

# SecureBike

Avec : LUCAS Michaël, PAGÉGIE Léo, CORNET Jérémie et MISIAK Roman

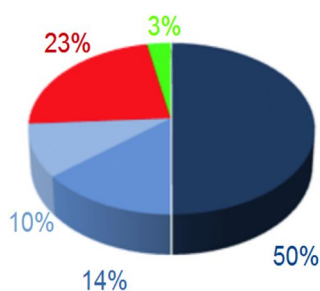
## Sommaire

1) Présentation du projet : .....	1
Problématique : .....	1
Analyse de compétitivité : .....	2
2) Analyse des besoins : .....	2
3) Planification et répartition des tâches : .....	4
Tâches personnelles : .....	5
4) Analyse fonctionnelle : .....	5
5) Solution technique : .....	6
6) Intégration et validation : .....	10
7) Conclusion : .....	10

## 1) PRESENTATION DU PROJET :

### **Problématique :**

Aujourd'hui en France, on compte 1 000 vols de vélos par jour soit près de 400 000 par an. 50% des français se sont déjà fait voler au moins un vélo. On estime à 23% le pourcentage de personnes qui ne rachèteront pas de vélo à la suite d'un vol de peur d'en être une nouvelle fois victime.



Chiffres de Janvier 2015

### **A la suite d'un vol de vélo :**

- 3% récupéreront leur vélo
- 10 % achèteront un vélo neuf avec un budget inférieur à 220 €
- 14 % achèteront un vélo neuf avec un budget supérieur à 220 €
- 23 % ne rachèteront jamais de vélo
- 50 % rachèteront un vélo d'occasion

Le *SecureBike* est un projet visant à sécuriser les vélos des utilisateurs grâce à un garage à vélos automatisé et sécurisé. Sa simplicité d'utilisation permet à toute personne de s'en servir ce qui facilite son implantation au sein d'un lycée par exemple. Pour ouvrir la porte et donc accéder au local, l'utilisateur doit posséder un code ou disposer d'un badge.

Le système possède aussi des antivols automatisés et équipés d'alarmes sonores.

Lors de l'ouverture de la porte, le système attribue un emplacement aux vélos de manière à ce que l'ouverture des antivols puisse être gérée automatiquement par le *SecureBike*.

## **Analyse de compétitivité :**

Des garages à vélos tels que le *SecureBike* existent déjà partout en France. Même si le concept n'est pas nouveau, le présent projet apporte une nouveauté avec ses antivols munis d'une alarme, gérés automatiquement par le système. Le projet est conçu pour être installé au lycée Aristide Briand, à Saint-Nazaire, ainsi nous allons pouvoir comparer le *SecureBike* et le garage à vélos mis en place à la gare de Saint Nazaire.

C'est la société Abri Plus qui a été en charge de la construction du garage à vélo de la gare de Saint Nazaire pour un coût total est de 21 658 €. Ce dernier se doit de respecter les tarifs de stationnement en parc en enclos imposés par la commune de Saint Nazaire.



STATIONNEMENT EN PARCS EN ENCLOS	1 heure	gratuit
	1 heure 30	0,50 €
• De 9h à 19h du lundi au samedi hors jours fériés.	2 heures	1 €
	2 heures 30	1,50 €
• La 1 <sup>ère</sup> heure est gratuite +1/2h gratuite sur la tranche horaire 12h-14h.	3 heures	2 €
	forfait journée à compter de 7h	10 €

La journée d'un étudiant au lycée est estimée à 10h. Celui-ci devrait donc dépenser 10 € par jour pour garer son vélo.

