HEALTHEYE: THE SOCIAL NETWORK

THE SOCIAL NETWORK FOR RARE DISEASES

Jonathan Hurtado Yrula

JUNIO DE 2015
TRABAJO FIN DE GRADO

Título: HealthEye: The Social Network for rare diseases.

Título (español): HealthEye: La red social de las enfermedades raras.

Autor: Jonathan Hurtado Yrula

Tutor: Joaquín Salvachúa

Departamento: ingeniería de Sistemas Telemáticos.

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente: Santiago Pavón

Vocal: Joaquín Salvachúa

Secretario: Gabriel Huecas

Suplente: Juan Quemada

FECHA DE LECTURA:

CALIFICACIÓN
ACKNOWLEDGMENTS

When I started this project I could never imagine what HealthEye was going to provide me such an amazing experiences that I’ve lived. All the people that I’ve known and all the things that I’ve learnt can’t be written in here, but they will go with me the rest of my life.

I have to thank you to all the people and institutions that believed in me and in the project, such as Joaquín Salvachúa, and the Orange Chair for granted me the StartPFC scholarship to do this project, and to SECOT and all the tutors that I’ve had for this hard path.

To those who really helped me to be a better person and a future engineer that want to innovate such as the professors Vicente Alcober & Pilar Mareca, to my friends that provided me such a wonderful moments during my career, and the last but not least, in fact the most important for me, to my parents that they’ve always been here with me, believing in me, always helping me and teaching me in the best way a parents can do all that.

Thanks to all,

Jonathan
**SUMMARY**

HealthEye is the health social network to connect people with the same rare disease. This project is considered as a social startup, addressing the development of the website, besides the business model of the project. The hardest part in this is, above all the things, to find a sustainable business model for the project, and of course, create a good quality website, to the benefit of the future users.

**RESUMEN**

HealthEye es la red social para unir a personas con la misma enfermedad rara. Este proyecto, es considerado como una startup social, abordando tanto el desarrollo de la plataforma así como el modelo de negocio del proyecto. La parte más complicada del proyecto, sin duda alguna, es encontrar un modelo sostenible para la empresa, y por supuesto, el crear una web de buena calidad para el beneficio del usuario.

**KEY WORDS**

Social network, diseases, Symfony, MVC, Business Model, Social Entrepreneurship, PHP, Yaml.
INDEX

1. INTRODUCTION TO THE PROJECT......................................................... 4
2. INTRODUCTION TO THE DEVELOPMENT ........................................ 6
1. INSIDE THE WEBSITE......................................................................... 7
2. HEALTHYE’S BUNDLES....................................................................... 9
   2.1. USER BUNDLE ............................................................................... 9
      3.1.1 DefaultController.php ................................................................. 10
      3.1.2. SitioController.php.................................................................... 17
      3.1.3. Entity Repository ........................................................................ 18
      3.1.4. RegistrationFormType.php .......................................................... 24
      3.1.5. Routing.yml .............................................................................. 25
      3.1.6. Tests .......................................................................................... 27
      3.2. Recapitulation ............................................................................... 29

I. Executive Summary(EN/ES) .................................................................... 31.

II. Problem/Opportunity ........................................................................... 33

III. Timing: why now? .............................................................................. 34.

IV. Solution ................................................................................................ 35

V. Business Model ..................................................................................... 35

VI. Strategic Analysis ............................................................................... 35.

VII. Marketing Plan ................................................................................... 36.

VIII. Company Ownership ...................................................................... 42

IX. Financial Projections ......................................................................... 44

Conclusions............................................................................................... 49

Bibliography............................................................................................ 50

ANNEX.................................................................................................... 51
CHAPTER 1

1. INTRODUCTION TO THE PROJECT

Imagine that you are living on the 60’s, you are living happily with your whole family, but, in one instant, everything change. You start feeling really weak, you don’t know what’s going on, and then, darkness.

My mother felt that for all her teenage, she doesn’t know anything about what happened to her, nor her parents too. She actually has a chronic disease: epilepsy. But, the key word here is actually, because when she was a little girl, epilepsy was considered as a rare disease.

To know that nowadays, can be many people in the same situation of abandonment, just for the misinformation about their diseases, is awful. And as a future engineer, when my mother told me how she felt, I thought in that people that actually have rare diseases, and for the lack of budget research, haven’t the opportunity to live a normal life.

My main aim with HealthEye (http://healtheye.es) is provide to all that people a platform where no matter the disease you have, you will be able to share your knowledge with others like you and know more about your disease not only for the people in your same situation, but with the big support of specialized doctors in the industry.

This project started thanks to a internship in Telefonica, the Talentum Startup Short Track, where I started to develop the beta website in the Wayra Academy of Madrid. Then, I was selected for the UEIA incubator of social projects, the best academy for social startups in Europe, and last but not least, I was able to get the Orange Professorship Scholarship, StartPFC, where I was able to jump to the real market.

1.1. Project Goals

In the long term, this project wants to be the best social network for rare diseases in Spain and in a big part of Latin America. To do so, I have to do different tasks such as:

- Development of the platform where every user can know others like them. The philosophy will be in MVC.
HEALTHEYE: THE SOCIAL NETWORK FOR RARE DISEASES

✓ Have the support of big entities in the rare diseases environment, such as charities and many other organizations.

✓ Design a great business model, to be able to present the project to social investors.

✓ Find the support of many different specialized doctors for the platform.
CHAPTER 2

2. INTRODUCTION TO THE DEVELOPMENT

We are going to explain the process of the development of the website, through the explanation of the framework, and examples of how I’ve created the website. The framework is Symfony2, that it was created to develop web apps through PHP. In this occasion, I’ve used the latest version of PHP, PHP 5.3.

To the frontend part, we have to use HTML5, with the extension of .html.twig for the files, and .css for the design. Besides, I’ve used Javascript to develop some special features of the website.

To the backend part, the main thing we have to know is that everything works around the Entities. These entities, are the objects that works as the link with the database, that will be Doctrine. The PHP code that we use in Symfony, is not working with SQL sentences, but with objects through PHP. Apart of the entities, we have the body of Symfony, the Bundles.

A bundle is a logic division of the application. Each bundle has their own PHP classes and web files, including the security and routing files that we have to include to the application.

So, I am going to start explaining the distribution of the website, the bundles and highlighting special features and coding matters of the project.

After this, I’m going to talk about the business plan of HealthEye. It’s very important to notice that this has been a real project, with real users actually and today, we have more than 350 users in the platform.

Besides, we’ve got the support of different entities such as the Minister of Health and Social Affairs in Spain, big tech companies like Telefonica, Orange or Microsoft, and above all the things, we’ve helped so many people that this project is not an usual one, but it’s a real one with proofs of success.
HEALTHEYE: THE SOCIAL NETWORK FOR RARE DISEASES

1. INSIDE THE WEBSITE

The heart of HealthEye is sharing everything you know about your disease. I first thought HealthEye as the “Facebook of rare diseases”, that is why it exists “My Profile” where you can share your day-to-day with the people inside HealthEye that have your same disease. Then, we have “My Story” where you can tell your own story about your disease.

Above all this, so particular of the whole social networks, we have what differentiate us against our competence: the creation of your “healtheyes”. A “healtheye” is a resource, in which you can share all you want with people with your same rare disease. For example, if I know an excellent specialized doctor, I can create a “healtheye” with this doctor. Not just doctors, but charities, medicaments, etc.

In “My Disease” you can see all the entire “healtheyes” created by people with your same disease.

And last, but not least, we have “My community”, where you can talk with people with your disease inside HealthEye.

