

**MEMOIRE DE FIN D’ETUDES**

**Pour l’obtention du**

**Diplôme National d’Ingénieur en Génie Informatique**

**Sujet :**

***Conception et développement d’une application***

***Domotique***

Elaboré par :

**Mohamed shadi SAAD**

***Soutenu le \*\*\*\*/2015 devant la commission d’examen :***

|  |  |
| --- | --- |
| **M.** | **Président** |
| **M.** | **Rapporteur** |
| **M. Anis Benammar** | **Encadreur** |

**Année universitaire 2014-2015**

**Introduction générale**

L’informatique s’est introduite peu à peu dans notre quotidien grâce aux différentes évolutions technologiques, nous produisons sans cesse de nouveaux besoins pour ensuite les satisfaire avec de nouvelles inventions. Tous comme notre vie en général, nos maisons se trouvent de plus en plus technologies. Les habitats du futur répondent à une probable insatisfaction innée de l’homme qui croit augmenter son pouvoir sur les choses par la technique. On voit donc que sa maison « s’adapte » à lui et à ses besoins.

Au même titre que nous gérons notre budget, organisons notre temps et nos activités, la domotique qui regroupe l’ensemble des techniques d’électronique, de physique de bâtiment, d’automatisme, de l’informatique et de télécommunication utilisés dans les maisons .Elle vise à assurer des fonctions de sécurité, de confort, de gestion d'énergie et de communications, nous offre la possibilité d’opter pour une maison qui s’adaptera à notre rythme de vie et à nos habitudes, eux-mêmes évolutifs au fil des années. Avec les nouvelles technologies domotique, l’homme peut quitter leur domicile pour le travail, appuyer sur un seul bouton pour éteindre les éclairages oubliés, mettre le chauffage en mode « économie », fermer les volets roulante. Ainsi, la domotique sert à la fois nos besoins et notre envie de confort. Elle prend en compte des situations qui ont un sens dans notre vie quotidienne avec un simple clic sur un smartphone.

C’est dans ce cadre que se situe notre projet de fin d’études intitulé «Domotique». Ce projet a pour objectif de développer une application Android qui permet de piloter des équipements domestiques à distance et consulter l’état de l’habitation.

Pour ce faire, nous procédons par une étude théorique afin de mieux cerner le contexte de notre travail. Cette étude fait partie des objectifs de notre rapport qui est subdivisé en trois chapitres : dans le premier chapitre nous analysons les besoins des utilisateurs et les défaillances des systèmes existants pour définir nos orientations. Dans le deuxième chapitre, nous entamons une phase de spécification et de modélisation conceptuelle de notre application en détaillant les différents diagrammes. Le troisième chapitre, sera consacré à la description des choix techniques pour la réalisation de l’application **Android** et la présentation des résultats obtenus.

Le présent rapport sera clôturé par une conclusion générale et quelques perspectives.

1. **Recueil**

L’étude préalable constitue une étape préliminaire pour la réalisation d’une application.

En effet, elle permet d’analyser, d’évaluer et de critiquer le fonctionnement habituel, tout en élaborant la liste des solutions possibles.

Ce chapitre sera réservé pour présenter l’étude préalable de notre projet. Nous commençons par la définition du champ de l’étude et les objectifs à atteindre. Ensuite nous analysons quelques solutions existantes sur le marché en discutant leurs avantages et leurs inconvénients. L’analyse et le critique de l’existant nous ont permis de cerner nos objectifs afin de développer un système de qualité dans le futur. Enfin, nous proposons les différentes solutions aux problèmes soulevés.

1. **Présentation du projet :**

L’objectif principal de notre travail est de concevoir et de développer une application domotique qui nous permet de superviser les équipements de notre futur application et notre habitas souhaité

## 2.1 Domotique

### Définition

La domotique est le domaine technologique qui s’intéresse à l’automatisation du do- micile, d’où l’étymologie du nom qui correspond à la contraction des termes "maison" (en latin"domus") et "automatique". Elle consiste à mettre en place des réseaux reliant différents types d’équipements (électroménager, hifi, équipement domotique, etc) dans la maison. Ainsi, elle regroupe tout un ensemble de services permettant l’intégration des technologies modernes dans la maison.

### Usages

Les domaines d’utilisation de la domotique sont très nombreux et en évolution exponentielle. Ils sont divisés généralement en trois catégories ou services principales, l’économie d’énergie, confort et loisir et la sécurité.

#### Sécurité

La sécurité présente un point très recherché dans ce domaine. Ainsi elle présente un service très critique du coté implémentation et conception. De nombreux systèmes de sécurités sont envisageables tels que :

* + - * + Une caméra vidéo à amplification de rayonnement et un code personnel pour

contrôler et faciliter les entrées

**2 .3 Techniques de la domotique :**

La domotique est basée sur la mise en réseau par une *« centrale de commande »* des différents appareils électriques de la maison.

**La centrale de commande** : Programmable et contenant des modules embarqués (passerelles domestiques) ou une interface micro-informatique (écran tactile, serveur, etc.) elle joue le rôle d'une « intelligence » centralisée et d'interface homme-machine centralisée pour l'usager ou des services distants de contrôle. Elle tend à devenir plus réactive aux changements du contexte. Pour cela elle réunit ou remplace divers appareils (programmateur/régulateur de chauffage, centrale d'alarme, système de centralisation des persiennes électriques, contacteur jour/nuit du cumulus, délesteur, programmateurs horaires, systèmes d'arrosage automatiques, etc.), qui peuvent fonctionner et interagir de manière *asynchrone*.

**L’écran de contrôle :** il est fixe dans le domicile, où il peut être émulé à distance via le réseau ADSL de la maison puis l'internet), permettant le pilotage de la maison à distance pour tout ou partie des fonctions domotiques. L'interface distante peut être par exemple un ordinateur de poche, un [téléphone portable](https://fr.wikipedia.org/wiki/T%C3%A9l%C3%A9phone_portable) ou [smartphone](https://fr.wikipedia.org/wiki/Smartphone), une [tablette tactile](https://fr.wikipedia.org/wiki/Tablette_tactile), une [télécommande](https://fr.wikipedia.org/wiki/T%C3%A9l%C3%A9commande) (universelle ou non), une interface sur télévision connectée, un écran + souris, etc.[ WIKI].

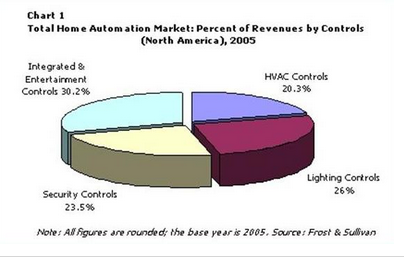
**Le pilotage à distance** : Il permet de faire face quasiment en temps réel à des situations particulières.

Figure1 : Architecture Domotique.

**2. 4 Le marché domotique :**

**. Les chiffres*:***  
D’après le cabinet de conseille cabinet de conseil en IT Frost & Sullivan, ces applications actuelles se chiffrent à 232,6 millions de dollars en 2006 avec une très garde part de la partie confort et loisirs. Toujours d'après Frost & Sullivan, ce chiffre devrait doubler à l'horizon 2013. D'après ON World, un grand fournisseur de produits sans fil, la part du sans fil devrait monter représenter 98% d’ici 2015.

