

**MASTER RAPID APPLICATION DEVELOPMENT**

**ANTONIN SANDOZ**

**MASTER RAPID APPLICATION DEVELOPMENT  [Le 10 juin 2014]**

**Projet ruby on rails**

**ANTONIN SANDOZ**

**La partie électronique et mécanique de ce projet et réalisé à l’aide de mon associé.  
Ce projet et son idée doit rester confidentiel**

**Cahier des charges**  
Développement d’une application en ruby on rails   
interface utilisateur - photobox arduino

Table des matières

[Introduction 2](#_Toc390178836)

[Architecture du projet 3](#_Toc390178837)

[Principe de fonctionnement 4](#_Toc390178838)

[Application web et API 4](#_Toc390178839)

[Cahier des charges de l’application web 4](#_Toc390178840)

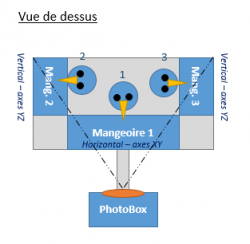
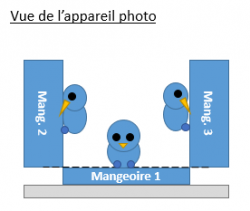
# Introduction

Le but de ce projet est de réaliser une « boîte à photos connectée » permettant de prendre en photos des oiseaux, cette boîte serait une sorte de «bird feeder » *(figure 2)* permettant : d’attirer les oiseaux, de les détecter et de les photographier.

Une fois la photo prise celle-ci est envoyée au serveur à l’aide du protocole de transfert http.

L’utilisateur peut donc de la sort être averti en directe de la visite d’un oiseau.

Le principe est illustré ci-dessous *(figure 1)*



**Fig. 1** Illustration de la « bird box ».

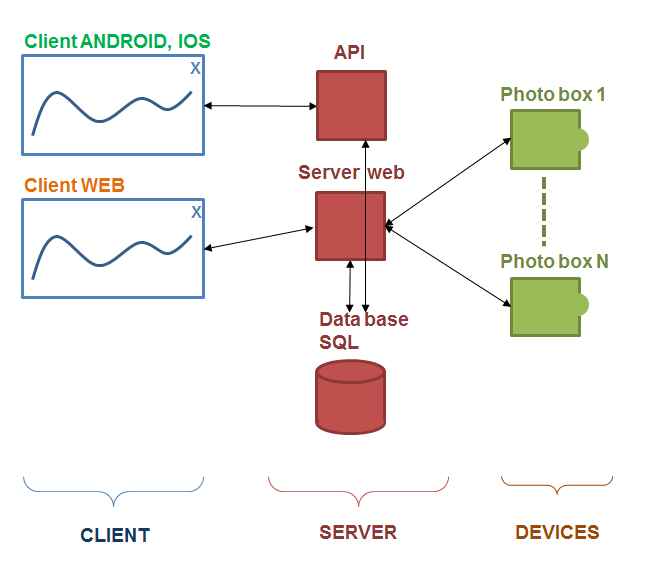


**Fig. 2** Exemple de « bird feeder » , vue de profil et de face.

# Architecture du projet

Le projet est constitué des éléments suivant *(figure 3)*:

* **Boîte à photo**  
  L’ensemble de l’électronique est développé à l’aide d’Arduino (plateforme électronique de développement open source)
* **Server Application et API**
* **Clients**

****

**Fig. 3** Illustration de l’architecture générale du projet.

La partie server a été séparée en deux parties :   
Un serveur web permettant de réceptionner les photos envoyées par la « Photo box » et il permet aussi de gérer l’application web  
Un server API qui permet de fournir les informations aux applications (non-web).

A terme il serait souhaitable de développer des clients mobiles (Android, IOS, WM)

# Principe de fonctionnement

Le principe de fonctionnement est le suivant :

Electronique (Arduino)

* Détection de l’oiseau
* Prise de la photo
* Stockage
* Envoi de la photo POST HTTP

Informatique – application web (Ruby on rails)

* Identification de la « photo box »
* Réception de la photo
* Stockage dans la BDD
* Gestion de l’application web permettant de consulter les photos par les utilisateurs

Informatique – application mobile (IOS, Android)

* Gestion de l’application mobile permettant de consulter les photos par les utilisateurs

Mécanique – conception de la « photo box »

* Inspiré des « birds feeder »
* Etanchéifie l’électronique

# Application web et API

L’application web ainsi que l’API seront réalisées en Ruby é l’aide du Framework web « Ruby On Rails »