We can see different screenshots of the website down here:

![Screenshot of “My Profile” where you can share your day-to-day](image)
**Figure 2**: Screenshot of “My Community” where you can see all the users with your same rare disease.
3. **HEALTHEYE’S BUNDLES**

First of all, I’m going to explain how it’s implemented the website, to the good-understanding of the app.

I’ve created 6 bundles: **UserBundle, CiudadBundle, EnfermedadBundle, HealthEyesBundle, HistoriaBundle and ProfileBundle**. Each bundle, as I’ve said before, it’s a logic division, so as the website is thinking is in that way: *the user is connected with other users of the platform just in case their disease is the same*. That’s why we need to create another special Bundle to connect users with diseases, so **EnfermedadBundle** is done for that reason.

The same thing happens with **CiudadBundle**, and the others, as the distribution of the website is based on the creation of “healtheyes”, your own story and your day-2-day posts, we need to create a bundle for each matter.

We are going to start explaining the **UserBundle**, the most important, to see how it works:

3.1. **USER BUNDLE**

The structure of each bundle is composed by:

1. **Controller**: In this part you give the functionality to the app. For example, in the **UserBundle**, we have to create functions to create a post, edit it, or delete it, and that connect it with **ProfileBundle**, the one who manage the posts.
2. **Entity**: We’ve talked about the entities before. It’s the link for the database, but not with SQL sentences, but PHP objects. For example, each user has his own disease, so we have to define the object: disease and the methods getDisease, setDisease, shown below this explanation.
3. **Form**: In case we have tables, we need to define it in that place.
4. **Resources**: In here we define the security, the routings of the website, through `.yml` files, and in the `config` folder. And the other important thing is the folder view, in which we create the `html.twig` files to give the frontend to the app.

5. **Tests**: We create the tests in this folder.

The way we create a Bundle is with the instruction in the console:

```
$ php app/console generate:bundle
```

I’m going to explain in a detailed way how I’ve proceed with the `UserBundle`, and the others will be in the same way, but not so detailed in this project.

So, after create the bundle, we have to define it in the file `AppKernel.php` in this way:

```
new HealthEye\UserBundle\UserBundle()
```

The first thing we have to do after this is go to the file `app/config/routing.yml` and add

```
user:
  resource: "@UserBundle/Resources/config/routing.yml"
  prefix: /
```

![Figure 3: Reference to the routing.yml of the UserBundle where all the url's are found.](image)

Now, and after this common things with all the bundles, we have to start coding the backend project, starting with the functions that each user will have. So we create in `src/HealthEye/UserBundle/Controller/` the file `DefaultController.php`.

### 3.1.1 DefaultController.php

The most important thing in here is that you have to write all the “links” that we are going to use creating the functions. For example, if we want to create a function in which we use the option to create a post, we need to write at the beginning of the `DefaultController.php` file this:

```
use HealthEye\ProfileBundle\Entity\Profile;
use HealthEye\ProfileBundle\Form\ProfileType;
```

Declaring in this way that we are going to need the Profile Entity and his form to create a post. It works like that with all the rest.

I’m going to explain how I’ve create the function to show a post, to create it, and to edit it with the option to delete it. We have to create three different functions, one for each action:
1) **PortadaPostAction()**

As we can see, we have to call the database in the first sentence, through the creation of an object in Doctrine, the database we use. After this action, we have to create the user, by the second sentence. As we’ll see later, we’ve created the method `getUser()` in `User.php`, so in that way, we can have access to the user logging in.

After that, we have the most important thing in this function, because we are going to call the function `findRecientes($enfermedad_id)`, that we have to create in the file `ProfileRepository.php`, created in the ProfileBundle (Entity folder).

We are going to dig in the `findRecientes($enfermedad_id)` function. As we can see, we must create an object `$em` in here, and we can name it whatever we want to. Then we call the `getEntityManager()`, the direct link that granted us the way to call our database. After this, we have to select from our database the selected post, from the logged user in the platform.
This next point is important in our case, the sentence *WHEN p.revisada= true AND p.enfermedad = :id* is written because not all the post created by each user will appear instantaneously, because I want to avoid the sharing of drugs, or things like that without consulting with specialized doctors. It will appear, when I modify the *revisada* table, and put a ‘1’ instead a ‘0’ by defect.

The next sentences are in this order, to filter the maximum post in three per page, then we have to define the *:id* object that we’ve called in the Doctrine sentences and finally, we have to return the post that we’ve found depending just of the disease.

So, with this function we are able to show all the posts created by every user with a selected disease, just the disease that the user has.

Returning to the function *portadapostAction()* we’ve explained all the process to show the posts depending exclusively of the disease of the user, not anymore.

So, finally, we have to return the posts, and the html.twig depending from it is: *portadapost.html.twig*

We show in the next page the creation of the portadapost file for a better understanding of the html.twig utility.

![Figure 6: Extract of the portadapost.html.twig file.](image)

We can see up to this, the html coding, but we have sentences in which we can call the objects we require in the functions, such as the posts created with the sentence *{% for post in posts %}*
We have to remember that in function we return ‘posts’ so it is well created. We can see a proof of this, in this image, a user logged as ‘Jonathan’ with a disease, can see another post created for ‘Pilar’, with his same disease in the same way we’ve coding in the html.twig

2) We are going to start with postNuevaAction():

```php
/**
 * Muestra el formulario para crear una nueva historia y se encarga del
 * procesamiento de la información recibida y la creación de las nuevas
 * entidades de tipo Profile.
 */
public function postNuevaAction()
{
    $petición = $this->getRequest();
    $post = new Profile();
    $formulario = $this->createForm(new ProfileType(), $post);
    $formulario->handleRequest($petición);

    if ($formulario->isValid()) {
        // Complete los propiciadugles del post que un usuario no puede establecer
        $user = $this->get('security.context')->getToken()->getUser();
        $post->setRevisada(false);
        $post->setUser($user);
        $em = $this->getDoctrine()->getManager();
        $em->persist($post);
        $em->flush();
        return $this->redirect($this->generateUrl('post_portada'));
    }
    return $this->render('UserBundle:Default:formulariopost.html.twig', array(
        'accion' => 'crear',
        'post' => $post,
        'formulario' => $formulario->createView()
    ));
}
```

*Figure 7: Function postNuevaAction() to create a new post*
The first line in the function, the `getRequest()` method is for the interaction client-server with the HTTP server; to do so, we create the object `$peticion`. Then we have to create a new post, that we call it through the creation of a new Profile (post).

The heart of this function is the creation of a form. We do so, by the sentence `createForm(new ProfileType(), $post)`, calling the object `$post` that we have created before. Now, we have to call again to the HTTP request by calling the `$peticion` object.

Now, after all these sentences, we have to create the form indeed. But there are properties that the user himself can’t manipulate, but we do. These properties are the user himself logged in and the option that allows us that the post doesn’t appear instantaneously.

Then we have to call to our database, creating us an object `$em`. After this action, we have to connect the objects that we’ve created in our database (our Entity `User.php` has all the information of these ones, shown below) and we do it with the `persist` action. This method let Doctrine to manage all the data related with the posts. With the next line, we call the `flush()` method, that allows Doctrine to examine all the objects that is managing to see if it’s necessary persist them in the database. Then, it’s easy to return a generated URL, through the `redirect` method, the redirect () method is simply a shortcut that creates a special Response object that redirects users.

