**Diapo 4 :**

Les objectifs de ce projet sont multiples. Le principal est la création d’un boîtier intelligent qui permettra de contrôler des équipements domotiques dans une maison. Ce boîtier devra communiquer avec ces équipements en non-filaire. Comme équipements, nous voulons pouvoir contrôler n’importe quels types d’équipements domotiques, que ce soit des lampes, des volets, des thermostats etc.

L’interaction entre l’utilisateur et le système se fera à travers une chat box textuelle à l’intérieur d’une application. Cette application devra être multiplateformes et accessible par tous les membres d’une famille, utilisant des comptes différents.

Nous voulons une solution domotique intuitive et facile d’utilisation. Elle doit être pratique et on ne veut pas que les données soient stockées sur des serveurs externes à notre système.

Nous avons défini dans le cahier des charges nos objectifs et les fonctionnalités que nous voulions implémenter en fonction de leur priorité. Mais nous avons également un objectif secondaire qui est de créer un site web permettant de suivre l’avancement du projet.

**Diapo 9 :**

Nous avons choisi de coder notre serveur en JavaScript. Ce choix est basé sur le fait que nous avons vu ce langage en cours et c’est donc le langage que l’on maîtrise le mieux. De plus, dans le cours de WebObject, nous avons fait un serveur de tchat. Nous allons donc utiliser les bases de ce serveur pour concevoir le nôtre. Cela facilitera sa conception. Nous avons également vu que node.JS était compatible avec CoreNLP. Il existe des packages (corenlp) et des wrapper (stanford-corenlp) disponibles. Ainsi, notre partie traitement de langage va pouvoir communiquer sans problème avec le serveur. De plus, vous verrez ensuite que le serveur traite également avec le client. Il existe des librairies compatibles avec Android, cela va donc faciliter la mise en place de la communication.

Notre serveur aura deux rôles dans l’envoi de données :

- Envoyer des données au client

- Envoyer des données aux objets connectés

**Diapo 10 :**

## 1. Envoyer des données au client

Il y aura deux moments où le serveur enverra des données au client. Le premier est lorsque que le client s'authentifie. Ainsi, lorsqu’il aura reçu les données du client, le serveur stockera ces données dans une base de données puis renverra au client la validité ou non de son authentification.

Dans un second temps, lorsque le client enverra des commandes via l’application mobile afin d’effectuer des actions sur les objets connectés, le serveur renverra l’état de la commande si un retour d’état est disponible avec l’objet en question. C’est-à-dire que le client aura la possibilité de savoir si sa commande a bien été effectuée ou non.

Ces différents retours d’état seront envoyés sous le format JSON.

Le format JSON d’un message envoyé au client lors d’une authentification sera le suivant :

*{nom : « nomClient », action : « authentification », réponse :  « acceptée ou non »}*

***Diapo 11 :***

## 2. Envoyer des données aux objets connectés

Pour les objets connectés, il y aura deux cas de figures. C’est-à-dire qu’il peut y avoir des objets connectés, tels que la lampe ikea tradfri, qui possèdent une box ainsi que d’autres qui n’en possèdent pas. Ainsi le serveur devra être capable de s’adapter. Lorsqu’une box sera présente, le serveur se servira des librairies préalablement installées permettant de communiquer avec la box (ex : libcoap, voir en annexe). Ensuite la box se chargera de communiquer avec l’équipement.

S’il n’y a pas de box, il faudra alors installer un module hardware faisant office de relai permettant au serveur de communiquer avec les différents équipements.

**Diapo 12 :**

Dans le serveur, il y aura deux parties, une partie traitement des commandes et une partie base de données. Comme vous l’avez vu, le client va devoir s’authentifier avec un nom d’utilisateur et un mot de passe. Il va falloir stocker ces données. Pour cela, les bases de données de données relationnelles sont les plus adaptés à notre projet. En effet, pour notre projet, nous aurons juste besoin de stocker le pseudo et le mot de passe de l’utilisateur, donc notre schéma n’aura pas besoin d’être modifié.

Nous avons choisi d’utiliser sqlite3 car nous l’avons déjà utilisé l’année dernière et donc nous avons des connaissances sur cette base.