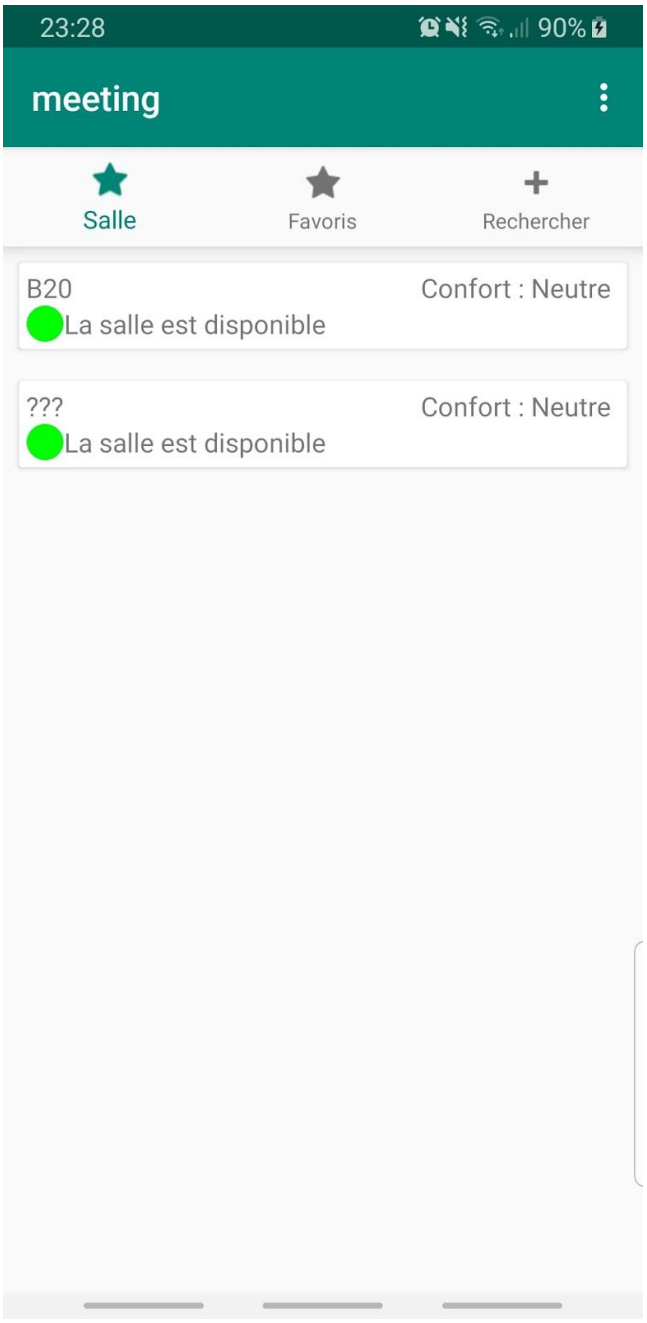


Diagramme des cas d'utilisation de la première itération

Interface Graphique

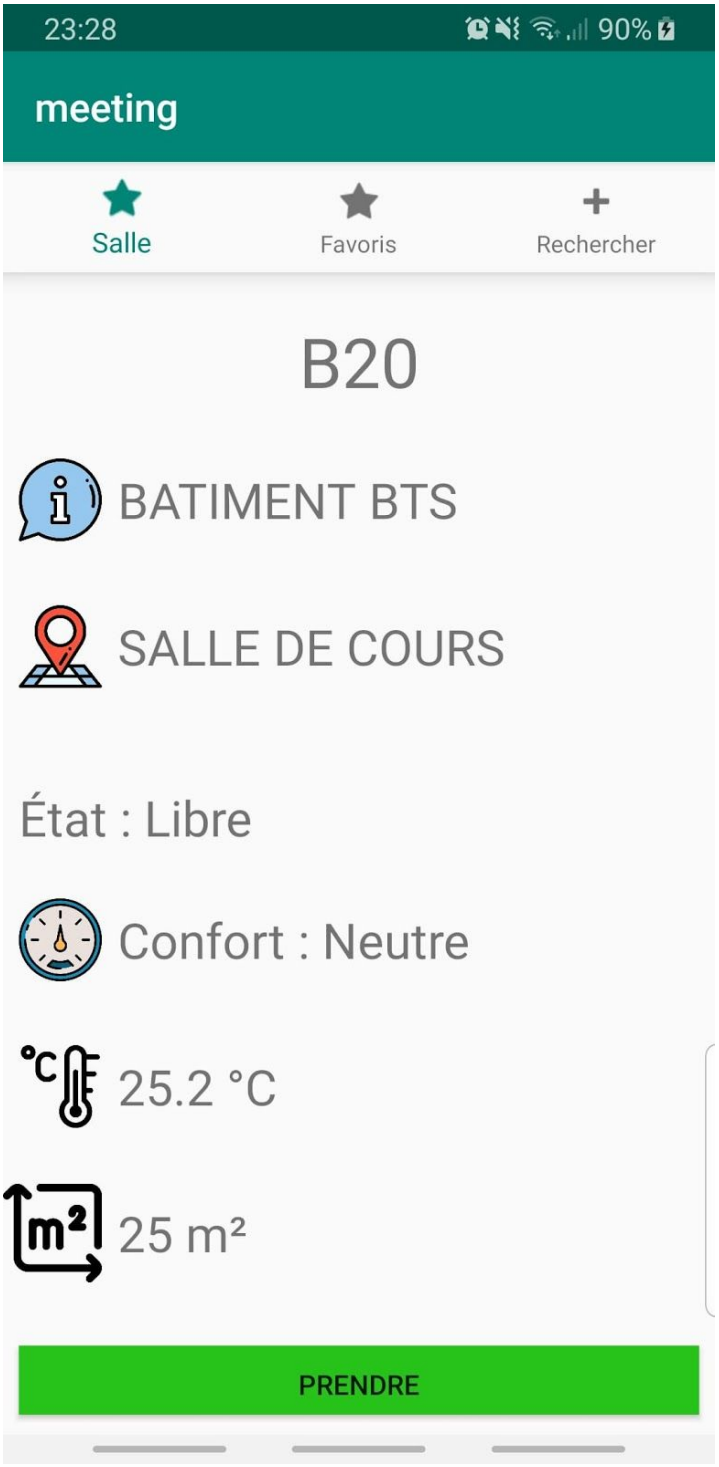


L'activité principale

L'application a détectée la présence de deux salles de réunion.
On peut noter que sur les deux salles de réunion sont disponibles :

- La première (B20) a déjà été éditée et les informations la concernant la salle sont enregistrées dans le portier ;
- La seconde (???) n'a jamais été éditée donc elle possède les valeurs par défaut (??? pour le nom, l'emplacement, la description et 0 pour la surface)

Il est possible de visualiser une salle en détail en cliquant dessus :



L'activité salle

Il est possible d'éditer une salle en utilisant le menu :