.**Les facteurs d’évolution *:***  
Cette évolution est possible du fait de caractéristiques intrinsèques aux produits et aux services domotiques proposés. Les équipements deviennent de plus en plus légers et tendent de plus en plus vers le sans fil, ce qui augmente la mobilité des utilisateurs. Ils consomment de moins en moins d'énergie, ce qui diminue les coûts globaux. Par ailleurs, les constructeurs se penchent de plus en plus sur l'ergonomie e l'utilisabilité de leurs produits. Mais le facteur le plus important est sans doute le fait que les possibilités sont uniquement limitées par l'imagination.



###### Figure2 :

1. **Objectifs à atteindre.**

Les principaux objectifs à atteindre à travers la réalisation de l’application domotique sont les suivants :

* Pouvoir contrôler à distance ou directement chez soit, le comportement des appareils qui sont intégré au système domotique.
* Automatiser les taches répétitives du quotidien (exemple: réveil, ouvrir les volets, activer la climatisation, désactiver l'alarme...)
* Mieux gérer la consommation en électricité et donc faire des économies.

1. **Etude de l’existant**

L’étude de l’existant est une étape fondamentale dans le développement des applications. Cette étape consiste à définir les caractéristiques du système existant et à dégager ses défaillances ce qui permet de définir les solutions adéquates.

Dans ce qui suit, nous présentons une analyse de l’existant, puis nous détaillons la critique de l’existant à fin de présenter une solution.

**4.1. Analyse de l’existant**

**PackTYXIA550 :** Pack Micromodules Eclairage est une application tunisienne caractérisé par :

 Simple Allumage

 Commande centralisée sans fil avec retour d’information

 Ajouter un nouveau point de commande grâce aux interrupteurs sans fil

 automatiser l’éclairage à votre passage

**Pack Micromodules Volets Roulants**

Centralise la commande de volets roulants qui peuvent être actionnés individuellement, par groupe ou en scénario.

 Pilotage des volets roulants à partir d’un seul point de commande.

 Une solution pour quitter un lieu sans oublier de fermer un volet.



**PACK DRIVER 630 RADIO/FP :**

Programmateur et récepteurs radio pour chauffage électrique

 Une solution pour la rénovation du chauffage électrique.

 Sans fil, pour le choix du bon emplacement.

**4.2. Critique de l’existant**

Cette étape nous permet de dresser un bilan du système actuel et d’en faire ressortir les faiblesses.

Voici les lacunes qui ont pu être décelés suite à cette analyse :

 Ces solutions appelés solutions semi domotique ne présentant pas toujours une solution complète : chaque service (lumière, volet roulant, chauffage, système alarme) est séparé et se géré avec des équipements différents incapables de dialoguer ensemble

 Ils existent des applications présentes dans le marché tunisien qui répondent à plusieurs problématiques mais elles sont trop chères

* 1. **Solution proposée :**

### Exemple de solutions domotiques

Comme tout domaine technologique qui vit une évolution remarquable, un nombre importants de sociétés se concentrent sur ce domaine en implémentant des solutions citant par exemple :

* My Home Domotique : c’est une solution proposé par la société ’legrand’ qui est décomposée sous des package qui décomposent la maison en des catégories tels que la gestion d’énergie, la gestion des portes, les caméras de surveillances. Les prix de solutions sont variés[9] selon la catégorie et la fonctionnalité. Par exemple un réseau multimédia basique qui comporte un coffrets et équipement pour distribution prise téléphone et Tv coute entre 508 et 540e.
* Home : c’une solution lancée par la société de télécommunication française SFR qui comporte deux packs Home sécurité avec un abonnement mensuel de 9.99e ou un pack Home sécurité Premium avec un abonnement mensuel de 19.99e. Aussi SFR offre la vente des équipements de la solution séparer avec des couts différentes.
* Solution vivint : solution proposée par la société américaine vivint qui comporte trois packs ’HOME Security’ avec un abonnement mensuel de 49.99$ , un pack ’Energy management’ avec un abonnement mensuelle de 57.99$ et ’Home auto- mation’ avec un abonnement mensuel de 68.99$.

Tebis : propose par la société Hager, comme les solutions présentées précédem- ment, Hager organise ses solutions selon des catégories telles que la sécurité et l’économie d’énergie.

Pour atteindre les objectifs fixés à l’avance nous proposons les solutions suivantes :

1. ***Problématique 1 (partie électrique) :***

Dans cette partie, nous allons étudier les composants électriques de notre produit en détaillant les critères de choix.

Les microcontrôleurs

Définition:

Un microcontrôleur est un système informatique formé généralement d’un processeur et divers périphériques capables d’effectuer des tâches spécifiques. On peut mentionner entre autres les périphériques d’Entrée/Sortie, les Timers, les convertisseurs Analogiques/ Numériques.

Les microcontrôleurs se caractérisent par un haut degré d'intégration. Ils trouvent pleinement leurs intérêts dans les applications embarquées qui ont plus besoin d’un grand nombre de lignes d’E/S pour la commande, l’acquisition ou la communication que d’une capacité de traitement élevée. Ils permettent ainsi la conception de cartes de commandes, de petites tailles et avec des rapports prix/qualité raisonnables et ce pour un grand nombre de systèmes embarqués [11].

Ci-dessous un tableau résumant les différentes caractéristiques de quelques types de microcontrôleurs disponibles sur le marché:

**Tableau 1 :** Table de critère entre les trois solutions :

**Tableau III-1:** Les

|  |  |  |  |
| --- | --- | --- | --- |
| **Dispositif Microcontrôleur** | **ATmega2560** | **Pic16f877** | **STM 32** |
| **Flash** | 256KB | 8KB | 128KB |
| **EEPROM** | 4KB | 256B | 4KB |
| **RAM** | 8KB | 368B | 8KB |
| **General Purpose I/O pins** | 86 | 33 | 86 |
| **16 bits resolution PWM channels** | 12 | 2 | 8 |
| **ADC Channels** | 16 | 8 | 12 |

--> Suite à ce tableau comparatif, nous avons choisi le microcontrôleur **ATmega2560** ayant les caractéristiques les plus performantes. Ce dernier est compatible avec la carte ARDUINO

Définition de la carte ARDUINO (microcontrôleur choisi)

C'est une carte basée sur un microcontrôleur Atmel ATmega 2560. Elle dispose 256KB de mémoire flash pour stocker les programmes qu'on transfère. Comme elle possède 54 entrées/sorties numériques (dont 15 peuvent être utilisés comme sorties PWM "voire annexe 2") et 16 entrées analogiques qui lui permettent de communiquer avec d'autres éléments (LED, interrupteurs, moteurs...).

Il existe bien sûr des microcontrôleurs autant voire plus performants que ceux de la carte Arduino, mais celle-ci est plus intéressante pour plusieurs raisons.

Tout d'abord, elle n'est pas chère comparée à ses concurrentes. En plus, le logiciel nous permettant de d'écrire, compiler puis transférer le programme de la carte est compatible à la fois avec Linux, Windows et MacOS. Enfin, elle utilise le langage C qui est très utilisé que ce soit chez les utilisateurs les plus avancés ou les débutants.

