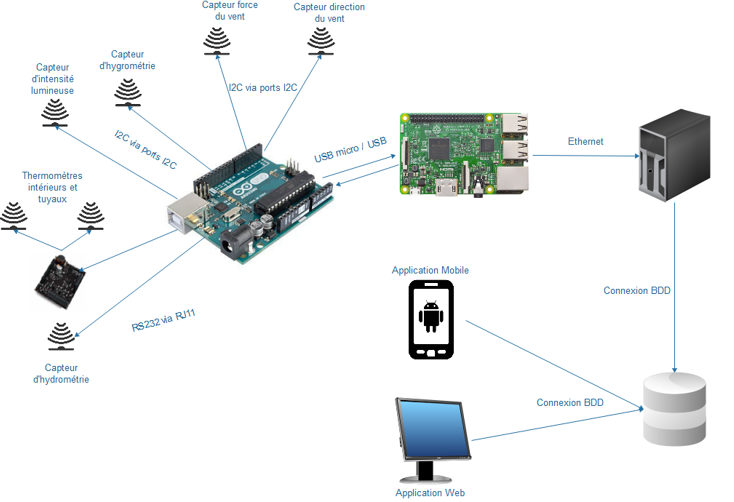
|  |
| --- |
| Groupe Olivier |
| Dossier technique du projet – Partie personnelle |
| Etudiant 4 : Visualisation de l’état du système et acquisition de mesure par capteur |

# RAYNAL Audran

# Situation dans le projet

### A – Synoptique de la réalisation



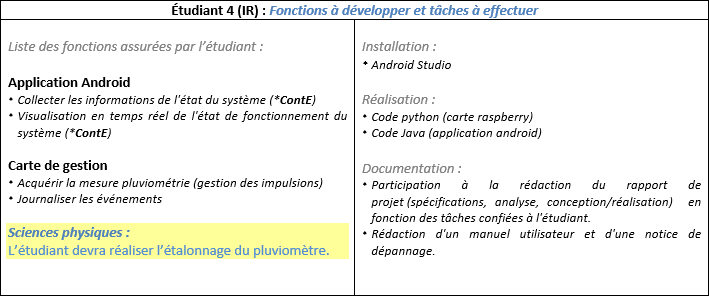
*Voici les parties de la synoptique me concernant.*

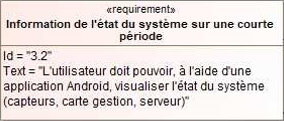
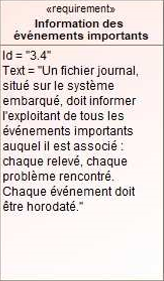
Dans le projet, ma principale tâche est le développement d’une application sur smartphone Android. Je programmerai sur Android Studio, l’IDE officiel de développement sous Android. L’application aura comme objectif de permettre à l’utilisateur de visualiser en temps réel l’état du système qui est composé des capteurs, de la carte de gestion et du serveur. Elle devra donc récupérer les valeurs de l’état du système et les afficher.

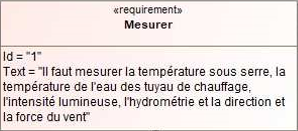
On m’a confié le capteur d’Hydrométrie. Il s’agit du Pluviomètre Rain Collector II de la société Davis. Je dois acquérir la mesure pluviométrie grâce a ses impulsions. Je réaliserai son étalonnage, c’est-à-dire que pour une certaine quantité d’eau, je dois connaitre la hauteur au sol que cela représente en millimètre.

Pour réaliser cette tâche, j’aurai besoin d’une carte d’acquisition Arduino. Je développerai un programme grâce à l’IDE Arduino

# MES TACHES :



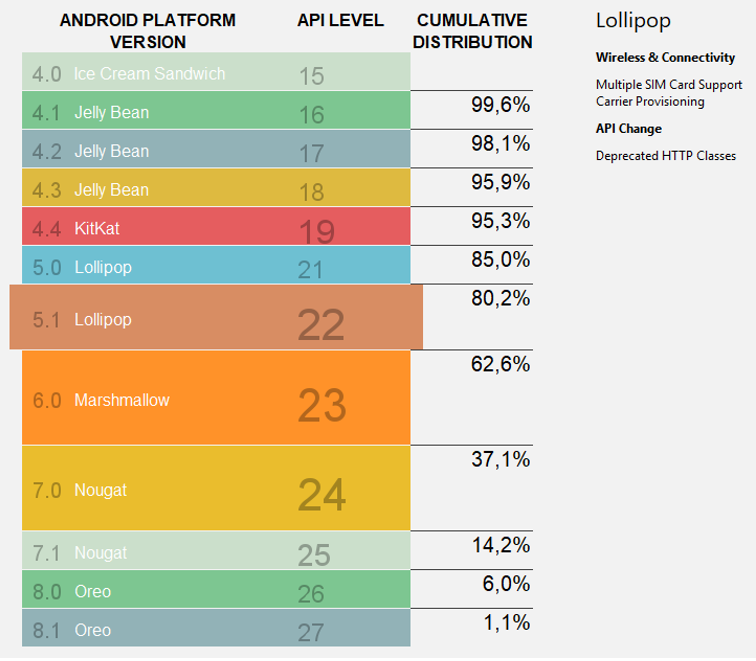






Un API de développement *(Application Programming Interface)* est une interface de programmation

Le choix de l’API de programmation minimum pour mon application s’est porté sur 2 critères : le plus important est la distribution globale de l’API.