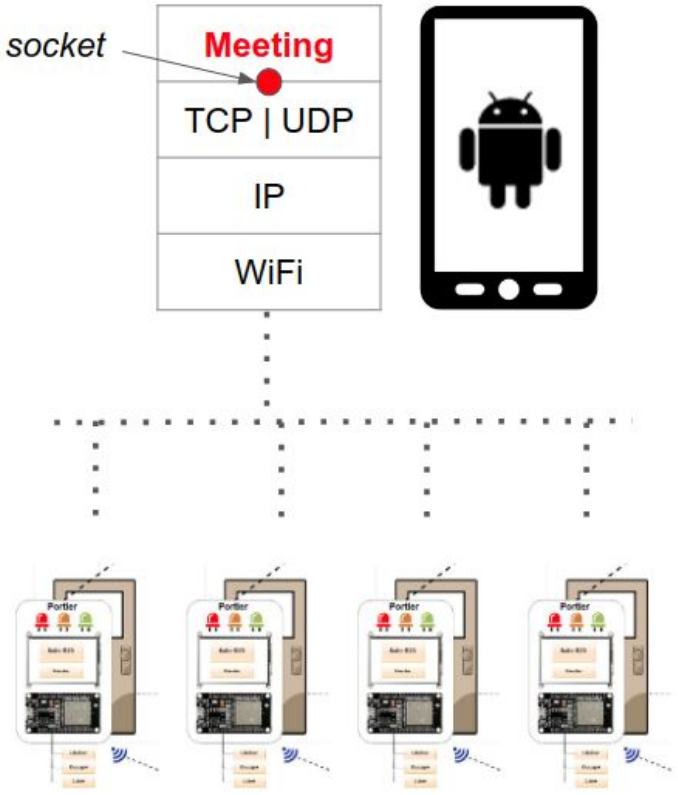
L'activité configurer salle

Choix technologique

Communication sans-fil

Dans le cahier des charges, la **communication WiFi** a été demandée entre l'application mobile et le portier. Il a fallu déterminer le choix du **protocole TCP ou UDP**. L'application mobile ne sauvegardent pas les informations des portiers. Il a été décidé de mettre tous les portiers dans un **réseau *multicast*** (adresse *multicast* des portiers : **239.0.0.42**) . Le protocole TCP ne permettant uniquement la communication *unicast* (point à point), le choix du protocole s'est porté sur **UDP**. La quantité d'informations échangée étant faible, cela convient parfaitement au protocole UDP (la plupart des systèmes utilisant UDP limite la taille des datagrammes à 8192 octets).

Il reste plus qu'à définir un protocole de communication de couche Application propre au projet. Le protocole a donc été décidé entre les étudiants 1 (étudiant EC travaillant sur le portier) et 3 (étudiant IR travaillant sur l'application mobile).



Le protocole Meeting

Le protocole Meeting et un protocole orienté ASCII.
Il a y des délimiteurs :

- de début : "**$**"
- de champs : "**;**"
- de fin : "**\r\n**"

Il y a deux styles de requêtes envoyées au portiers par le simulateur.

- les requêtes commençant par "**GET**".
  Ces requêtes demandent une information aux portiers. Elles sont envoyées sur **l'adresse multicast** (239.0.0.42)
  - Envoie : $GET;1\r\n
  - Retour : $nom;description;emplacement;surface;disponibilité;niveauDeConfort;température\r\n

  - Envoie : $GET;2\r\n
  - Retour : $nom;disponibilité;niveauDeConfort;température\r\n

  - Envoie : $GET;3\r\n
  - Retour : $nom;disponibilité\r\n

- les requêtes commençant par "**SET**".
  Ces requêtes envoient une information au portier (n'attend pas de retour). Elles sont envoyées sur **l'adresse unicast** de la salle
  - Envoie : $SET;1;nom;emplacement;description;surface\r\n
  - Envoie : $SET;3;disponibilité\r\n

Stockage des informations

Le stockage d'informations concerne les salles et leurs informations telles que le nom, l'emplacement, la description, la surface et la disponibilité de la salle. Pour la température et le niveau de confort, c'est la sonde qui les fournit au portier qui lui même les transmet à l'application sur demande.

Pour stocker cette information, deux solutions sont envisagées :

- créer un serveur pour centraliser l'information
- stocker directement dans chaque portier

La solution sélectionné a été le stockage par le portier car celui par serveur prenait trop de temps de mise en oeuvre et il aurait fallu un étudiant IR en plus.

Android

Dans le jargon d'Android, une fenêtre d'application est appelée Activity ou Activité en Français. Une Activité contient différent éléments graphiques tels que des boutons, des champs de saisies de texte, des images, etc …

Chaque Activity possède son cycle de vie géré par le système d'exploitation Android. Le système, pour des raisons de priorisation d'activités, peut tuer une activité quand il a besoin de ressources. L'Activity peut être dans les différents états suivants :

- « Active » : l'activity est lancée par l'utilisateur, elle s'exécute au premier plan.
- « En Pause » : l'activity est lancée par l'utilisateur, elle s'exécute et est visible, mais elle n'est plus au premier plan. Une notification ou une autre activité lui a volé la priorité et une partie du premier plan.
- « Stoppée » : l'activity à été lancée par l'utilisateur, mais n'est plus au premier plan et est invisible. L'activity ne peut interagir avec l'utilisateur qu'avec une notification
- « Morte » : l'activity n'est pas lancée

Afin de passer d'un état à un autre, l'Activity appele différentes méthodes qui lui sont propre:

- onCreate() : elle appelée lors du premier lancement de l'activity et peut réaliser des initialisations.
- onDestroy() : est appelée lors la mort de l'activity qu'elle soit naturelle (créée par l'action de l'utilisateur) ou provoquée par le système par manque de ressources.
- onPause() & onResume() : sont appelées lors de l'entrée ou la sortie de l'état en pause de l'activity. Elles s'occupent de sauvegarder ses états et les restituer.