We redirect to `post_portada` URL, that we’ve created in the UserBundle, specifically, in `Resources/config/routing.yml`: (explain in paragraph 3.1.5 Routing.yml in a better way)

```
use Symfony\Component\HttpFoundation\RedirectResponse;
return new RedirectResponse($this->generateUrl('post_portada'));
```

*Figure 8: Redirect Response to the post_portada URL found in the routing.yml file*

We have to finish the explanation of the `postNuevaAction()` function, so after closing the if condition, the `renderview()` method takes a template and returns its rendered content, but we can call it by the `render(‘’, array)` method, creating the same result. So, in our case we render the `formularioestpost.html.twig` file, creating the form. We show the file in here, located in `UserBundle/Resources/views/Default`.

As we can see, we have to call the ‘post’ object, created in the returning action of the render method in the function `postNuevaAction()`, to create a block and then, we have to call to the ‘accion=’crear’, to create in the database a new row of the post content. After all this important things, we have the common ones for example the creation of the form by `{{ form_label(formulario.post) }}`.

`postNuevaAction()`.
Now, and to finish these explanations to create the most important functions for the
success of HealthEye, we have to explain how we’ve coded the action to edit something
created by the own user, so we are going to explain the `postEditarAction()` function:

3) **postEditarAction()**:

```php
function postEditarAction($id) {
    $petición = $this->getRequest();
    $em = $this->getDoctrine()->getManager();

    $post = $em->getRepository('ProfileBundle:Profile')->findOne($id);

    if (!$post) {
        throw $this->createNotFoundException('el post indicado no está disponible');
    }

    $formulario = $this->createForm(new ProfileType(), $post);
    $formulario->handleRequest($petición);

    if ($formulario->isValid()) {
        $em->persist($post);
        $em->flush();

        return $this->redirect($this->generateUrl('post_portada'));
    }

    return $this->render('UserBundle:Default:formulariopost.html.twig', array(
        'accion' => 'editar',
        'post' => $post,
        'formulario' => $formulario->createView()
    ));
}
```

**Figure 9: Extract of the file formulariopost.html.twig**

**Figure 10: PostEditar function**

```php
public function postEditarAction($id) {
    $petición = $this->getRequest();
    $em = $this->getDoctrine()->getManager();

    $post = $em->getRepository('ProfileBundle:Profile')->findOne($id);

    if (!$post) {
        throw $this->createNotFoundException('el post indicado no está disponible');
    }

    $formulario = $this->createForm(new ProfileType(), $post);
    $formulario->handleRequest($petición);

    if ($formulario->isValid()) {
        $em->persist($post);
        $em->flush();

        return $this->redirect($this->generateUrl('post_portada'));
    }

    return $this->render('UserBundle:Default:formulariopost.html.twig', array(
        'accion' => 'editar',
        'post' => $post,
        'formulario' => $formulario->createView()
    ));
```
As we’ve done before, we are going to explain line-to-line. The first one, and the second one, we’ve explained with the functions `postNuevaAction()` and `portadaPostAction()`, respectively. But this one it’s the most important in this function:

```php
$post = $em->getRepository('ProfileBundle:Profile')->find($id);
```

With this action, we call to our Database in Doctrine, and we look for the post with the concrete `$id`, depending on the user, and of course, of the `$post`. After that, we must to create an exception, in case the found post is not created.

Then, we have to re-create again a form element, to edit the specific post, we do so by the method `createform(new Class, object)`. So, the next step will be to call the specific post in our database and see if the form is submitted, we do that through the method `handleRequest()`.

Finally, to highlight, in the above method, we returned through the render method the form we wanted to create. It’s important to see that we’ve written ‘`accion => ‘editar’`’, to distinguish from the ‘`accion => ‘crear’`’ of the `postNuevaAction()` function.

![Figure 11: Extract of the formulriopost.html.twig highlighting this special part](image)

We do so, just because it will be necessary for the `formulariopost.html.twig`. If we go there, we can see that:

We see that we have two cases: the one for creating the form, and the other to edit it, going to the routes show in the Figure 27.

Finally, in the `portadapost.html.twig` file, we can see a part of the code, very important indeed:

![Figure 12: Extract of the portadapost.html.twig](image)

We see that for the body class, we have to put the condition to find out if the post belongs to that concrete logged in user, or another one. If so, then you will be able to edit it, by calling again the route `post_post editar`. 
3.1.2. SitioController.php

To finish the Controller part of the UserBundle, we are going to talk about the SitioController.php file. In this file, we call to the links that we have in our footer:

We just have to redirect to our routing.yml file and with just a simple line of code for example, the About Us (Sobre Nosotros t he image):

```php
public function aboutAction()
{
    return $this->render('UserBundle:sitio:about.html.twig');
}
```

Very simple, just we render to our about.html.twig and we’ve done everything.

However, the next function is not so easy as the shown above. We are going to explain about the contactoAction() function, in which everyone can contact us through just a simple form:

```php
public function contactoAction()
{
    $petition = $this->getRequest();
    $formulario = $petition->createFormBuilder()->add('nombre', 'text')
        ->add('email', 'email')->getForm();
    if ($petition->isSubmitted() && $petition->isValid()) {
        $formData = $petition->getData();
        $sucursal = $this->container->get('sucursal_manager');
        $sucursal->create($formData);
        $this->addFlash(
            'success',
            'Your contact has been successfully created.'
        );
    }
    return $this->render('UserBundle:sitio:contacto.html.twig', array(10));
}
```

Figure 15: Function contactoAction()
First of all, as we’ve explained in above examples, we have to create the specific form, just for the email area, and the body one. We have to request a petition to the HTTP Request, based on the POST form, to generate the contact between the website and us. The most important thing in this function is the contact with the server through the instruction `get(HTTP_USER_AGENT)` and the `(REMOTE_ADDR)`, which we are able to connect with our server. Then, the message we send it to jonathanyrula@gmail.com as we can see in setTo(’email’). Then we have to return the `contacto.html.twig` file, shown below.

![Tu dirección de email](image)

**Figure 16: Extract of the contactohtml.twig**

3.1.3. Entity Repository

The Symfony2 applications do not handle your information by directly accessing the database. They create, modify and delete information using PHP objects instead of creating and executing statements SQL. This is possible thanks to external libraries or call ORM Object-Relational Mapping, one of those is Doctrine.
The entity repository is the direct link with our database in Doctrine. Each entity is considered as a table in our database, so we need to create 6 entities, Ciudad.php, Enfermedad.php, Profile.php, Historia.php, HealthEyes.php and User.php.

As we focus just in the UserBundle to see the general action, in here we have two files, the User.php and the UserRepository.php, that we’ve talked about explaining the method findRecientes(), Figure 5.

1. User.php

First of all, we have to create the file, and then the first lines must be these ones:

```php
<?php
namespace HealthEye\Entity;
use FOS\UserBundle\Model\User as BaseUser;
use Doctrine\ORM\Mapping as ORM;
use Doctrine\Common\collections\ ArrayCollection;
use Symfony\Component\Validator\Constraints as Assert;
use Symfony\Component\Security\Core\User\UserInterface;

class User extends BaseUser implements UserInterface
{

/**
 * Método requerido por la interfaz UserInterface
 */
public function eraseCredentials()
{
}

/**
 * Método requerido por la interfaz UserInterface
 */
public function getRoles()
{
    return array('ROLE_USER');
}

/**
 * Método requerido por la interfaz UserInterface
 */
public function getUsername()
{
    return $this->getEmail();
}

/**
 * @ORM\Id
 * @ORM\GeneratedValue(strategy="AUTO")
 */
protected $id;
```

Figure 17: Extract of the UserRepository header to call to these methods.