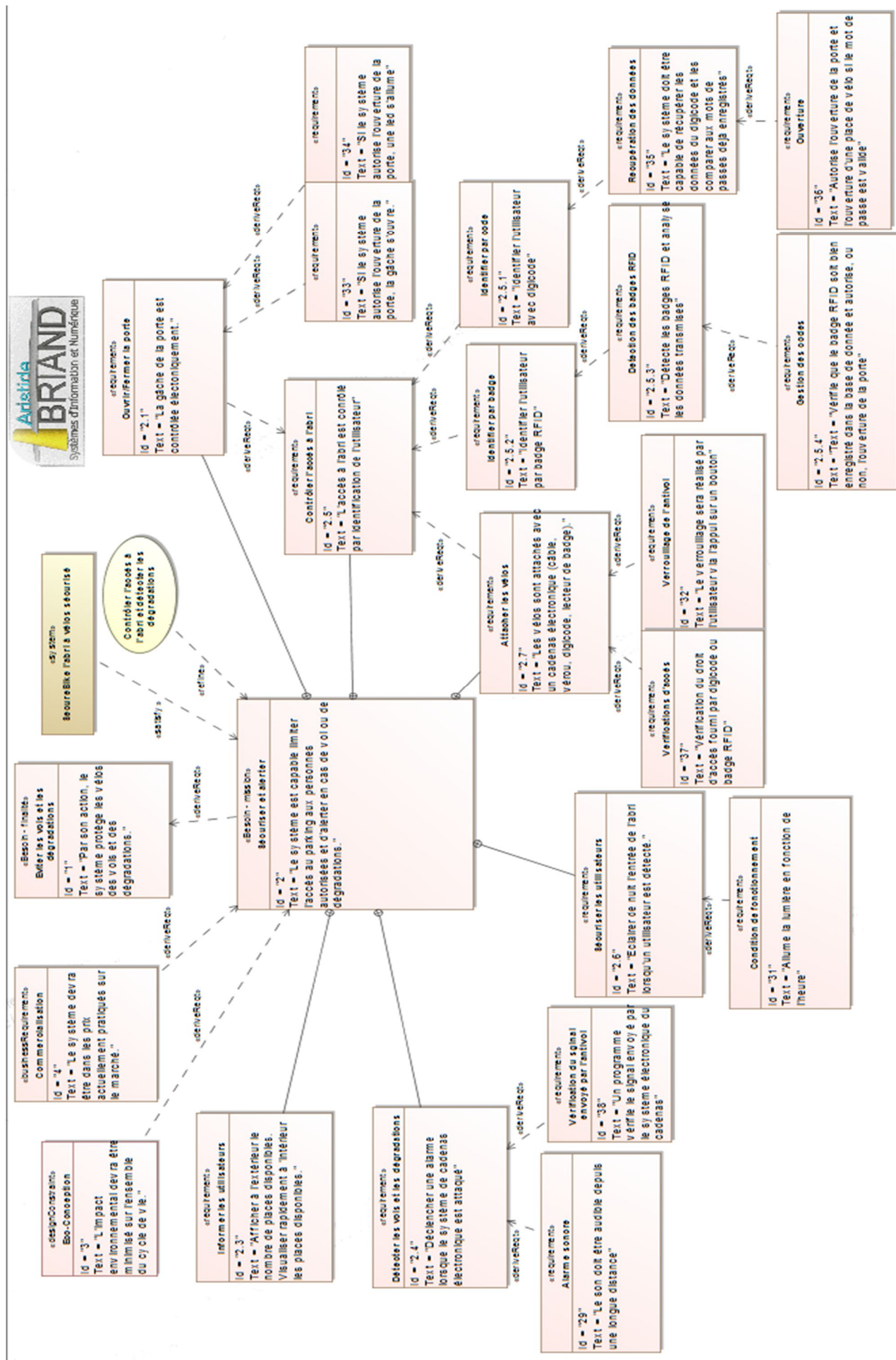
En 2015, il y avait 144 jours scolaires, son budget annuel de parking reviendrait à 1 440 €, une somme astronomique !

On peut estimer le coût de fabrication du projet au même prix de revient que le garage déjà existant à la gare de Saint Nazaire. Il est évident que notre projet n'a pas pour but de générer des bénéfices, à l'inverse du garage à vélos de la gare. C'est pourquoi le coût annuel de l'utilisation du parking à vélos sécurisé ne s'élève qu'à 3 €, soit le prix d'une carte de self au lycée. Ces 3 € permettront d'acheter un badge d'accès au local qui sera remis à l'utilisateur.