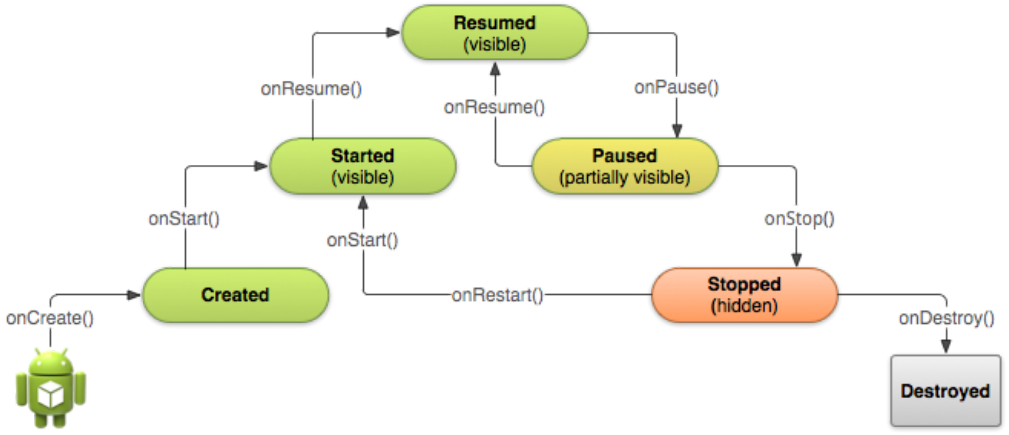
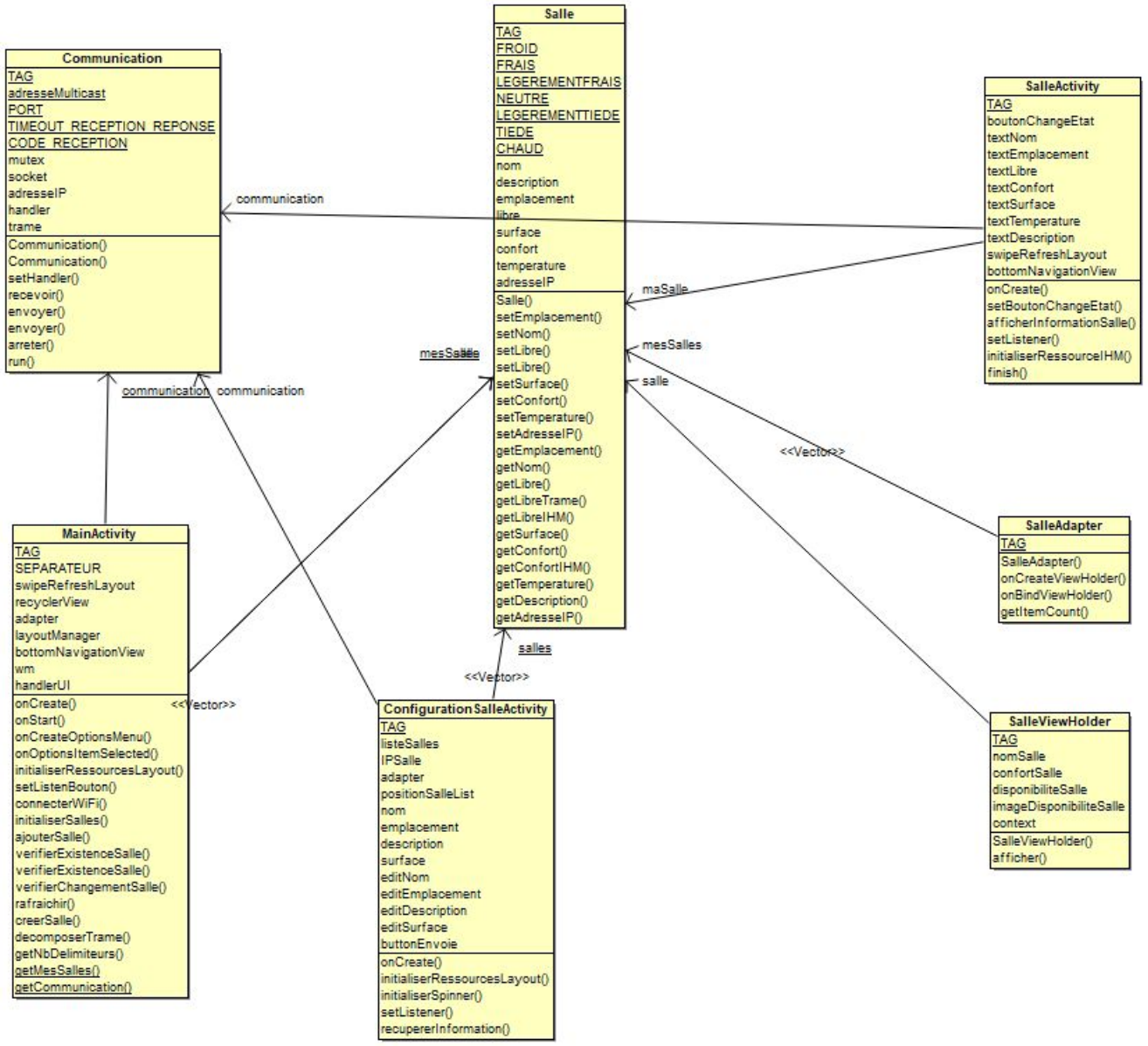