To write the annotations in Doctrine, we have to do it with the prefix @ORM, so we must import the Mapping in ORM. The rest is for the correct function of the entity.

```php
/**
 * @ORM\Entity(repositoryClass="HealthEye\Bundle\Entity\UserRepository")
 * @ORM\Table(name="users")
 */

class User extends BaseUser implements UserInterface
{

/**
 * Método requerido por la interfaz UserInterface
 */
public function eraseCredentials()
{
}

/**
 * Método requerido por la interfaz UserInterface
 */
public function getRoles()
{
    return array('ROLE_USER');
}

/**
 * Método requerido por la interfaz UserInterface
 */
public function getUsername()
{
    return $this->getEmail();
}

/**
 * @ORM\Id
 * @ORM\GeneratedValue(strategy="AUTO")
 */
protected $id;
```

Figure 18: Extract of the User.php in the Entity folder.
The first thing we have to do in every Entity file is calling the @ORM notation, to be able to create the table with simple instructions in the terminal. We do so, with:

```php
@ORM\Entity(repositoryClass="HealthEye\UserBundle\Entity\UserRepository")
@ORM\Table(name="users")
```

With this, we call the table we want to create “users” and we refer it to the functions created in `UserRepository.php` (shown below). After this, we require some functions to define the User such as his role, and his username, that in our case, is the email.

We go deep into the Entity, so we need to create the object we need to create to define the user. In our case is the identity ($id), name ($nombre), surname ($apellidos), disease ($enfermedad), and city ($ciudad). We see in here that there is a dependence with CiudadBundle, and EnfermedadBundle. So we define the objects on this way:

```php
/*
 * @ORM\Column(type="string", length:255)
 * @Assert\NotBlank(groups=["Registration", "Profile"])
 */
protected $apellidos;

/*
 * @ORM\ManyToMany(targetEntity="HealthEye\EnfermedadBundle\Entity\Enfermedad")
 * @Assert\NotBlank(message="Please enter your name.", groups=["Registration", "Profile"])
 */
protected $enfermedad;
```

Figure 19: Detail of two objects in the User.php Entity file.

For example, we have two properties: $apellidos (surname) and $enfermedad (disease). We must put this kind of data to be able to register in HealthEye, so we put the notation `@Assert/NotBlanck`.

Besides, the properties of Doctrine2 entities can only be protected or private, never public. This avoids problems when accessing a property of an object related not yet loaded.

Another interest thing is that if the object has dependence with other special Bundle, for example in the case of $enfermedad, we have to call the specific entity to know that we’ve created a property that must have the properties of `Enfermedad.php`, that is why we must write before the creation of $enfermedad:

```php
@ORM\ManyToOne(targetEntity="HealthEye\EnfermedadBundle\Entity\Enfermedad")
```

The many-to-one association is always the owning side of any bidirectional association, so that is why we put the ManyToOne; then we call the `Enfermedad.php` created in EnfermedadBundle/Entity.
Then, we must create the getters and setters of each property, but this is one of the easiest part of the project. Just by writing in the terminal:

```
$ php app/console generate:doctrine:entities UserBundle
```

So, all the getters and setters of each property will appear with the dependence of his specific entity. For example, in the image below we have the example of the $apellidos getter, and the $enfermedad setter. As we can see, getApellidos() is a simple getter that return the property, but in the method setApellidos(), we have to call the Enfermedad entity as we’ve explained before.

```
/**
 * @ORM\Id
 * @ORM\Column(type="string", length=100)
 * @ORM\GeneratedValue(strategy="AUTO")
 */
protected $id;

/**
 * @ORM\Column(type="string", length=100)
 */
protected $nombre;

/**
 * @ORM\Column(type="string", length=100)
 */
protected $slug;
```

Figure 20: Extract of the Enfermedad.php Entity file

Now, that we’ve finished the explanation of the User.php file, we have to talk a little bit about the heart of the application: the `UserRepository.php` file.
2) UserRepository.php

In here we have to link our application, with the database. To see some examples, in fact we’ve talked before at Figure 5, about the `findRecientes()` function, we are going to explain 2 more examples to clarify the function of the Repository: the `findCercanas()` function and `findUltimosPostsPublicados( $id, int, $disease)` one.

![Figure 22: findCercanas() Function in UserRepository.php](image)

We can see that is necessary pass two parameters: $user and $enfermedad. We have to find the similar users to the one logged in depending exclusively on his disease. So, we have to start creating the link with the Database.

Then, we have to create the $consulta object, just to find out if there is any user with your same disease. After this, with the generator of consults of Doctrine we have to call the QueryBuilder object that made possible to find between all the parameters you want to look for. Then, we call, “u” to the user, and “e” to the disease.

The next instruction is the FROM one, that looks for in the entity User the user “u” joining it with the disease “e” in the Enfermedad entity.

After finding out what users with a concrete disease are similar to the logged user, we have to define the maximum results shown in the website, specifically in “My community” and we’ve defined a maximum of 60 users with your same disease, and finally we just have to return the $consulta object.
Now, it is the turn of the `findUltimosPostsPublicados` function:

```php
public function findUltimosPostsPublicados($user_id, $limite = 5, $enfermedad) {
    $em = $this->getEntityManager();
    $consulta = $em->createQuery('SELECT p, u, e
                                    FROM ProfileBundle:Profile p JOIN p.user u JOIN u.enfermedad e
                                    WHERE p.revisada = true AND p.user = :id AND e.slug = :enfermedad
                                   ');
    $consulta->setMaxResults($limite);
    $consulta->setParameter('id', $user_id);
    $consulta->setParameter('enfermedad', $enfermedad);
    return $consulta->getResult();
}
```

**Figure 23: findUltimosPostsPublicados() in UserRepository.php**

This function looks for all the posts written by a specific user. It’s important to notice in here what we mentioned before, the importance of the review content sent by every single user to our database. As we started in the last function, we have to call the `getEntityManager()` method, and we create us the `$consulta` object with this logic:

1) We select the profile ‘p’ from the user ‘u’ with the disease ‘e’
2) We have to do that by looking for in the Profile entity joining them, because it’s like an if and only if condition.
3) The post then will appear just if the revisada condition it’s true, I mean, I have to do it myself, to see if what the user has shared is truly trustable or not, so it’s very important this part.
4) We finish selecting the results that will be shown with a limit of 5 posts per page.

As a brief recapitulation, we’ve seen how to create a bundle with one of the most important things in the project, the `DefaultController.php` with all the functions that you want to create by the explanation of how to create an html.twig file calling the different actions you prefer to be shown.

Besides, we’ve seen the heart of each Bundle, with the Entity and the Repository files, the direct link with the database.

But, we have to explain a very important few things still, the creation of the formularies, and the tests.
After this, I’m going to explain a little things, highlighting functions in other bundles, to clarify even more the creation of each part of the project.

### 3.1.4. RegistrationFormType.php

```php
namespace HealthEye\UserBundle\Form\Type;

use Symfony\Component\Form\FormBuilderInterface;
use FOS\UserBundle\Form\Type\RegistrationFormType as BaseType;

class RegistrationFormType extends BaseType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('nombre')
            ->add('apellidos')
            ->add('enfermedad', 'entity', array(
                'class' => 'HealthEye\enfermedad\Entity\Enfermedad',
                'empty_value' => 'Selecciona la enfermedad que padece',
            ))
            ->add('ciudad', 'entity', array(
                'class' => 'HealthEye\Ciudad\Entity\Ciudad',
                'empty_value' => 'Selecciona una ciudad',
            ));

        parent::buildForm($builder, $options);
    }

    public function getName()
    {
        return 'healtheye_user_registration';
    }
}
```

*Figure 24: RegistrationFormType.php file*

When we want to create a formulary, that is a place where the user writes a string that will appear in our database, we have to create a .php file with this structure. We have to call to the `FormBuilderInterface` and create an array to fill in with the object we want to create as formularies.