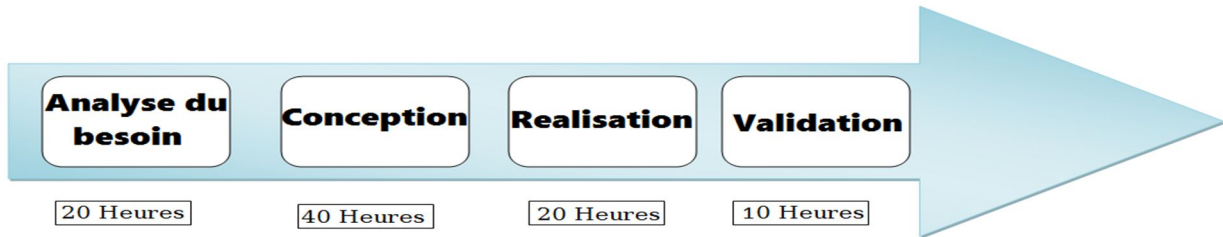
## **2) ANALYSE DES BESOINS :**

Notre projet se doit de répondre à un cahier des charges précis ainsi qu'à un diagramme d'exigence général déterminant les différentes parties du projet et comment les éléments du système doivent interagir entre eux.

Ce diagramme d'exigence aborde aussi bien l'aspect économique et écologique, que l'aspect technique (ex. : si le système autorise l'ouverture de la porte, la gâche s'ouvre et une LED s'allume (ID 2.1, 3.3, 3.4)).



### 3) PLANIFICATION ET REPARTITION DES TACHES :



Pour créer le projet *SecureBike*, nous disposons de 90h divisées en 4 phases.

- La phase d'analyse du besoin est une tâche effectuée en groupe. Elle sert à répartir le travail de chacun de manière à utiliser au mieux le temps imparti. Cette phase est principalement construite d'idées émises par le groupe, de réflexions et de création de diagrammes.

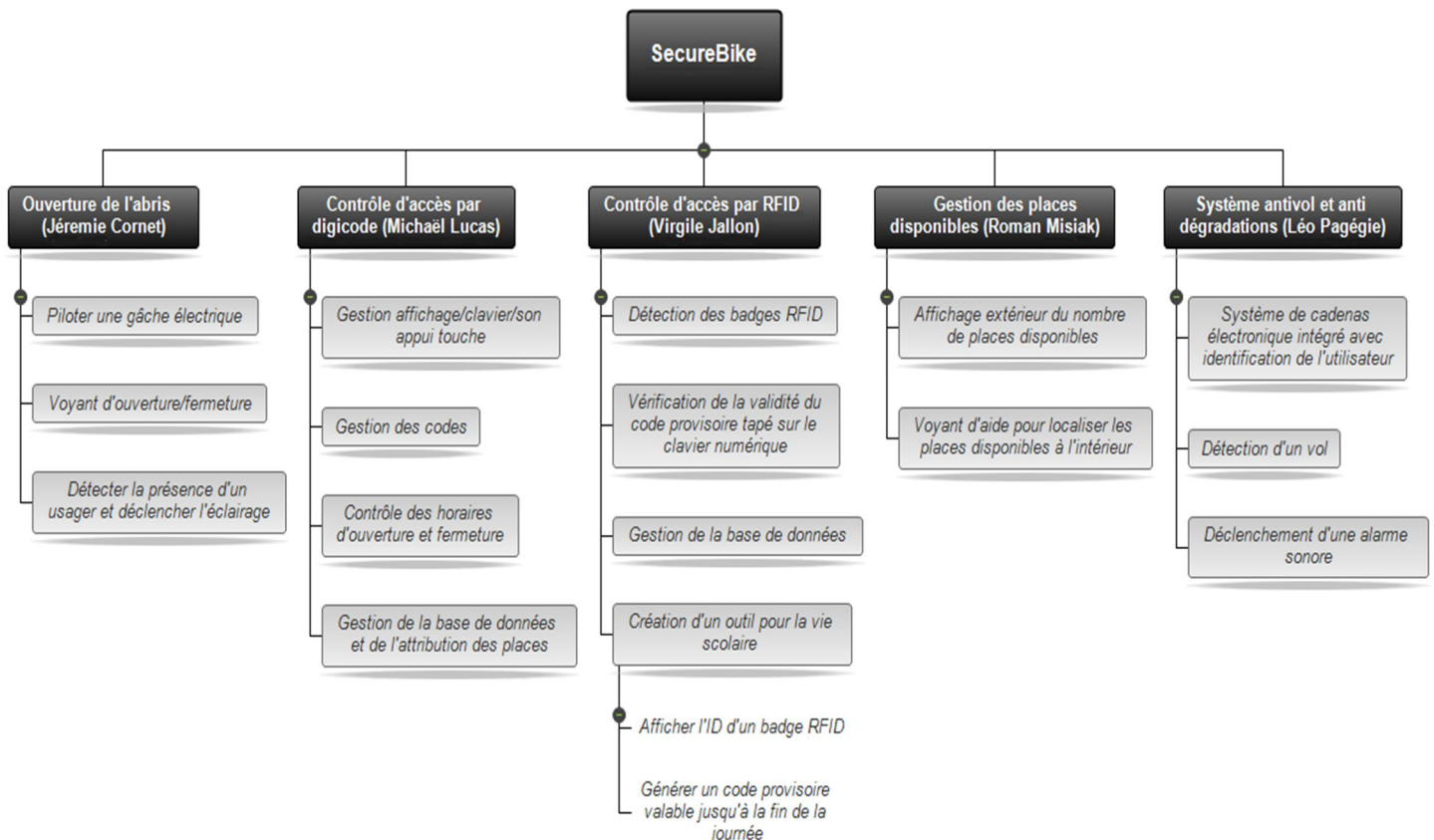
- les phases de conception et de réalisation sont deux phases qui se rejoignent.