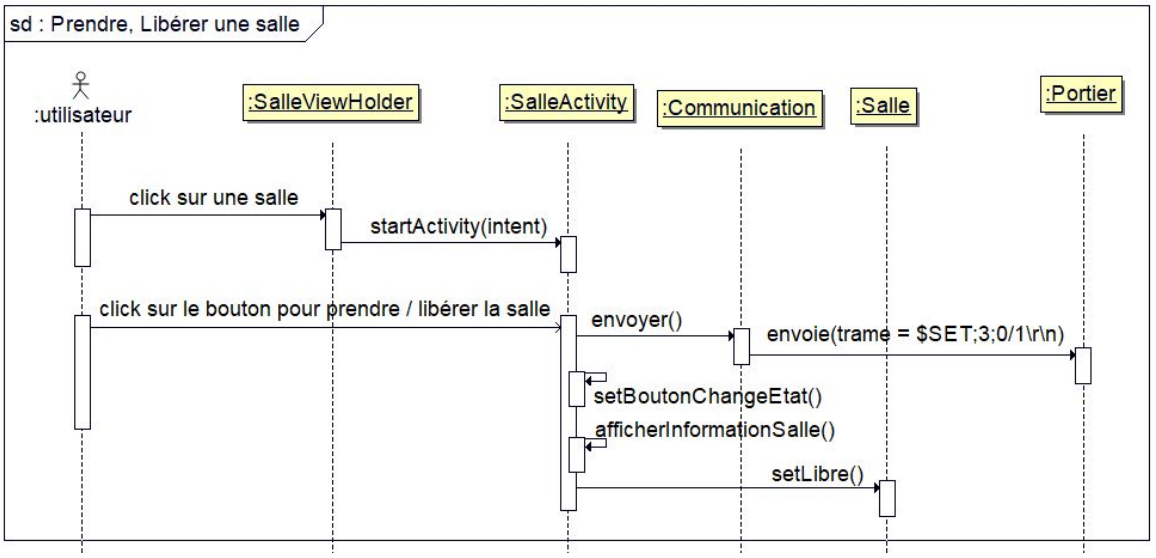