So, we must start creating the formularies, such as ‘nombre’ (user’s name), ‘apellidos’ (user’s surname) and important things such as ‘enfermedad’ (user’s disease) or his city with the object ‘ciudad’.

In these two last cases, we can see that we have to call to the entity of the object we have created. For example with ‘enfermedad’ we should call the ‘Enfermedad Entity’ and write what the user is going to see at the registration time:
As we can see in the figures shown in this page, the left one shows how a drop-down list is created with all the elements of the ‘enfermedad’ table of our database. In the right one we can see the usual aspect of the registration part in the website.

### 3.1.5. Routing.yml

When you create a bundle in Symfony, there will be different pages (parts of the url) to be considered. For example, in our UserBundle, we have this part of the **routing.yml** file, concretely the part of the posts to keep on going with our first example of the project:

```
post_portada:
    pattern: /posts
    defaults: { _controller: UserBundle:Default:portadaPost }

# Ruta para crear un nuevo post
post_post_nueva:
    pattern: /post/nuevo
    defaults: { _controller: UserBundle:Default:postNueva }

# Ruta para modificar los datos de un post
post_post_editor:
    pattern: /post/editor/{id}
    defaults: { _controller: UserBundle:Default:postEditor }
```

*Figure 27: Reference to the routing.yml of the UserBundle where all the url's are found.*
We’ve explained before that we made the functions to create, edit and delete a post. That’s why we must create an URL for each one of the actions. We can see in the figure above, that we simply gave the title page in the profile as /posts.

When we create a new post, we have to call to the DefaultController.php file, concretely to the postNueva() function. It’s the way the routing file works.

Important to notice that when an user want to edit a post, in addition to call the postEditar() function, we have to create a random id to know in the database which one belongs.

We can see an example in here:

![Add Post Button](image1)

![Screenshot of a possible post created by an user](image2)

Figure 28: Screenshot of a possible post created by an user

And when we click to Edit (Modificar at the image), we can see this (with a special highlight in the url):

![Edit Post Action](image3)

Figure 29: Screenshot of the edit post action in the website. Look at the url too.
3.1.6. Tests
One of the most important things for any development project is the tests. In Symfony, we can create two different kinds of tests: unitary (for example to validate the entities you’ve created) and functional (for example to validate the register action for the users).

We are going to explain two examples of each kind of tests, starting for the unitary one to validate in the `HealthEyes.php` Entity (HealthEyesBundle) the correct assignment of the disease in the healtheye that every user can create.

1) Unitary Test:

Then, we should call this test, just as simple as write in our terminal `phpunit --c app`, executing instantaneously all the recently created test.

Let’s going to explain a little bit the `testValidarEnfermedad()` function. We’ve created a HealthEye object called `$enfermedad`, we’ve provided it a name and a commentary to be shown in the terminal.

Then, we’ve created a test disease, in a private function (`getEnfermedad()`) that we call in the `testValidarEnfermedad()`.

After this, we can see in the terminal the next message in terminal:

So, we’ve proved that the disease is correctly assigned to the healtheye created by every user.
2) **Functional Test:**

We are going to test in here if the user registration action is working as it should. Here a screenshot to explain better what we’ve done:

![Figure 31: Extract of the Test.php in the UserBundle for the registration action.](image)

The first thing we have to do is creating a new client by calling the `createClient` action in PHP unit.

The main thing we do in here is see above all the string the user has to write (name, surname, email and password), the listing elements such as city and disease must be tested to see if they are assigned to the user.

To do so, we create a form with two dropdowns, one for disease, one for city, and we test if the operation is correctly done.

When we put in terminal `$phpunit -c app`, we see an error, because we haven’t create a tester user, in here we can see what we’ve written to do it:
Finally, with this data, we can see in the terminal an OK, so the registration is done successfully.

3.2. Recapitulation

We’ve been able to see how create each option in a Bundle. As we’ve said in the introduction, we’ve created different bundles for each option we want to provide to the user in our website. As a summary way, I’m going to explain briefly how to create another bundle, for example the HealthEyeBundle to see the main concepts of the project:

1) Creation of the bundle in the terminal. As simple as writing `$php app/console generate:bundle`

2) Define what we want to do in this bundle. Concretely in our HealthEyeBundle, we want to create a HealthEye, a formulary for the user to provide data. So, we should create 2 functions in our DefaultController.php: portadaAction ($enfermedad) that shows the healtheye created, and healtheyeAction($enfermedad, $slug), that shows to every user the detailed of the healtheye.

3) What do we want in our formulary? We can see a screenshot of what we want in the next page:
So, in our entity we need 9 different objects for our table in the database: the id, the title of the healtheye, the address provided for the user, the city, the comment, the review action (just for me to verify the comment), the disease and the user.

4) Now, in our Repository we should call to the action with the database, linking the healtheyes with it.

5) We can’t forget the most important thing for this, we have to create the form in the Form folder linking too with the city and disease entities to show the dropdown.

6) Finally, we have to create our html.twig and .css files to make a “pretty” place for the user and don’t forget the tests.

In here, we’ve seen how I’ve created HealthEye with the Symfony framework, digging into the creation of one Bundle, we’ve been able to make us an idea of how starting creating a project like this one.

But now, as I said before, it’s the turn of make it real, not just a project, but a “touchable” thing. That is why in the next pages, I’m going to show you the business plan for a social startup like this.

Besides, I provide you the link of the website to see the result of the development as well as figure out all the different thing that I’ve been using: http://healtheye.es

With no more hesitation, we start with the Business Model of HealthEye:
I. Executive Summary (EN)

1. Mission

HealthEye provides an innovative way to connect people with the same rare diseases. The form is a website platform, for users who seek to stay connected with people that have their same rare disease, and discover all the actual information about their disease, in an easy, fast and more highly personal way who value a product that can be seemingly integrated with their lifestyle on the move.

2. Product

HealthEye is a social network which allows users to share their feelings and day to day through everything they know about their disease.

3. Market

Segments. Patients with rare diseases, bloggers, doctors, charities and pharmas as B2B partners.

Size. Spain: social networks +11% in 2012. average users/social network: 11.6M and people or families living with a rare disease: 6.5% (3M), assuming a 1.7% reach we will get +50k users.

4. Competitors

<table>
<thead>
<tr>
<th>Competitor</th>
<th>Country</th>
<th>Launch</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatientsLikeMe</td>
<td>USA</td>
<td>2005</td>
<td>25000</td>
</tr>
<tr>
<td>MyHealthTeams</td>
<td>USA</td>
<td>2011</td>
<td>55000</td>
</tr>
<tr>
<td>Personasque</td>
<td>Spain</td>
<td>2012</td>
<td>5000</td>
</tr>
</tbody>
</table>
5. User Acquisition

We have two kinds of user, the active: play a big role in the platform, sharing information and being 10 times per month. The register one only interacts 2-4 times per month.

<table>
<thead>
<tr>
<th></th>
<th>Dec 2015</th>
<th>Dec 2016</th>
<th>Dec 2017</th>
<th>Dec 2018</th>
<th>Dec 2019</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spanish users</strong></td>
<td>5000</td>
<td>12000</td>
<td>20000</td>
<td>35000</td>
<td>50000</td>
</tr>
<tr>
<td><strong>South American users</strong></td>
<td>15000</td>
<td>36000</td>
<td>75000</td>
<td>105000</td>
<td>150000</td>
</tr>
</tbody>
</table>

6. Financial forecasts

<table>
<thead>
<tr>
<th>Financials (€)</th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Revenues</strong></td>
<td>130.750 €</td>
<td>217.091 €</td>
<td>373.226 €</td>
</tr>
<tr>
<td><strong>Expenditure</strong></td>
<td>55.150 €</td>
<td>60.499 €</td>
<td>83.501 €</td>
</tr>
<tr>
<td><strong>Net income</strong></td>
<td>75.600 €</td>
<td>156.592 €</td>
<td>289.725 €</td>
</tr>
</tbody>
</table>

7. Team

- Jonathan Hurtado Yrula, CEO

Telecommunications Engineer. Experience as a developer and project manager in well-known companies like Telefónica Digital, as well as in startups like Invoost or cuaQea. jonathanyrula@healtheye.es.