#### Description

L’ArduinoMega 2560 est une carte microcontrôleur basée sur l’ATMEGA2560. Elle contient tout le nécessaire pour soutenir le microcontrôleur, il suffit de la connecter

à un ordinateur avec un câble USB (Universal Serial Bus) ou d’alimentation avec un adaptateur AC-DC ou batterie pour commencer. Le Mega 2560 est une mise à jour de l’ArduinoMega, qui la remplace.

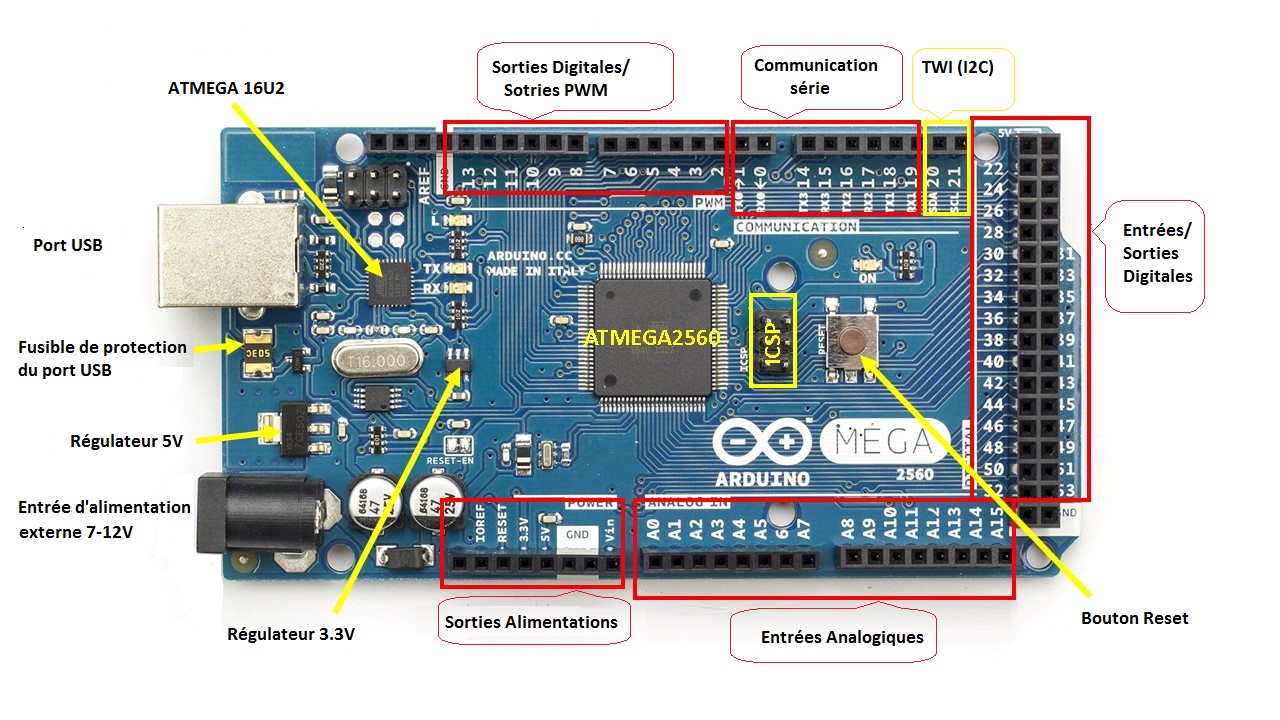
La Mega2560 diffère de toutes les cartes précédentes en ce qu’elle n’utilise pas la puce du pilote FTDI (Future Technology Devices International) USB-série. Au lieu de cela, elle dispose de l’ATmega16U2 programmé comme un convertisseur USB-série.

#### Caractéristiques

La carte Arduino Mega 2560 est équipée d’un microcontrôleur ATMEGA2560 avec un Vitesse d’horloge de 16 MHz, fonctionne avec une tension de 5V (Tension d’entrée re- commandé 7-12V Tension d’entrée limites 6-20V).

Contient 54 broches d’entrée/sortie numérique 54 (dont 15 fournissent une sortie PWM(Power)) et dont broches d’entrée analogiques 16. Un Mémoire flash 256 Ko dont 8 Ko utilisés

par le bootloader, 8 Ko de SRAM et EEPROM de 4 Ko.



###### **Figure III-18: Carte arduino**

1. ***Problématique 2 :***

### Systèmes d’exploitation mobile

Les Smartphones et les tablettes numériques fonctionnent grâce à leurs systèmes d’exploitation (OS) mobile qui sont spécifiques à ces terminaux. Il existe des systèmes ouverts, tesl que Android de Google et des systèmes propriétaires qui ne fonctionnent uniquement sur les machines vendues par le fabriquant, tels que iOS d’Apple et Win- dows Mobile de Microsoft.

#### iOS d’Apple Inc.

iOS a été développé par Apple pour l’Iphone, l’iPod touch, et l’iPad. Il est dérivé de Mac OS X dont il partage les fondations. iOS comporte quatre couches d’abstraction, similaires à celles de Mac OS X : une couche "Core OS", une couche "Core Services", une couche "Media" et une couche "Cocoa"1,2. Il occupe moins d’un demi-gigaoctet (Go) de la capacité mémoire totale de l’appareil.

La plus part des outils de développement du SDK de Apple étaient déjà présents dans Mac OS. L’outils le plus connu est : Xcode qui est un environnement de développement intégré par défaut sur Mac OS X qui permet d’écrire, gérer et compiler des projets écrit en Objective-C. L’iPhone SDK y ajoute les bibliothèques de développement pour iOS, Interface Builder et iPhone Simulator (anciennement Aspen Simulator) est le seul de ces outils à avoir été développé spécifiquement pour l’iPhone SDK.

Plusieurs limites ont étaient imposés par Apple sur son système d’exploitation les plus connu sont : iOS est un système fermé : il faut débrider ’cracker’ le système afin de bénéficier des applications et des services non fournit par Apple.

Certains formats ou produit sont interdits sur ce système tels que les fameux Flash de Adobe puisque Apple voit que l’utilisation de ce produit influe négativement sur la durée de vie de la batterie et en vue de ne pas ralentir le système. Ce système est dédié seulement au produit ou les terminaux Appel.

#### 2 Bada

Bada est un système d’exploitation inventé par Samsung pour certain marque de ses produits tels que Samsung Wave. Ce système basé sur le système d’exploitation propriétaire SHP OS a la 3éme part du marche en France et équipe plus d’un million de Smartphone. Le 25 Février 2013 Samsung a annoncé l’arrêt de développement de Bada et de se concentrer sur le développement de son nouveau SE Tezen.

#### BlackBerry OS

Il s’agit d’un SE dédié aux téléphones mobiles de la gamme BlackBerry , conçu par la societé canadienne Research In Motion (RIM). Ce système multitâche est connu surtout dans le domaine des affaire ’Busnesse ’ puisque la plus part des utilisateurs de ce Smartphone sont les hommes d’affaires puisque ce dernier support les courriels à travers le protocole Mobile Information Device Profil (MIDP) qui permet la synchronisation avec les messageries d’entreprise tel que Microsoft Exchange et IBM Lotus Dimino.

#### Windows Mobile de Microsoft

Il est conçu pour des appareils mobiles tels que les Smartphones ou Pocket PC. Ce système est largement distribué en Asie. Il permet à des logiciels Microsoft tels que Microsoft Office ou Windows Live Messenger de fonctionner sur un téléphone. Il permet de recevoir des courriels en temps réel,

La dernière version annoncée par Microsoft est la version 7. Elle a été bien améliorée au point de vue interface, ergonomies et fonctionnalités.