Diagramme de classes



Salle : la classe salle représente une salle de réunion.
Communication : la classe communication permet la communication avec les portiers en Wi-Fi.
MainActivity : IHM (Interface Homme Machine) principale.
ConfigurationSalleActivity : IHM permettant de configurer une salle.
SalleViewHolder : Liste des salle de réunion dans MainActivity.
SalleAdapter : Adaptateur pour SalleViewHolder.
SalleActivity : IHM d'une salle de réunion.

Diagramme de séquence : Prendre / Libérer une salle



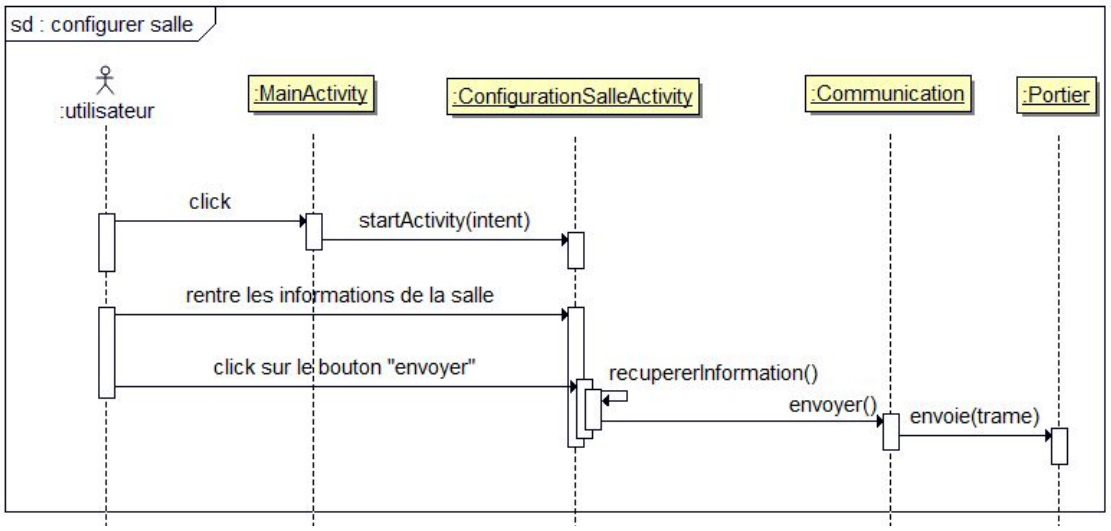
Le diagramme de séquence est un diagramme d'interaction dont le but est de décrire comment les objets collaborent au cours du temps et quelles responsabilités ils assument. Il décrit un scénario d'un cas d'utilisation.

Pour prendre/libérer une salle, l'application affiche les informations de la salle et propose à l'utilisateur de cliquer sur le bouton (Prendre / Libérer)

```
public void onClick(View v)
{
   maSalle.setLibre();
   if(communication != null)
   {
      communication.envoyer("$SET;3;" + maSalle.getLibreTrame() +
"\r\n", maSalle.getAdresseIP());
   }
   setBoutonChangeEtat();
   afficherInformationSalle();
}
```

Quand l'utilisateur va cliquer sur le bouton "Prendre/Libérer", la méthode "onClick" sera appelée. Elle changera l'état de la salle. On vérifie que la communication est bien créée avec les portiers. Si oui, on envoie (avec la méthode "envoyer" de la classe communication) une trame "SET;3;" pour changer l'état de la salle au portier. Pour finir, "setBoutonChangeEtat" et "afficheInformation" permettent d'afficher le nouvel état de la salle.

Diagramme de séquence : Configurer une salle



Dans un premier temps, l'utilisateur accède à ConfigurationSalleActivity par le MainActivity. Puis après avoir saisies les informations dans les champs prévus à cette effet, il peut les envoyer au portier sélectionné.

```
@Override
public void onClick(View v)
{
   recupererInformation();
   if(communication != null)
   {
      communication.envoyer("$SET;1;" + nom + ";" + description + ";" +
emplacement + ";" + surface + "\r\n", IPSalle.get(positionSalleList));
   }
}

public void recupererInformation()
{
   nom = editNom.getText().toString();
   description = editDescription.getText().toString();
   emplacement = editEmplacement.getText().toString();
   surface = editSurface.getText().toString();
}
```

La méthode "recupererInformation" permet de récupérer les informations saisies par l'utilisateur.

On vérifie que la communication est bien créée avec les portiers. Si oui, on envoie (avec la méthode “envoyer” de la classe communication) une trame “SET;1;...” avec les informations récupérées plutôt.

Tests de validation

Description	Action à réaliser	Attendu	Résultat
Visualiser les informations d’une salle sélectionnée	Appuyer sur la salle voulue dans la MainActivity	Affichage de SalleActivity avec la bonne salle	**OK**
Prendre une salle	Appuyer sur le bouton “Prendre” dans SalleActivity	L’état du portier en question change (en état “prise”). Les informations de la salle changent sur son état dans toute l’application	**OK**
Libérer une salle	Appuyer sur le bouton “Libérer” dans SalleActivity	L’état du portier change (en état “libre”) .Les informations de la salle changent sur son état dans toute l’application	**OK**
Configurer une salle	Après avoir rempli le formulaire et sélectionné la salle, appuyer sur “Envoyer”	Le portier reçoit les nouvelles informations. Les informations de la salle changent dans toute l’application	**OK**