Resumen Ejecutivo (ES)

1. Misión

HealthEye ofrece una innovadora manera de conectar a personas con la misma enfermedad rara, en forma de plataforma web, para usuarios que buscan estar conectados con otras personas que tienen su misma enfermedad rara y descubrir toda la información actual sobre su enfermedad, de una manera, fácil, rápida y más personal en la que tenemos un producto valorado para el estilo de vida de las personas.

2. Producto

HealthEye es una red social que permite a los usuarios compartir sus sentimientos y su día a día, además de todo lo que conocen sobre su enfermedad.

3. Mercado

Segmentos. Pacientes con enfermedades raras, bloggers, doctores, fundaciones y farmaceúticas actúan como B2B.
HEALTHEYE: THE SOCIAL NETWORK FOR RARE DISEASES

Tamaño. España +11% crecimiento en redes sociales en 2012. Cantidad de usuarios/red social: 11.6M y personas o familias viviendo con una enfermedad rara 6.5% (3M), asumiendo un 1.7%, alcanzamos más de 50.000 usuarios.

4. Competidores

<table>
<thead>
<tr>
<th>Competidor</th>
<th>País</th>
<th>Lanzamiento</th>
<th>Usuarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatientsLikeMe</td>
<td>USA</td>
<td>2005</td>
<td>25.000</td>
</tr>
<tr>
<td>MyHealthTeams</td>
<td>USA</td>
<td>2011</td>
<td>55.000</td>
</tr>
<tr>
<td>Personasqué</td>
<td>España</td>
<td>2012</td>
<td>5.000</td>
</tr>
</tbody>
</table>

5. Adquisición de usuarios

Tenemos dos tipos de usuarios, los activos: juegan un gran papel en la plataforma, compartiendo información y entrando de media 10 veces al mes. El otro tipo son los "curiosos", quieren información sobre alguna enfermedad, e interactúan una media de 2-4 veces por mes.

<table>
<thead>
<tr>
<th></th>
<th>Dec 2015</th>
<th>Dec 2016</th>
<th>Dec 2017</th>
<th>Dec 2018</th>
<th>Dec 2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usuarios España</td>
<td>5.000</td>
<td>12.000</td>
<td>20.000</td>
<td>35.000</td>
<td>50.000</td>
</tr>
<tr>
<td>Usuarios LatAm.</td>
<td>15.000</td>
<td>36.000</td>
<td>75.000</td>
<td>105.000</td>
<td>150.000</td>
</tr>
</tbody>
</table>

6. Previsiones Financieras

<table>
<thead>
<tr>
<th>Financials (€)</th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingresos</td>
<td>130.750 €</td>
<td>217.091 €</td>
<td>373.226 €</td>
</tr>
<tr>
<td>Gastos</td>
<td>55.150 €</td>
<td>60.499 €</td>
<td>83.501 €</td>
</tr>
<tr>
<td>Beneficio Neto</td>
<td>75.600 €</td>
<td>156.592 €</td>
<td>289.725 €</td>
</tr>
</tbody>
</table>

7. Equipo

- Jonathan Hurtado Yrula, CEO

Estudiante de Ingeniería en Telecomunicaciones. Experiencia como desarrollador y project manager en grandes empresas como Telefónica Digital, así como en startups como Invoost o cu4ea. jonathanyrula@healtheye.es.

II. Problem / Opportunity

Potential users enters HealthEye's market the moment they cannot know people with the same rare disease as themselves and even when they do not know anything about it, nor even specialised doctors, researches, and this is the moment when they are totally lost. The majority of social networking platforms focus on chronic diseases, but there are many people that need to be connected and know all the available information with a rare disease, so our purpose is to let users convey their true feelings and share everything they know with others like them through the most important tool in these moments: information.
Today, potential users are not truly addressing that issue. They are:

- Recently diagnosed with a rare disease between 35-45 year old, and they don't know what to do.
- Parents with a recently born child diagnosed with a rare disease and with the fully need to know everything to save their child or make him a better life.
- People with the need to share their stories and resources such as physiotherapists, psychologist or specialized doctors to help others like them.

**III. Timing: why now?**

**Recent trends that make our solution possible:**

- **There are a 6.5% of people with a rare disease in each country**

According to WHO, in each country there are a 6.5% of people that have a rare disease. Actually, we have registered more than 7,000 rare diseases in the planet, and not all of them have resources to research them or know the causes.

- **Pharmas giants are shifting towards social networks**

Pharmas like Pfizer or Roche have their own foundation to research about rare diseases. But, the real need is that the want to provide to the people that are not-healthy with everything that they are doing, and they want to spread out to the social networks market, first, for the presence and the benchmarking, and the second reason to make profitable the possible medicine for a specialised rare disease.

- **Major LATAM people prefer social networks to discover and meet others**

A study from the McKinsey Company indicates that medical social networks are used to discover what kind of patients have a concrete disease, what range of ages, the possible causes of the appearance, all this kind of information would be very useful in a social network instead a big database with just names and places where these people lives.

Source: Mckinsey and WHO.

**IV. Solution:**

**Concrete social network for rare diseases**

For those who have a rare disease and want to be connected with people like them and discover all the available information of their illness and with the support of specialised doctors in the platform HealthEye is what these kind of people need.
V. Business Model

HealthEye is based on a freemium system for the users of the platform, but we will charge for the following to different places:

Revenue generation

**B2B**

- Deals with existing companies such as private physiotherapy clinics, psychology and services of appliances specialised for people with limited mobility. We will offer to the HealthEye's users a discount in these kind of business, and these ones, paid to HealthEye a percent per client.

- Deals with charities of rare diseases and pharmas focused in these kind of diseases - Product integration

<table>
<thead>
<tr>
<th></th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2B Revenues</td>
<td>60.750 €</td>
<td>85.601 €</td>
<td>123.179 €</td>
</tr>
</tbody>
</table>

The details of this table are based on the economical plan, show it later.

- Geo-location advertising:

Data is based on Facebook and Twitter web standards. This will be B2C, we will use this when we've got more users in the platform. This table, is explaining our previsions.