Dans la première, chaque membre du groupe met en place ses idées en créant sa partie du projet, en réfléchissant à comment il va procéder et quels sont les résultats qu'il va obtenir.

Dans la seconde il crée sa partie en réalisant des câblages, des programmes, des calculs etc.

- La phase de validation permet de regrouper les parties réalisées par chacun des membres du groupe en une seule. On y fait un câblage commun ainsi que la mise en commun des programmes.

Lors de la phase d'analyse du besoin, nous avons réparti les tâches entre chacun des membres du groupe de la manière la plus judicieuse pour pouvoir finir le projet dans le temps imparti. Cette répartition a été établie selon le diagramme ci-dessous :





## **Tâches personnelles :**

Une fois les tâches réparties, il faut savoir exactement comment va fonctionner le système pour pouvoir le construire. Lors de sa première utilisation du SecureBike, l'utilisateur devra se rendre à la vie scolaire, laquelle lui remettra un badge RFID ainsi qu'un code provisoire valable jusqu'à la fin de la journée.

Il devra se rendre auprès du garage à vélos et composer le code fourni. Le système entamera une procédure de création de compte et demandera à l'utilisateur de présenter son badge RFID et de saisir un mot de passe à 5 chiffres qu'il devra retenir.

Le compte est ensuite sauvegardé et l'utilisateur n'aura plus qu'à présenter son badge ou taper son mot de passe pour accéder au local.

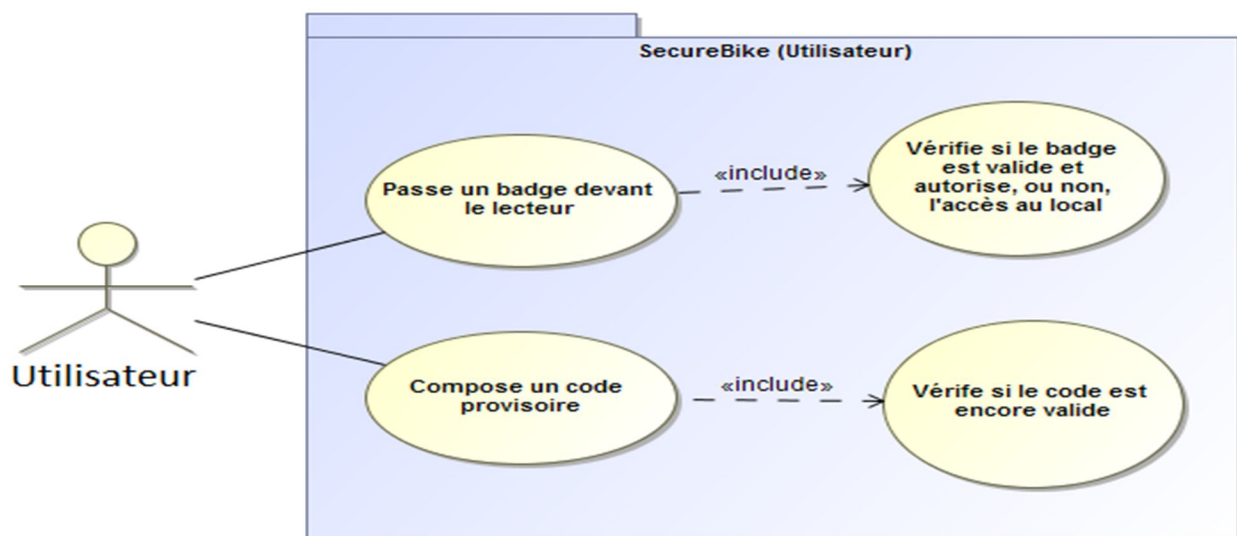
Ma tâche personnelle sera composée en deux parties, à savoir :

-Créer un boîtier pour la vie scolaire lequel permettra de voir l'ID des badges fournis aux utilisateurs et de générer le mot de passe provisoire.