**Tableau 2 :** Table de critère entre les trois solutions :

Le tableau ci-dessous représente les 3 OS notés de 1 étoile (\*) pour le score le plus faible, à 3 étoiles (\*\*\*) pour le score le plus élevé, en fonction de critères comparatifs.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Android | iOS | Windows  Phone |
| Homogénéité | **\*\*** | **\*** | **\*\*\*** |
| Flexibilité | **\*\*\*** | **\*** | **\*\*\*** |
| Feedback | **\*\*\*** | **\*\*** | **\*\*** |
| Facilité des interactions | **\*\*** | **\*** | **\*\*** |
| Guidage | **\*** | **\*** | **\*\*** |
| Saisie des formulaires | **\*\*** | **\*\*\*** | **\*** |
| Accessibilité  aux commandes | **\*** | **\*\*\*** | **\*\*\*** |
| Densité informationnelle | **\*\*\*** | **\*** | **\*** |

Cette étude nous montre que les 3 OS mobiles permettent aux utilisateurs d’exécuter leurs tâches sans trop de difficultés. Néanmoins, Android ressort comme le système

le plus facile à prendre en main et permettant d’exécuter les tâches le plus rapidement pour les utilisateurs réguliers.

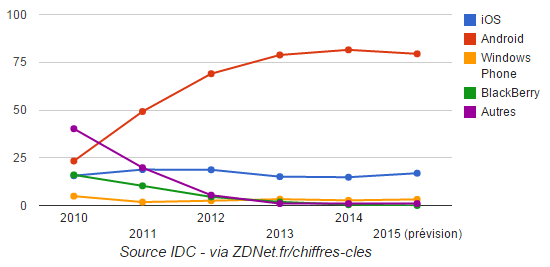


Figure 3 : **Part de marché mondiale des OS mobiles (%)**

**Android à son plus haut en 2014 -** La domination d’Android sur le marché mondial des smartphones est indiscutable. Et en 2014 cette prépondérance n’a pas été remise en cause comme en témoignent les [derniers indicateurs](http://www.idc.com/getdoc.jsp?containerId=prUS25450615) communiqués par IDC ([International Data Conseil](https://fr.wikipedia.org/wiki/International_Data_Conseil))

🡺Dans notre cas, nous avons choisi de développer une application mobile à base de l’Android.

## Système d’exploitation Android

### Définition

C’est un SE (Système d’exploitation) open source qui utilise le noyau Linux dédié au Smartphone, les tablettes tactile, PDA et quelques terminaux mobiles. Un projet racheté par Google, est annoncé officiellement le 5 novembre 2007 [2, 8].

Ce produit se basera sur la gratuité des applications fournies pour attirer les utilisateurs. Dans un premier temps, ce sont les constructeurs et les opérateurs Télécom qui se sont regroupés dans un consortium pour encadrer le développement d’Android : l’Open Handset Alliance. Plutôt que de construire un unique téléphone, Google a fait le choix de créer l’OS qui pourra être utilisé sur tous les futurs téléphones. Android SDK fournit

les outils et les API nécessaires de commencer à développer des applications sur la plate-forme Android utilisant le langage Java.

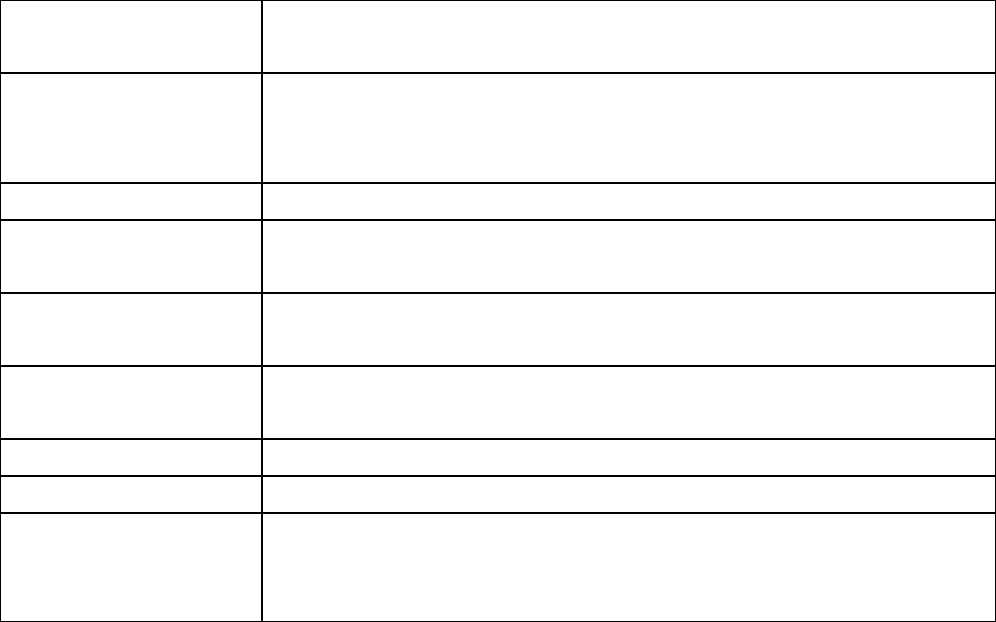
### Caractéristiques

Comme tout SE mobile, Andorid offre un nombre important des fonctionnalités communes avec les autres SE mobiles similaires tels que l’agenda, calendrier, l’interac- tion avec le matériel (capteur,boussle ..). Le système android a son propre innovation tels que les Widget, le bureau virtuel,la centralisation des notifications et le fil de dis- cussions des messages.

Le store en ligne de Google pour Android connu sous le nom de Google Market per- met à l’utilisateur d’Android de télécharger des jeux, des applications, des livres, de la musique et d’autres choses payantes ou gratuites. En septembre 2011, il y avait plus de 520 000 applications sur Android Market, dont 65 % sont gratuites[6]. Le 6 mars 2012 le Google Market a été remplacé par Google Play.

Plusieurs outils de développement Android sont disponibles citant par exemple Eclipse qui utilise Le SDK proposé par Google avec un nombre important des exemples et un simulateur virtuel. On peut utiliser aussi NetBeans, Sencha SDK Tools .Le déve- loppement Android n’est pas limité à un système d’exploitation spécifique. Les SDK de Android ainsi les outils de développement sont disponibles sur Windows, Linux et Mac OS. Les applications Andorid sont généralement écrit en Java et C++ et il ya des applications qui sont écrit en HTML5 CSS3 et JavaScript.

Le système d’exploitation Android est doté d’un ensemble de caractéristiques faisant de lui un terrain d’application lucratif. Les principales caractéristiques sont résumées dans le Tableau 1.2.

 **Tableau 3 :** – Les Caractéristiques du système d’exploitation Android

**Framework** Framework Java pour le développement d’application pour la pla- teforme Android

#### Machine virtuelle Dalvik

Machine virtuelle spécialement développée pour Android. Cette ma- chine virtuelle permet d’exécuter les applications java développées avec le Framework.