**Dans ce projet uniquement l’application web est réalisée.**

# Cahier des charges de l’application web

L’application web est réalisée à l’aide du Framework « Ruby On Rails »

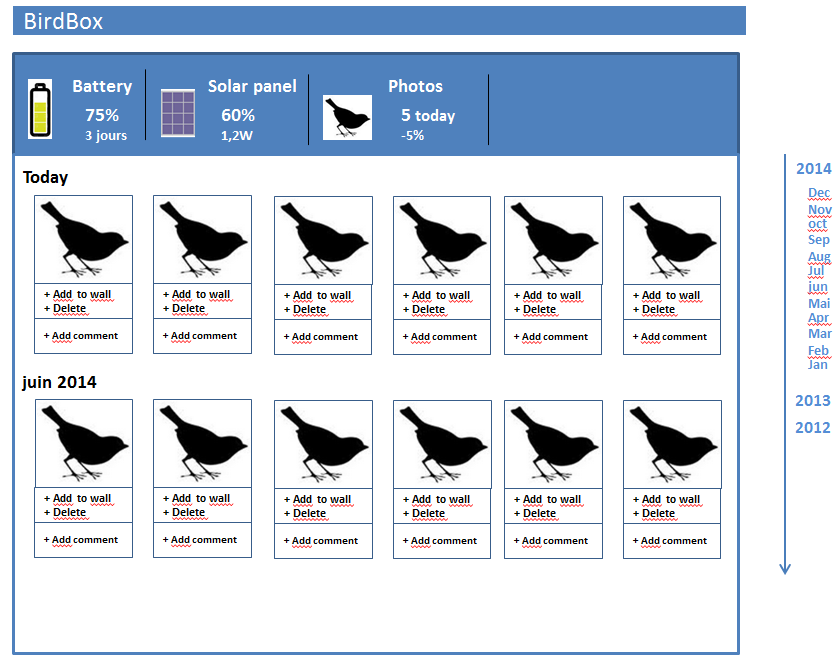
L’application permet aux utilisateurs de s’identifier et de consulter les photos prisent par la « photo box ».

L’application est principalement séparée en deux parties :

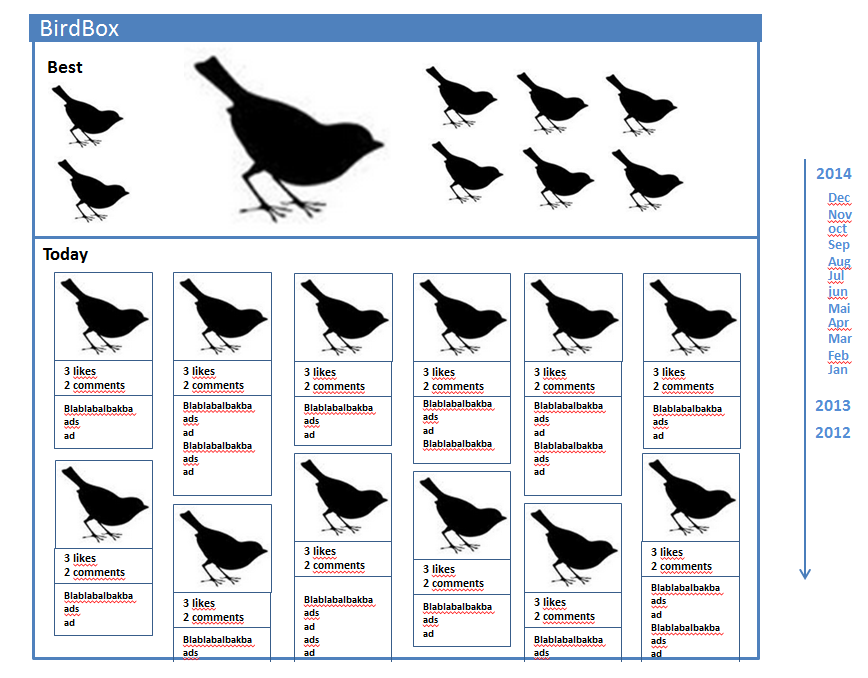
* Partie « Administrateur » (identification requise)
* Partie « Public »

La partie « Administrateur » permet à l’utilisateur de gérer ces photos (basculer sur son « mur » partie « Public » ou effacer.), on retrouve aussi les informations relatives à la « photo box »

La partie « Public » est un album photo ou les photos favorites de l’utilisateur sont exposées dans un ordre chronologique la plus récente en dernier.  
  
Les *figures 4 et 5* représentent le « Layout » des deux principales pages de l’application.



**Fig. 4** « Layout » Page administrateur



**Fig. 5** « Layout » Page publique