-Vérifier si les badges présentés à la borne d'accès du parking à vélos sont valides et si le mot de passe provisoire saisi est encore valable.

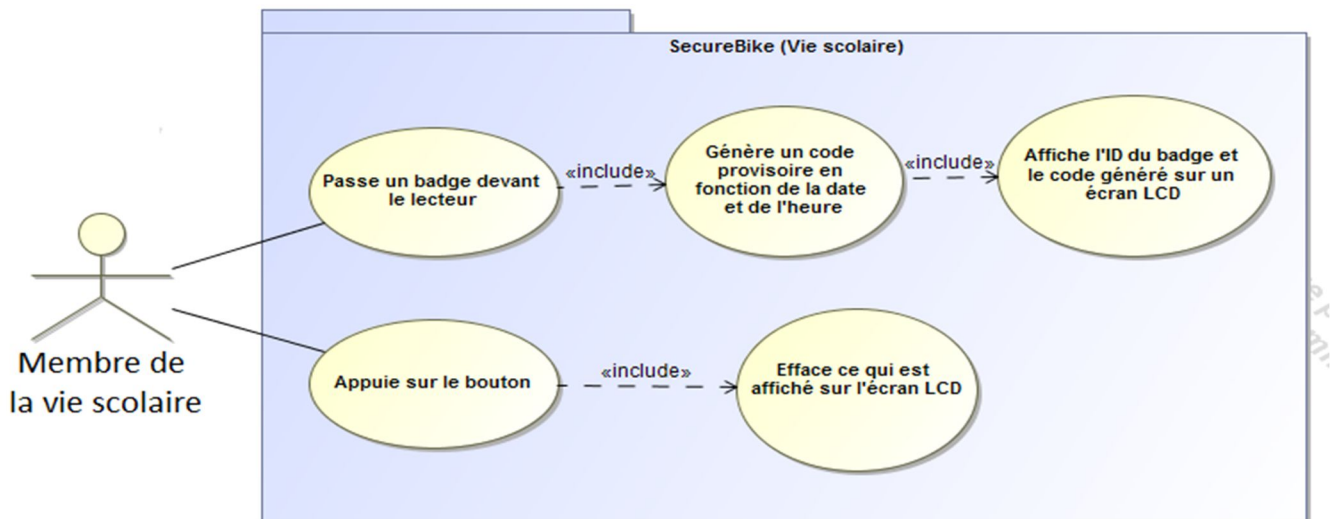
## **4) ANALYSE FONCTIONNELLE :**

Nous allons utiliser deux diagrammes de cas d'utilisation pour voir comment réagissent les deux systèmes au passage d'un badge RFID.



- Si l'utilisateur présente un badge devant le lecteur de badges de la borne d'accès du garage, le système va vérifier dans la base de données si le badge est bien valide. Si c'est le cas, il autorise l'ouverture de la porte du local.

- Si l'utilisateur compose un code provisoire et que celui-ci est encore valide, le système entamera une procédure de création de compte pour permettre à l'utilisateur d'accéder au garage à vélos.



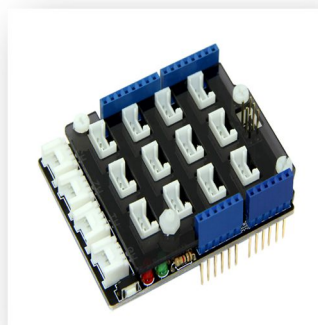
- Si un membre de la vie scolaire passe un badge devant le lecteur de l'outil destiné à la vie scolaire, le système générera un code provisoire et affichera l'ID du badge ainsi que le code provisoire sur un écran LCD. Ces informations permettent à la vie scolaire de tenir un registre sur les utilisateurs du *SecureBike*.
- Si un membre de la vie scolaire appuie sur le bouton, le texte affiché sur l'écran s'effacera.