**Navigateur web** Navigateur web basé sur le moteur de rendu Webkit

**Graphique** Librarie graphique 2D, librarie graphique 3D basé sur OpenGL ES

1.0. Accélération matériel possible.

**Stockage** Base de données SQL : SQLite est utilisé pour le stockage des don- nées

**Média** Android supporte les formats audio/video/image suivants : MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF

**Connectivité** gsm, edge, 3G, bluetooth, wifi

**Support Matériel** Android est capable d’utiliser Camera, GPS, accéléromètre

#### Environnement de développement

Android possède un environnement de développement complet contenant : un émulateur, un débuggeur, un analyseur de mémoires et de performances et un plugin eclipse.

### Architecture

Android bénéficie d’une architecture en couche complète faisant de lui une plate- forme riche, dédiée aux appareils mobiles. Il est à base de noyau linux profitant des services système de base tels que la sécurité, la gestion mémoire, gestion de processus, etc. À un niveau supérieur se trouvent un ensemble de librairies écrites en C/C++ jouant le rôle d’un middleware. C’est sur cette couche que se greffe l’Android Run- time, comprenant la machine virtuelle java et ses bibliothèques. Ensuite on trouve la plateforme logicielle, nommée aussi Framework de développement, écrite en java et permettant de mutualiser les ressources entre applications Java. Elle offre aux déve- loppeurs la possibilité de produire des applications diverses et innovantes à travers un ensemble d’API. Enfin, et à un niveau plus supérieur se situe un ensemble d’appli- cations sous forme de paquets .apk. Les applications fournies par Android sont telles qu’un navigateur web, un client mail, un calendrier, un gestionnaire de contacts, etc. Cette architecture est synthétisée sur la Figure NN.

******

Figure 3– Architecture du système Android

L’étude de l’existant nous a permis de dégager plusieurs anomalies que nous avons détaillées dans la section précédente. Pour faire recours à ces anomalies nous proposons de concevoir et d’implémenter une application Android, qui donne la possibilité aux utilisateurs de consulter plusieurs bâtiments et de piloter ses différents équipements d’une manière rapide et facile.

**Conclusion**

Dans ce chapitre, nous avons défini le champ de notre étude suivi d’une étude de l’existant afin de préciser nos objectifs à atteindre.

En effet, l’étude de l’existant est essentielle pour bien spécifier l’application qui sera réalisée. Cette étude représente le cadre général de la conception de l’application.

**Analyse et conception**

## 

## 

## Introduction

Dans le deuxième chapitre, nous nous intéressons à une étape fondamentale dans le développement de notre application illustrée à travers l’étude conceptuelle. Nous présentons en premier lieu les processus de modélisation ainsi que la validation du choix du processus opté. Nous développons par la suite la stratégie suivie en présentant son principe ainsi que l’exploitation des différents diagrammes et phases de conception.

## Processus de modélisation et choix du processus

Afin de rendre la tâche de conception et le développement des logiciels plus flexible, plusieurs solutions et outils ont été proposés. Les processus de modélisation sont des solutions qui assurent à la fois aux entreprises, au chef de projet ou le développeur un ensemble des stratégies et des principes. Ils peuvent bien organiser ou gérer les taches ou le travail en mettant en considération le développement et l’apparition de nouvelles technologies. Ainsi, nous présentons les différents processus de développement ainsi que la validation de notre choix.

### Processus unifié

Le processus unifié (Unified Process UP) est un processus de développement logi- ciel itératif, centré sur l’architecture, piloté par des cas d’utilisation et orienté vers la diminution des risques. C’est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d’application, à différents types d’en- treprises, à différents niveaux de compétences et à différentes tailles de l’entreprise.

Les UP sont caractérisés par leur aspect itératif, centré sur l’architecture et piloté par les cas d’utilisation du langage de modélisation UML (Unified Modeling Language).

Les processus unifiés comportent cinq principales activités qui sont l’expression des besoins, l’analyse, le test et deux autres phases exprimées par l’analyse des besoins et l’élaboration

Les plus célèbres implémentations de la méthode UP sont : RUP (Rational Unified Process) et 2TUP (2 track unified process).

### Méthode Agile

La "modélisation agile", prônée par Scott Ambler s’appuie sur des principes simples et de bon sens. En effet, on doit avoir une grande palette de techniques à votre disposi- tion et connaître les forces et les faiblesses de chacune de manière à pouvoir appliquer la meilleure au problème courant. Il est aussi possible de changer de diagramme quand on se sent qu’on n’avance plus avec le modèle en cours. Le changement de perspective permet de voir le problème sous un autre angle et de mieux comprendre ce qui blo- quait précédemment. On se trouve souvent plus productif si on crée plusieurs modèles simultanément plutôt qu’en on focalise sur un seul type de diagramme.

Le terme "agile" fait référence à la capacité d’adaptation aux changements de contexte et aux modifications de spécifications intervenant pendant le processus de dévelop- pement. La spécification vise à réduire le cycle de vie du logiciel en accélérant son développement. En fait, on développe une version minimale, puis en intègre les fonc- tionnalités par un processus itératif basé sur une écoute client et des tests tout au long du cycle de développement.

On peut dégager quatre valeurs principales de la méthode agile :

**L’équipe** :"Les individus et leurs interactions, plus que les processus et les outils" [1]) : dans la méthode agile l’équipe de travail a une grande valeur même plus importante que ces outils (structurants ou de contrôle) ou les procédures de fonctionnement. Avoir une équipe de développeurs qui communiquent entre eux et qui ont une même vision sur les problèmes est plus mieux qu’une équipe d’experts où chacun voit les choses de sa vision de manière isolée.

**L’application** ("Des logiciels opérationnels, plus qu’une documentation exhaustive" [1]) : avoir une application qui tourne et qui fonctionne correctement est le but prin- cipale. Le reste el les documentations techniques sont important mais non pas un but principal .Une documentation technique précise, claire et à jour, forme une source d’aide très importante. Il est référable qu’elle soit des commentaires sur le code pour un gain de temps et de ressource.

**La collaboration** ("La collaboration avec les clients, plus que la négociation contrac- tuelle" [1]) : le client forme une partie très importante dans le développement de l’ap- plication. La modélisation agile préfère considérer le client comme un élément collabo- rateur qui fournit son avis (feed-back) sur l’adaptation de l’application vise à vis de ses attentes, que de négocier un contrat en début de projet puis négliger les demandes de ce dernier.

**L’acceptation du changement** ("L’adaptation au changement, plus que le suivi d’un plan" [1]) : la planification initiale et la structure du logiciel doivent être flexibles afin

de permettre l’évolution de la demande du client tout au long du projet. Les premières livraisons du logiciel vont souvent provoquer des demandes d’évolution. **eXP(eXtrem Programming** : été mise en place par Kent Beck, en collaboration avec Ward Cunningham et Ron Jeffries sur un projet pour la compagnie Chrysler dans les années 1990. C’est en 1999 qu’est officialisée la méthode XP avec la sortie du livre "eXtreme Programming explained"[4]. Elle recherche l’efficacité maximale en concen- trant l’effort de travail sur l’objectif de développer le bon logiciel et de ne pas s’égarer. La démarche est légère, pragmatique, disciplinée, empirique et adaptative[5].

L’XP améliore un projet de développement en cinq points essentiels : la communica- tion, la simplicité, la rétroaction, le respect et le courage.