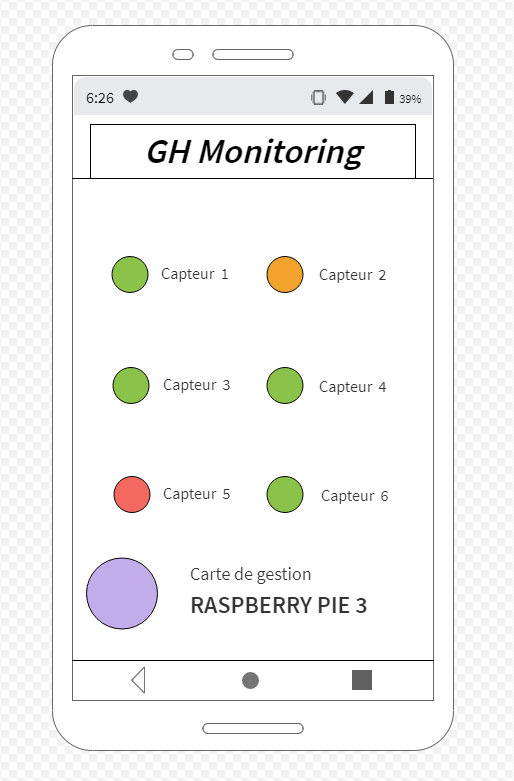
L’application n’étant pas destiné au grand public mais à seulement quelques utilisateurs professionnels, on doit pouvoir l’utiliser sur le plus de modèle de smartphone Android possible. Le second critère est l’âge de l’API, car il doit être assez récent pour que je puisse profiter d’un maximum de fonctionnalité et de méthode de développement. Il fallait faire un compromis et donc l’API de niveau 22, Android 5.1 Lollipop à été choisie, car largement répandu, et assez récente car cette version d’Android est sortie en 2015.

Ce tableau récapitulatif peut être trouvé au démarrage d’un projet sur Android Studio mais aussi sur le site officiel de développement Android.

Ce prototype d’IHM a servi de base pour le développement de l’application.

L’objectif de l’Application est simple : il faut afficher en temps réel l’état du système et de ses composants : La carte de Gestion et les 6 capteurs la composant. C’est-à-dire que l’utilisateur de l’application doit être informé rapidement et simplement si un composant ne fonctionne plus.

J’ai donc décidé de réaliser des indicateurs avec un code couleur. Si la pastille de couleur est verte, il n’y a pas de problème détecté, le composant fonctionne. Si elle est rouge, il y a un problème et une intervention de maintenance est requise. Il n’y aura qu’une seule fenêtre, pour garder la simplicité.

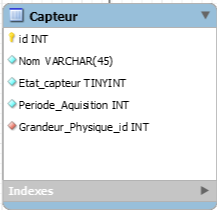


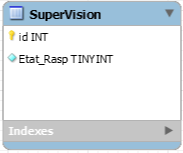
GH monitoring signifie Surveillance de Serre *(Greenhouse monitoring)*

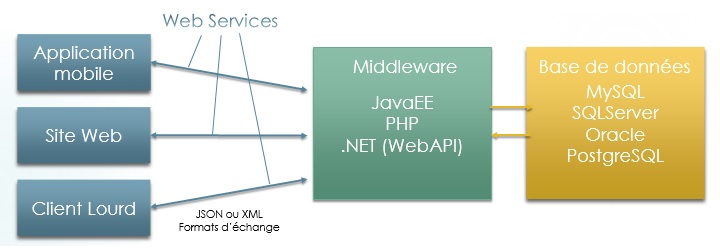
# FONCTIONNEMENT DE L’APPLICATION

Les états des capteurs sont stockés dans la base de données, tout comme l’état de la Raspberry.

Le champ Etat\_capteur fait partie de la table Capteur et le champ Etat\_Rasp fait partie de la table SuperVision.





Connecter une application Java directement à une base de données est impossible. C’est pourquoi je suis obligé de passer par un middleware, ici ce sera un service web, une page PHP que je devrai écrire.

Source : http://tutorielandroid.francoiscolin.fr/index.php

L’application web en PHP contiendra les requête SQL qui permettra de se connecter à la base de données pour pouvoir récupérer les valeurs : les états du système. Ensuite, la page PHP enverra les données en format JSON *(JavaScript Object Notation)* à l’application Android qui pourra les interpréter.