## 5) SOLUTION TECHNIQUE :

Pour réaliser ma partie du projet, je vais utiliser une carte Arduino. Ces cartes électroniques ont la particularité d'être programmables. Pour faciliter le câblage des capteurs, je vais utiliser un module Grove (directement emboîtable sur la carte Arduino) qui me permettra de brancher des éléments de type Grove. Une carte Arduino classique est largement suffisante pour le boîtier de la vie scolaire mais il faudra une Arduino Mega pour le boîtier du garage à vélos afin de pouvoir brancher les composants de chaque membre du groupe.



Carte Arduino Uno Rev3

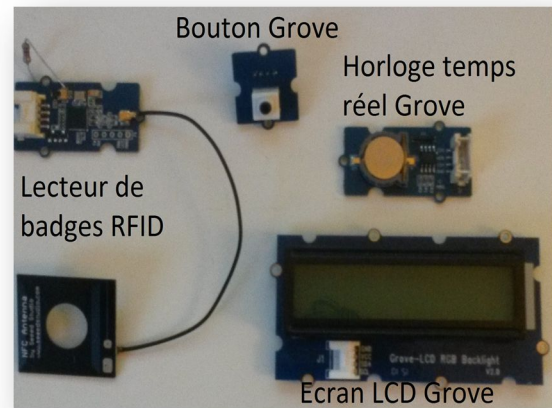


Module Grove



Câble Grove

Le boîtier destiné à la vie scolaire doit pouvoir détecter les badges présentés, générer un code provisoire en fonction de la date et de l'heure et enfin d'afficher l'ID et le code sur un écran qui s'effacera lors de l'appui sur le bouton. Pour ce faire, voici le matériel dont nous avons besoin.



Une fois les différents éléments branchés correctement, il suffit de les programmer. Le code ci-dessous est le code complet du boîtier destiné à la vie scolaire.

Dans un premier temps on déclare les bibliothèques qui vont être utilisées, suivie ensuite de l'instruction *passGenerate()* qui va générer le mot de passe provisoire. Suivie par l'instruction *scanTag()* qui va récupérer l'ID du tag présenté. Le *setup()* initialise l'horloge temps réel, le lecteur de badges RFID, le LCD ainsi que le port du bouton. Il définit aussi la vitesse d'exécution du programme.

```
//Bibliothèques//
#include <Wire.h>
#include "rgb_lcd.h"
rgb_lcd lcd;
#include "DS1307.h"
DS1307 clock;

#if 0
#include <SPI.h>
#include <PN532_SPI.h>
#include <PN532.h>
#include <NfcAdapter.h>
PN532_SPI pn532spi(SPI, 10);
NfcAdapter nfc = NfcAdapter(pn532spi);
#else
#include <Wire.h>
#include <PN532_I2C.h>
#include <PN532.h>
#include <NfcAdapter.h>
PN532_I2C pn532_i2c(Wire);
NfcAdapter nfc = NfcAdapter(pn532_i2c);
#endif
//////////

String YearS, MinuteS, DayOfMonthS, MonthS, HourS, GeneratedPass, UID;
int ButtonState, ButtonPin;

void passGenerate()
{
    clock.getTime(); //Récupère l'heure de l'horloge temps réel
    YearS=String(clock.year, DEC);
    MinuteS=String(clock.minute, DEC);
    DayOfMonthS=String(clock.dayOfMonth, DEC);
    MonthS=String(clock.month, DEC);
    HourS=String(clock.hour, DEC);
    //Exemple de transformation : Var='2'; Var devient Var='02'
    if(MinuteS.length()<2) {MinuteS="0"+MinuteS;}
    if(DayOfMonthS.length()<2) {DayOfMonthS="0"+DayOfMonthS;}
    if(MonthS.length()<2) {MonthS="0"+MonthS;}
    if(HourS.length()<2) {HourS="0"+HourS;}

    //Génère le mot de passe provisoire
    GeneratedPass=YearS+" "+MinuteS+" "+DayOfMonthS+" "+MonthS+" "+HourS;
    //DEBUGSerial.print("Generating code : "+GeneratedPass+"\n");
}

void scanTag()
{
    if (nfc.tagPresent()) //Vérifie si un badge est présenté
    {
        NfcTag Tag = nfc.read();
        Serial.print("ID du tag scanné : "+Tag.getUidString());
        passGenerate(); //Génère le mot de passe
        //Affiche l'ID du tag ainsi que le mot de passe provisoire sur le LCD
        lcd.clear();
        lcd.setCursor(0,0);
        GeneratedPass.replace(" ", "");
        UID=Tag.getUidString();
        UID.replace(" ", "");
        lcd.print(GeneratedPass);
        lcd.setCursor(0,1);
        lcd.print(UID);
        delay(2000);
    }
}

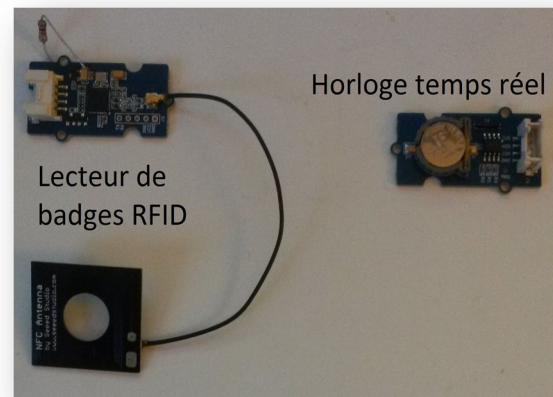
void setup()
{
    Serial.begin(9600);
    nfc.begin(); //Initialisation de lecteur de badges RFID
    lcd.begin(16, 2); //Initialisation du LCD

    //Initialisation de l'horloge temps réel
    clock.begin();
    clock.fillByYMD(2016,1,14);
    clock.fillByHMS(20,01,0);
    clock.fillDayOfWeek(THU);
    clock.setTime();
    lcd.clear();
    //Initialisation du port du bouton
    ButtonPin=2;
    pinMode(ButtonPin, INPUT);
}

void loop()
{
    scanTag();
    ButtonState = digitalRead(ButtonPin);
    if (ButtonState == HIGH)
    {
        lcd.clear();
    }
}
```