<table>
<thead>
<tr>
<th>Rate</th>
<th>10-15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page views/month</td>
<td>30</td>
</tr>
<tr>
<td>Rate</td>
<td>85-90%</td>
</tr>
<tr>
<td>Page views/month</td>
<td>4</td>
</tr>
<tr>
<td>CTR</td>
<td>8,00%</td>
</tr>
<tr>
<td>Revenue/100click</td>
<td>0,62 €</td>
</tr>
<tr>
<td>Revenue/1000 clicks</td>
<td>5,5 €</td>
</tr>
</tbody>
</table>

CTR-→ Number of clicks/ times shown. (297k impressions)

The details of this table are based on the economical plan, show it later.
<table>
<thead>
<tr>
<th>Year</th>
<th>Total Revenues</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015</td>
<td>70,000 €</td>
</tr>
<tr>
<td>2016</td>
<td>132,300 €</td>
</tr>
<tr>
<td>2017</td>
<td>250,047 €</td>
</tr>
</tbody>
</table>

**Cost Structure**

- Product development and maintenance
- User acquisition

**VI. Strategic Analysis**

<table>
<thead>
<tr>
<th>Threats</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key players in the market coming up with a similar solution</td>
<td>Market Research</td>
</tr>
<tr>
<td>Existing competitors</td>
<td>Different target market and clear differentiation</td>
</tr>
<tr>
<td>Lack of Market Validation</td>
<td>New website with renovated design will be the “moment of truth” for market validation</td>
</tr>
<tr>
<td>Low demand in Spain</td>
<td>There is a large market in South America, which is the natural extension of the Spanish Market.</td>
</tr>
<tr>
<td>Existence of substitutes</td>
<td>Although substitutes will limit the potential profits of the market, it is also an opportunity for us to lure new customers. Right now, we are the substitute as well as complement for other services.</td>
</tr>
</tbody>
</table>

**Entry barriers**

- Need to leverage on key partners to get traction, which could lead to brand dissolution.
- Low switching cost for business.
- Dependency on medical network to grow userbase.

**VII. Marketing Plan**

1. **Segmentation**

**Case Uses: Scenarios and People**

- Recently born child

James and Paula's first daughter is a few weeks shy of her first birthday, but she did not even crawl yet, nor even is babbling her first words. They go to the hospital and the little Paula is diagnosed with the Wolf-Hirschonn Syndrome a rare disease. Their parents are totally desperate because the doctors can't give them information of what to do with her. They search whatever on the Internet and they find a social network to connect with other parents with childs diagnosed with this horrible syndrome.
Specialised doctor

There are 400 million with a rare disease in the planet; of these, there are around 900 rare diseases with just no more that 10 cases in a country. Luisa, that has the Sjögren Syndrome, connected with a specialised doctor in Germany through HealthEye and they want to make a trial with humans, starting with her. The costs of accommodation and more are expensive, but Luisa connected with other woman with Sjögren Syndrome, and they are going to live together while the trial goes.

New psychologist in the city

Taylor is a new psychologist that want to help the families who have an illness person around them and has built a private clinic. He need people, of course, and they decide to search it in a website where all the fellows he want to study and help are, so he decide to invest some money in advertising. When he made this decision, just in a couple of months their clinic was a reference through these families.

2. Market Size

Total addressable market: social networks users

Spain: 18.5M

Latin America$: 243,4M

Total: 280,4M

Segmented Adressable Market: Rare diseases affected

Spain: 3M

Latin America:

Brasil: 13M

Mexico: 7,8M

Argentina: 2,7M

Total: 26,5 M

Share of market

Spain: 0,02% (600 users) within the last month of 2014..

Latin America: 0,015% (37200 users) within the second year of the launch in LATAM.
3. S.W.O.T. Analysis

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>- First social network focused on rare diseases.</td>
<td>- Need for excellent strategy for fast user acquisition.</td>
</tr>
<tr>
<td>- Broad appeal: families and niche.</td>
<td>- Limited budget.</td>
</tr>
<tr>
<td>- Backed by Telefónica, Orange and UEIA.</td>
<td>- Need for funding.</td>
</tr>
<tr>
<td>- Easy scalability.</td>
<td>- Brand awareness to be created.</td>
</tr>
<tr>
<td>- Cross platform availability.</td>
<td></td>
</tr>
<tr>
<td>- Strong PR connections.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Market size</td>
<td>- Big players coming up with a similar product (competition)</td>
</tr>
<tr>
<td>- Potential partnership with well-known companies (e.g. Telefonica or Orange)</td>
<td></td>
</tr>
<tr>
<td>- 85% of competitors do not include rare diseases in their platforms.</td>
<td>- Big players have more resources.</td>
</tr>
<tr>
<td>- Latin America has the fastest growing internet population. +12% in the past year to more than 147 million unique visitors in 2013.</td>
<td>- Competitors could copy the idea.</td>
</tr>
<tr>
<td>- Consumers in Latin America spent 10 hours online per month on social networking sites; 5 of the top 10 most engaged markets with social content worldwide are located in Latin America. *</td>
<td></td>
</tr>
<tr>
<td>- 2012-2013 increase in social media use in Western Europe (+10%) and Asia-Pacific (+21%). **</td>
<td></td>
</tr>
</tbody>
</table>

** emarketer: Your social media strategy.

*comScore: Latin America Future in Focus.
4. Positioning

- **HealthEye**
  - Rare Diseases: Physiotherapy & Psychologists
  - Social Network: MyHealthTeams

5. List of competitors

<table>
<thead>
<tr>
<th>Company</th>
<th>PatientsLikeMe</th>
<th>MyHealthTeams</th>
<th>Personasque</th>
<th>MedHelp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>USA</td>
<td>USA</td>
<td>Spain</td>
<td>USA</td>
</tr>
<tr>
<td>Payment</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Additional Monetisation</td>
<td>Hospitals and pharmas</td>
<td>Doctors</td>
<td>Created by a pharma</td>
<td>Doctors and hospitals</td>
</tr>
<tr>
<td>Languages</td>
<td>EN</td>
<td>EN</td>
<td>10</td>
<td>EN</td>
</tr>
<tr>
<td>Users</td>
<td>250k</td>
<td>50k</td>
<td>5,000</td>
<td>12M</td>
</tr>
<tr>
<td>Web interface</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Rare diseases</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Geolocation</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

6. Competitive Advantage

<table>
<thead>
<tr>
<th>Name</th>
<th>Features</th>
<th>Threats</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personasque</td>
<td>- Created by a pharma.</td>
<td>- Distrust of the users, there is a complete anonymity.</td>
<td>- We are not focused in just 5 chronical, we have rare diseases.</td>
</tr>
<tr>
<td></td>
<td>- Focus on 6 chronical diseases.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Doctors in the platform.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MyHealthTeams</td>
<td>- Good team with strong PR contacts.</td>
<td>- People with other diseases want to use it, they feel displaced.</td>
<td>- Open new communities when users ask for it.</td>
</tr>
<tr>
<td></td>
<td>- Trust of the users</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7. Vision

We are always worried about work, about our partner or about the discussion with a friend. What happens if you have all these problems and you are healthy? Nothing, you just fixed. When the bad luck in health is close to us, all these problems are nothing, we just want to overcome and fight for our life, or for the person who we love. We strongly believe that what you need, is closer that you think.

8. Mission

HealthEye provides an innovative way to connect people with the same rare diseases. The form is a website platform, for users who seek to stay connected with people that have their same rare disease, and discover all the actual information about their disease, in an easy, fast and more highly personal way who value a product that can be seemingly integrated with their lifestyle on the move.

9. Values

Satisfied users, long-term success, contribution to the community.

10. Slogan

'Always with you'

11. What we have done

- **December 2013**
  Prototype development for contacted users through Twitter and Facebook.

- **February 2014**
  UEIA Accelerator period starts.

- **First fortnight March 2014**
  Launch the platform with diverses functionalities for everybody, open it with 10 rare diseases.

- **Second fortnight March 2014**
  We will focus each 15 days in a rare disease, making interviews, presenting to everyone overwhelming experiences. We started with the AME.

- **April 2014**
  B2B deals with physio and psychologist, Incorporation of specialised doctors of the Instituto de Salud Carlos III and Hospital Universitario La Paz.

- **May - July 2014**
  User growth.
12. Demand - Projections for Spain

Social Networks Market Share in Spain | 79%
---|---
Social Networks Users in Spain | 11% (families and people with rare diseases)

In Spain, there are a 79% of people that every day use social networks, but concretely in Spain, users living directly or indirectly with a rare disease, is around 11%, 4,3 M of people. But this percent is based on the knowledge about the percent of internet penetration in Spain: 54%. So, our market is 2,1M of people.