Le XP offre plusieurs spécificités de telle façon qu’il devient la méthode agile le plus connus et les critères dont lesquels s’est basé notre choix. Parmi ses spécificités on cite :

* Xp est utilisé pour les projets de petite et moyenne taille.
* XP considère tout membre de l’équipe (chef de projet, développeur etc.) sont égaux.
* Communication orale au sein d’une petite équipe mettant le client au centre du projet. La rédaction de spécifications se fait au fil des sprints. Un ensemble de bonnes pratiques permet un travail de qualité.

### Validation de choix eXtrem Programming

Le choix de la méthode agile était basé sur plusieurs avantages citant :

* + - * Flexibilité : la méthode agile assure une démarche de développement souple et facile qui permet aux développeurs de faire un retour en arrière, une mise à jour ou une maintenance à n’importe quelle étape de développement.
      * Simplicité : la méthode agile suit un principe qui essaie de réduire le maximum

la complexité de développement en réduire les outils utilisés en se concentrant sur l’essentielle de développement.

* + - * Un processus dédié au projet avec une équipe de nombre réduit.
      * Réduire le cycle de vie de développement du logiciel.

L’adaptation avec l’évolution exponentielle des technologies.

**1 Capture des besoins**

**1.1. Identification des acteurs**

a. Les acteurs

« *Un acteur représente l’abstraction d’un rôle joué par des entités externes (utilisateur,*

*dispositif matériel ou autre système) qui interagissent directement avec le système étudié.* » [mg]

Tous simplement un acteur est une entité physique (personne) ou abstraite (logiciel)

capable d’utilisée le système afin de répondre à un besoin bien définit. Les acteurs de notre

application sont :

* **Utilisateur** : Il désigne la personne ou l'entité qui prend la décision de consulte les états des équipements, pilote les différent équipements.
* **Administrateur** : l’administrateur a accès à toutes les fonctionnalités du système sans limite, ces fonctionnalités qui sont principalement la gestion des maisons, les endroits, les équipements.

**2. Les besoins fonctionnels**

Les besoins fonctionnels ou les cas d’utilisations en terme d’UML peuvent être définis comme suit : « *Un cas d’utilisation (use case) représente un ensemble de séquences d’actions réalisées* *par le système et produisant un résultat observable intéressant pour un acteur particulier.* » [MG]

Un cas d’utilisation est une suite d’actions effectuées par le système afin de répondre à

une demande d’un utilisateur (acteur). Dans ce qui suit, nous décrivons les différents besoins fonctionnels de notre système par acteur :

**Utilisateur :** cet acteur attend les fonctionnalités suivantes au système :

* Authentification
* Consulter l’état des équipements électriques
* Piloter les différents équipements électriques
* Consulter température et humidité

 **Administrateur** : cet acteur se focalise essentiellement sur les fonctionnalités

* Gérer les maisons, les endroits.
* Gérer les endroits.
* Gérer les équipements.

**3. Les besoins non fonctionnels**

Les besoins non fonctionnels sont des besoins qui ont un aspect visible pour l’utilisateur, mais qui ne sont pas reliés directement au comportement du système. Les besoins non fonctionnels de notre système se décrivent comme suit :

* **De la sécurité**

La solution proposée permet à l’utilisateur une navigation sécurisée. Elle n’est accessible qu’avec une authentification.

* **Besoins d’utilisation :**

Tous les standards d’ergonomies doivent être présents : interface utilisateur bien claire et simple dans l’utilisation.

* **Besoins de performance :** il s’agit d’optimiser le temps de chargements des pages par l’utilisation des bonnes pratiques du développement. [9]

**1 .2.Diagramme de contexte statique**

Ce diagramme d’UML permet simplement de montrer la relation des différents acteurs avec le système. Il spécifie le nombre d’instances de chaque acteur relié au système à un

moment donné comme il est indiqué dans la figure

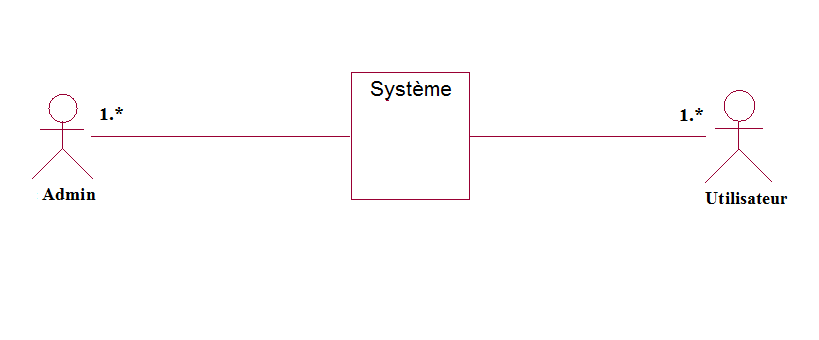


Figure 5 **: Diagramme de contexte statique**

Pour expliquer le diagramme ci-dessus, nous pouvons dire qu’à un instant *t* nous pouvons

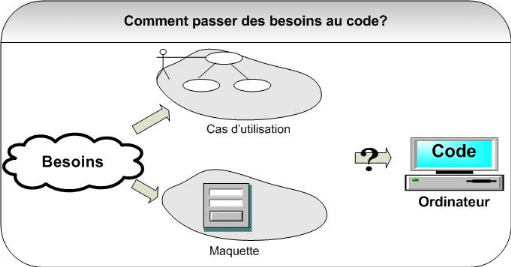
avoir 0 ou plusieurs administrateurs qui manipulent l’application, et 0 ou plusieurs utilisateur qui sont en train d’utiliser l’application.

## Conception de l’application

### Stratégie de conception

Comme on a déjà mentionné précédemment, nous suivons la méthode agile.

Le principe suivi consiste à passer des besoins de l’utilisateur jusqu’à l’obtention du code de l’application. Donc à partir du besoin nous essayons de crocher le point de départ de la conception. Pour cela nous avons besoin d’un langage de modélisation or les méthodes agile s’appuient sur l’existant pour fournir des outils flexible.Ainsi UML (Unified Modeling Language)[11], cet langage de modélisation qui permet d’élaborer des modèle objet indépendamment du langage de programmation est la meilleure solution. D’après ce qu’on a mentionné, le point de départ ou l’expression préliminaire des besoins donne lieu à une modélisation par les cas d’utilisation puisque c’est le diagramme de départ, et une maquette homme-machine (IHM : Interactions Home-Machine),



Etape 1

**1.Élaboration du diagramme de cas d’utilisation général**

Un cas d’utilisation correspond à un certain nombre d’actions que le système devra exécuter en réponse à un besoin d’un acteur.

Un cas d’utilisation doit produire un résultat observable pour un ou plusieurs acteurs ou parties prenantes du système.

Nous commençons par présenter le diagramme de classe d’une manière générale, pour avoir une vue globale du système.