Voici un exemple de code que générera le système : si nous sommes le 17/3/2016 et qu'il est 9h30, le code sera 1630170309.

Passons maintenant à la partie directement reliée au garage à vélos. Pour cette partie nous utilisons un module SD qui va nous permettre de créer une base de données regroupant les informations des utilisateurs en les stockant sur une carte SD.



La base de données est séparée en 2 fichiers, MDP.txt et RFID.txt qui sont organisés de la manière suivante :

Fichier	Édition	Recherche
new 1 x		
1	001;12345;07;	
2	002;54321;00;	
3	003;15243;12;	
4	004;75992;03;	

Fichier	Édition	Recherche
new 1 x		
1	CB8BE60C;	
2	042B50C23E3E80;	
3	01E3B72F;	
4	D15A80C9A16;	

-Le premier fichier est parfaitement organisé, les informations de chaque utilisateur sont stockées sur 13 caractères (ID;MotDePasse;Place;).

-Le second n'est pas organisé de la même manière car on ne sait pas quelle est la longueur de l'ID d'un badge.

Le code suivant permet de savoir si le badge qui est passé devant le lecteur est présent dans la base de données.

J'avais créé un programme permettant de récupérer les informations de la base de données ligne par ligne mais nous avons finalement opté pour une solution plus simple, qui permet de stocker le contenu de la base de données dans une seule variable. Le programme suivant a donc été réalisé en fonction de la variable toutefois, ayant des doutes sur le fait que la variable puisse stocker autant de caractères, je mets ici un lien vers un de mes programmes qui pourra être utilisé si cette méthode ne fonctionne pas.

Partie 1 et 2 :

<https://www.dropbox.com/s/uxadupcg5bxrlrv/CodeBDDRecup1.PNG?dl=0>

<https://www.dropbox.com/s/mwq9pzb25prcxv/CodeBDDRecup2.PNG?dl=0>



```

///Bibliothèques///
#include <Wire.h>
#if 0
#include <SPI.h>
#include <PN532_SPI.h>
#include <PN532.h>
#include <NfcAdapter.h>
PN532_SPI pn532spi(SPI, 10);
NfcAdapter nfc = NfcAdapter(pn532spi);
#else
#include <Wire.h>
#include <PN532_I2C.h>
#include <PN532.h>
#include <NfcAdapter.h>
PN532_I2C pn532_i2c(Wire);
NfcAdapter nfc = NfcAdapter(pn532_i2c);
#endif

String UID, ID, MDP, PORT, RFID;
int UIDValid, UIDIndexOf;

void setup()
{
    Serial.begin(9600);
    nfc.begin();
}

void loop()
{
    tagValidity()
}

void tagValidity() //Instruction permettant de savoir si un badge
{
    //RFID présenté est présent dans la base de données
    if (nfc.tagPresent())
    {
        NfcTag Tag = nfc.read();
        UID=Tag.getUidString();
        UID.replace(" ", "");
        Serial.print("ID du tag scanné : "+UID);

        //UIDIndexOf=-1 si l'ID du RFID n'est pas présent dans la base de données
        UIDIndexOf=NomVariableBaseDeDonnées.indexOf(UID);
        if(UIDIndexOf==-1){UIDValid=0}else{UIDValid=1}
        delay(2000);
    }
}

```

Il faut aussi savoir si le code provisoire que l'utilisateur a saisi est encore valide. le code ci-dessous remplit ce rôle en vérifiant si l'année, le mois et le jour qui sont présents dans le code provisoire sont aussi ceux de la date et de l'heure actuelle.

```

//Bibliothèques//
#include <Wire.h>
#include "DS1307.h"
DS1307 clock;

int CodeYear, CodeMonth, CodeDayOfMonth, RealYear, RealMonth, RealDayOfMonth, provisoryCodeState;
String ProvisoryTypedCode;

void setup()
{
  ProvisoryTypedCode="1630010309"; //Affecte une valeur string à ProvisoryTypedCode correspondant au code saisi
  Serial.begin(9600);

  clock.begin(); //Paramétrage de l'horloge
  clock.fillByYMD(2016,1,14);
  clock.fillByHMS(20,01,0);
  clock.fillDayOfWeek(THU);
  clock.setTime();
}

void loop()
{
  provisoryCodeVerif();
}

void getRealTime() //Instruction permettant de récupérer des informations sur la date actuelle
{
  clock.getTime();
  RealYear=(clock.year, DEC);
  RealMonth=(clock.month, DEC);
  RealDayOfMonth=(clock.dayOfMonth, DEC);
}

void provisoryCodeVerif() //Instruction vérifiant la validité du code
{
  getRealTime();
  CodeYear=(ProvisoryTypedCode.substring(0,2),DEC);
  CodeMonth=(ProvisoryTypedCode.substring(6,8),DEC);
  CodeDayOfMonth=(ProvisoryTypedCode.substring(4,6),DEC);

  if(RealYear!=CodeYear || RealMonth!=CodeMonth || RealDayOfMonth!=CodeDayOfMonth)
  {provisoryCodeState=0; Serial.print("Code expiré");}else{provisoryCodeState=1; Serial.print("Code valide");}
}

```

## 6) INTEGRATION ET VALIDATION :

La mise en commun des codes a déjà été effectuée.

La partie concernant le module de la vie scolaire n'a pas été mise en commun étant donné que le boîtier est séparé de la borne d'accès du garage à vélos. Toutefois il fonctionne parfaitement.

Michaël se charge de regrouper les codes. C'est également lui qui se charge principalement de la base de données même si cette tâche est répartie entre nous deux. Si son programme pour la base de données venait à ne pas fonctionner, le mien fonctionne correctement et il ne resterait plus qu'à changer quelques paramètres pour l'adapter dans le code général.

Le code vérifiant la validité du code provisoire fonctionne d'après des tests effectués par mes soins en définissant moi-même le code tapé dans le *setup()* du programme. Il faut maintenant le modifier pour qu'il fonctionne avec la valeur que retourne le clavier numérique du garage à vélos.

## 7) CONCLUSION :

Pour conclure, j'ai trouvé ma partie de projet très intéressante. Les badges RFID étant de plus en plus présents au quotidien il était intéressant de voir comment fonctionne cette technologie. Le lecteur de badges RFID que nous possédons est aussi un lecteur NFC, il permet aussi de se servir d'un téléphone comme d'un badge par exemple. J'ai moi-même fait le test avec le téléphone d'un camarade, équipé de la technologie NFC ce qui nous a donné envie de pousser le projet un peu plus loin comme par exemple une base de données stockée sur un serveur web qui permettrait à la vie scolaire d'ouvrir ou de verrouiller le local depuis un ordinateur via un logiciel que nous pourrions créer ou encore l'identification via un téléphone portable.