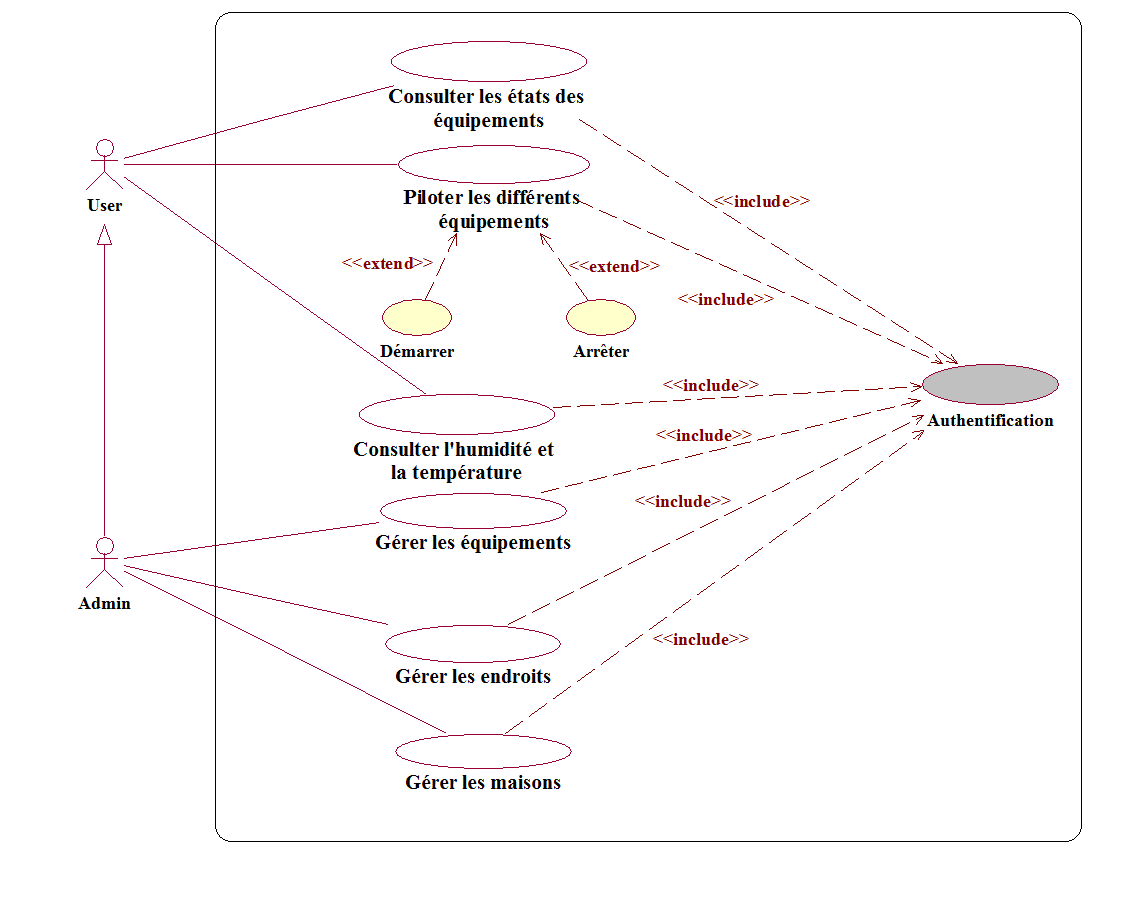


Figure 6**: diagramme de cas d'utilisation global**

* ***Description textuelle :***

**Cas d’utilisation : Authentification**

|  |  |
| --- | --- |
| **Description** |  |
| **Titre** | Authentification |
| **But** | Authentification et autorisation d’accès |
| **Acteur** | Utilisateur |
| **Précondition** | Lancer l’application Android |
| **Post condition** | L’utilisateur est connecté au système est dirigé vers la  section qui lui convient |
| **Scénario nominal** | **1.** Saisie du login et mot de passe  **2.** Connexion  **3.** Le système vérifie leurs coordonnées.  **4.** Le système affiche la page d’accueil |
| **Enchainement d’exception** |  Le login ou le mot de passe est incorrects.   Le système informe l’acteur que les données saisies sont erronées et le scénario reprend au point 1 du scénario nominal. |

**Cas d’utilisation :**Consulter état d’équipement

|  |  |
| --- | --- |
| **Description** | |
| **Titre** | Consulter état d’équipement |
| **But** | Visualiser l’état des équipements de chaque  Chambre |
| **Acteur** | Utilisateur |
| **Précondition** | L’utilisateur doit être identifié. |
| **Post condition** | Affichage du l’état de l’équipement choisi. |
| **Scénario nominal** | **1.** Le système affiche la liste des bâtiments.  **2.** L’utilisateur sélectionne un bâtiment.  **3.** Le système affiche la liste des équipements.  **4.** L’utilisateur sélectionne l’équipement voulu.  **5.** Le système affiche la liste des chambres  liées à l’équipement choisi.  **6.** Le système renvoie l’état de cet équipement. |
| **Enchainement d’exception** |  Le système affiche un message d’erreur indiquant à l’utilisateur qu’il y a un problème de connexion. |

**Cas d’utilisation : arrêter l’équipement d’éclairage**

|  |  |
| --- | --- |
| **Description** | |
| **Titre** | arrêter l’équipement d’éclairage. |
| **But** | l’utilisateur arrête l’équipement d’éclairage. |
| **Acteur** | Utilisateur |
| **Précondition** | L’utilisateur doit être identifié. |
| **Post condition** | Modification de l’état de l’équipement choisi |
| **Scénario nominal** | **1.** Le système affiche la liste des bâtiments.  **2.** L’utilisateur sélectionne un bâtiment.  **3.** Le système affiche la liste des équipements.  **4.** L’utilisateur sélectionne l’équipement voulu.  **5.** Le système affiche la liste des chambres liées à  l’équipement choisi.  **6.** Le système renvoie l’état de cet équipement (ON).  **7.** L’utilisateur cliquer sur OFF.  **8.** Le système renvoi un message « lampe éteinte » |
| **Enchainement d’exception** |  Le système affiche un message d’erreur indiquant à l’utilisateur  qu’il y a un problème de connexion. |

|  |  |
| --- | --- |
| **Cas d’utilisation** | **Gérer les équipements** |
| **Acteur :** | Administrateur |
| **But:** | Ce cas d’utilisation est utilisé pour gérer les équipements |
| **Pré condition :** | L’administrateur doit s’authentifier |
| **Post condition :** | La mise à jour de l’équipement est effectuée dans le système |
|  |  |
| **Scénario nominal :** | * Ajouter *équipement* :   L’administrateur choisit l’interface d’ajout et le processus se poursuit selon les étapes suivantes :   1. L’administrateur lance l’interface contenant la liste des équipements. 2. L’administrateur demande l’ajout d’un équipement. 3. Le système affiche le formulaire pour saisir les informations relatives à l’équipement. 4. L’administrateur saisit les informations nécessaires. 5. L’administrateur demande la validation de l’ajout. 6. Le système vérifier les données saisies, s’il y a une information manquante, il exécute exception1, en cas d’erreur, il exécute exception 2, sinon le système ajoute l’équipement dans la base de données. |
| **Scénario alternatifs :** | * Modifier un équipement :   Modifier l’information d’un équipement :   1. L’administrateur demande de modifier les données d’un équipement. 2. Le système affiche le formulaire et propose à L’administrateur de chercher l’équipement selon leur nom. 3. L’administrateur saisit le critère de recherche. 4. Le système affiche toutes les informations sur cet équipement, sinon le système exécute l’exception 2. 5. L’administrateur saisit les modifications à apporter. 6. Le système vérifie les informations si sont correctes et les enregistre dans la base de données. 7. Le système enregistre les données modifiées après validation  * Supprimer un équipement :   Suppression d’un équipement:   1. L’administrateur lance la liste des équipements. 2. Le système affiche le formulaire de gérer équipement. 3. L’administrateur cherche l’équipement correspondant. 4. Le système affiche toutes les informations sur cet équipement, sinon le système exécute l’exception 3 5. L’administrateur supprime l équipement. |
| **Exceptions :** | 1. Le système affiche le message : « L’information est manquante : il faut remplir tous les champs » et le cas reprend à l’endroit correspondant à l’erreur. 2. Le système affiche le message : « L’information erronée : champ invalide vérifie la saisie » et le cas reprend à l’endroit correspondant à l’erreur. 3. Le système affiche le message : «équipement inexistant». |

Après la présentation des diagrammes de cas d’utilisation, nous nous intéressons maintenant à la deuxième partie qui est la présentation des IHM qui seront détaillée d’avantages dans le chapitre suivant pendant la phase d’implémentation.

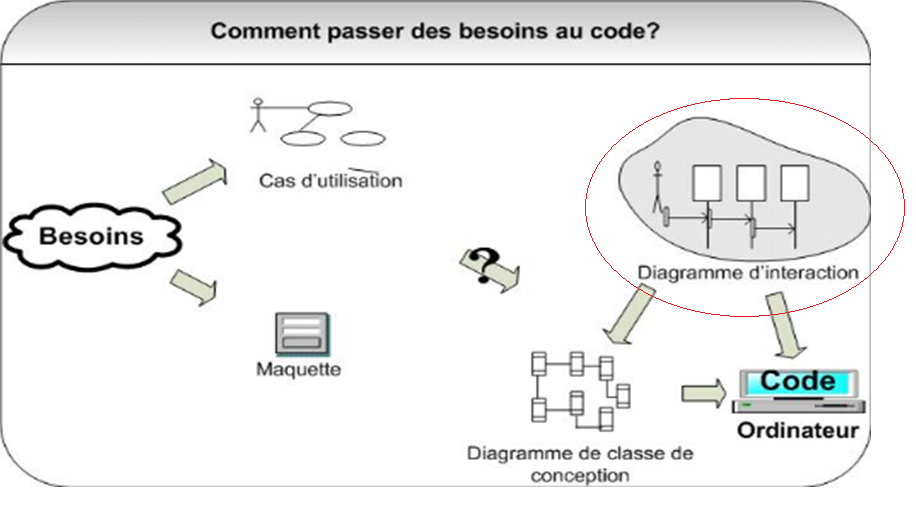
Dans la section suivante, nous passons de présentation générale à la présentation dé- taillée.

### Diagramme de séquence

Nous décrivons maintenant de façon détaillée les cas d’utilisation que nous avons déjà définit dans la section précédente. Le diagramme de séquence présente une pré- sentation graphique UML très utile pour compléter la description textuelle des cas d’utilisation.

#### Définition

Le diagramme de séquence est une présentation graphique UML qui permet de présenter les interactions entres les objets par chaque cas d’utilisation.



Etape 2



**diagramme de séquence «Authentification »**

**Diagramme de séquence : « consulter les états des différents équipements »**

L’utilisateur peut savoir l’état d’un équipement de chaque bâtiment dans chaque chambre.



**diagramme de séquence « consulter état d'un équipement »**

**Diagramme de séquence « Activer équipement »**

Pour activer un équipement électrique, l’utilisateur doit tout d’abord saisir son login et son mot de passe. Ensuite, il demande d’afficher l’interface de la liste des équipements électriques et il choisit l’équipement qu’il veut l’activer. Après que les actions associées à cet équipement sont affichées, il sélectionne une chambre parmi la liste puis il clique sur le bouton d’activation « ON ». Enfin, le message « équipement activé » sera affiché.



**diagramme de séquence «activer équipement »**

**Diagramme de classe :**

Les diagrammes de classe représentent la structure statique d'un système. Il contient principalement les classes ainsi que leurs associations mais on peut aussi y trouver des objets. En pratique, l'intérêt majeur du diagramme de classes est de modéliser les entités du système d’information. [rn]

**Dictionnaire des donnes :**

Un dictionnaire des données regroupe toutes les propriétés de toutes les classes de notre système. Dans ce tableau illustratif on retrouve les attributs qu’on a utilisé leurs significations.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Code** | **Classe** | **N°** | | **Attributs** | **Désignation** | |
| 01 | Utilisateur | 01 | | Id\_user | Identifiant de l’utilisateur | |
| 02 | | Nom\_user | Nom de l’utilisateur | |
| 03 | | Prénom\_user | Prénom de l’utilisateur | |
| 04 | | Mdp\_user | Mot de passe | |
| 05 | | Email\_user | Adresse mail | |
| 06 | | Etat\_user | Etat d’activation du compte | |
| 07 | | Tel\_user | Numéro de téléphone | |
| 02 | Bâtiment | 08 | | Id\_batm | Identifiant du bâtiment | |
| 09 | | Adr\_batm | Adresse de bâtiment | |
| 10 | | Latitude\_batm | Latitude de bâtiment | |
| 11 | | Longitude\_batm | Longitude de bâtiment | |
| 03 | Chambre | 12 | | Id\_ch | Identifiant chambre | |
| 13 | | Description\_ch | Description de chambre | |
|  |  | | 14 | Humidité\_ch | Humidité de chambre |
| 15 | Température\_ch | Température de chambre |
| 04 | Equipement | | 16 | Id | Identifiant de l’équipement |
| 17 | libellé | Libellé de l’équipement |
| 18 | Etat | Etat équipement |
| 05 | Lampe | | 19 | Id\_lampe | Identification de lampe |
| 06 | Climatiseur | | 20 | Id\_clim | Identification de climatiseur |
| 07 | Volet | | 21 | Id\_volet | Identification de volet |

**Tableau : Dictionnaire de données**

**Représentation des méthodes**

Les méthodes d’un objet caractérisent son comportement, c’est-à-dire l’ensemble des actions (appelés opération). Ces opérations permettent de faire régir l’objet aux sollicitations extérieur (ou d’agir sur les autres objets). De plus, les opérations sont étroitement liées aux attributs, car leurs actions peuvent dépendre des valeurs des attributs, ou bien les modifier.

Dans cette section, nous allons présenter un tableau décrivant les différentes méthodes de notre application.

|  |  |
| --- | --- |
| **Méthode** | **Description** |
| Affiche\_listeBâtiment () | Affiche la liste des bâtiments disponibles |
| Affiche\_listeChambre () | Affiche la liste des chambres disponibles |
| Activer\_lampe () | Allumer une lampe |
| Fermer\_lampe () | Eteindre une lampe |
| Activer\_ventilateur () | Démarrer un ventilateur |
| Fermer\_ventilateur () | Fermer un ventilateur |
| Ouvrir\_volet () | Ouvrir un volet |
| Fermer\_volet () | Fermer un volet |
| Authentification () | Authentification des utilisateurs |

**Tableau : Représentation détaillée des méthodes**

**Représentation des associations**

|  |  |  |
| --- | --- | --- |
| **Désignation** | **Classe participantes** | **Multiplicité** |
| **Piloter** | Utilisateur  équipement | **1..\***  **1..\*** |
| **Consulter** | Utilisateur  bâtiment | 1..\*  1..\* |

**Tableau : Représentation détaillée des associations**