Source: Kantar World Panel 2013 and WHO.
13. Demand- projections for LATAM

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>%Market Share HealthEye</td>
<td>0,00%</td>
<td>0,00%</td>
<td>0,00%</td>
<td>0,07%</td>
<td></td>
</tr>
<tr>
<td>Users in Latam</td>
<td>265</td>
<td>530</td>
<td>1060</td>
<td>1855</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>%Market Share HealthEye</td>
<td>0,01%</td>
<td>0,02%</td>
<td>0,03%</td>
<td>0,04%</td>
<td></td>
</tr>
<tr>
<td>Users in Latam</td>
<td>2650</td>
<td>5300</td>
<td>7950</td>
<td>10600</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LATAM</th>
<th></th>
<th>01/05/15</th>
<th>June 2015</th>
<th>July. 2015</th>
<th>Aug 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>%Market Share HealthEye</td>
<td>0,05%</td>
<td>0,05%</td>
<td>0,05%</td>
<td>0,06%</td>
<td></td>
</tr>
<tr>
<td>Users in Latam</td>
<td>11200</td>
<td>12500</td>
<td>14200</td>
<td>15000</td>
<td></td>
</tr>
</tbody>
</table>

VIII. Company Ownership

1. Founder

Jonathan Hurtado Yrula: CEO

Jonathan is a Telecommunications Engineer graduated from the University Politechnic of Madrid, ETSIT-UPM. He was selected through 7,000 students to enjoy the Talentum Scholarship granted by Telefonica Digital. In addition, he worked in startups like Invoost or cuaqea.

Company Summary & Location

HEALTHEYE is a limited society which was founded on ---

Its VAT number is ---

The legal address is ----

2. Board of Advisors

We count with the following board of advisors:

- Mario López de Ávila

He is the founder of NODOS, professor at IE Business School and successful entrepreneur.
Félix González Herranz
Succesful social entrepreneur, Telecommunications Engineer and M.Sc, Stanford.

Jose María Llorente
He worked in LATAM for succesful enterprises with an excellent vision of digital startups.

Ana Martín
She worked in the Minister of Health and she is a financial excellent woman.

3. Legal

HEALTHEYE S.L. is a company whose registered office is in Madrid, Avenida de Asturias, number 19, second B (28029), with VAT number --- and duly registered in the Mercantile Registry of Madrid in Volume ---. The contact email is jonathanyrula@healtheye.es

Terms of Service

HEALTHEYE's Terms of Service govern the users' access to and use the services. The user's access to and use the Services are conditioned on their acceptance of and compliance with these Terms. By accessing or using the Services they agree to be bound by these Terms.

Users are responsible for their use of the Services, for any Content they share to the Services, and for any consequences thereof.

Users may use the Services only if they are not a person barred from receiving services under the laws of the Spanish Government. If they accept the Terms of Services, on behalf of a company, organization, government, or other legal entity, they represent and warrant that they are authorized to do so.

Privacy Policy

HEALTHEYE operates in accordance with the current Spanish legislation regarding data protection and in particular Law 15/1999 of December 13, Data Protection Personal Development and Regulation 1720/2007, of December 21 and all European and Spanish legislation applicable to this matter, data voluntarily provided by the User in the registration form will be included for treatment to a HEALTHEYE file duly registeres by the Spanish Agency of Data Protection with the purpose of identifying and contacting users, and providing information

HEALTHEYE takes measures to maintain the level of security required. When users sign up to use HEALTHEYE we collect some personal information about them. This can be either their email address, their name or their disease. This information will enable them to use our application and to share content to other applications. We do not share this data with any third partiers.
IX. Financial Projections

1. Current Status

<table>
<thead>
<tr>
<th></th>
<th>Jan 2014</th>
<th>Febr. 2014</th>
<th>March 2014</th>
<th>April 2014</th>
<th>01/05/14</th>
<th>June 2014</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Development</td>
<td>385 €</td>
<td>385 €</td>
<td>385 €</td>
<td>385 €</td>
<td>385 €</td>
<td>385 €</td>
<td>2310 €</td>
</tr>
<tr>
<td>SGA</td>
<td>25 €</td>
<td>25 €</td>
<td>25 €</td>
<td>25 €</td>
<td>25 €</td>
<td>25 €</td>
<td>150 €</td>
</tr>
<tr>
<td>Total</td>
<td>510 €</td>
<td>510 €</td>
<td>510 €</td>
<td>510 €</td>
<td>510 €</td>
<td>510 €</td>
<td>3060 €</td>
</tr>
<tr>
<td>Total assets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3060 €</td>
</tr>
</tbody>
</table>

We currently have €3,000 raised. The founder put the entire quantity. We currently have an operating buffer of 6 months with current financing and are seeking to raise €50,000 in exchange of 15% equity. The use of these funds will provide a cushion for delayed operations, increased software development, and user acquisitions techniques to consolidate our position in the market.

2. Funding needs

* User interface and design, change of the platform to Ruby on Rails. Salary.

** User Acquisition Campaigns, Digital Marketing Campaigns, Advertising, High-Profile Endorsements. for Spain and LATAM.
HEALTHEYE: THE SOCIAL NETWORK FOR RARE DISEASES

3. Breakdown and strategy up to March 2016:

<table>
<thead>
<tr>
<th></th>
<th>Jan. 2015</th>
<th>01/02/15</th>
<th>Marc. 2015</th>
<th>April. 2015</th>
<th>01/05/15</th>
<th>June 2015</th>
<th>July 2015</th>
<th>Aug. 2015</th>
<th>Sept. 2015</th>
<th>01/10/15</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Development</strong></td>
<td>364 €</td>
<td>364 €</td>
<td>364 €</td>
<td>364 €</td>
<td>364 €</td>
<td>364 €</td>
<td>364 €</td>
<td>364 €</td>
<td>364 €</td>
<td>364 €</td>
</tr>
<tr>
<td><strong>Marketing &amp; Sales</strong></td>
<td>1,000 €</td>
<td>1,000 €</td>
<td>1,000 €</td>
<td>1,000 €</td>
<td>1,000 €</td>
<td>1,000 €</td>
<td>1,000 €</td>
<td>1,000 €</td>
<td>1,000 €</td>
<td>1,000 €</td>
</tr>
<tr>
<td><strong>International Expansion</strong></td>
<td>1,388 €</td>
<td>1,388 €</td>
<td>1,388 €</td>
<td>1,388 €</td>
<td>1,388 €</td>
<td>1,388 €</td>
<td>1,388 €</td>
<td>1,388 €</td>
<td>1,388 €</td>
<td>1,388 €</td>
</tr>
<tr>
<td><strong>SGA</strong></td>
<td>278 €</td>
<td>278 €</td>
<td>278 €</td>
<td>278 €</td>
<td>278 €</td>
<td>278 €</td>
<td>278 €</td>
<td>278 €</td>
<td>278 €</td>
<td>278 €</td>
</tr>
<tr>
<td><strong>Total (€)</strong></td>
<td>4,524 €</td>
<td>4,524 €</td>
<td>4,524 €</td>
<td>4,524 €</td>
<td>4,524 €</td>
<td>4,524 €</td>
<td>4,524 €</td>
<td>4,524 €</td>
<td>4,524 €</td>
<td>4,524 €</td>
</tr>
</tbody>
</table>

**ASSUMPTIONS**

// Product Development → First 12 months 75% then 25% for maintenances, change to Ruby on Rails when we’ll get more users.

// Business Development → First 12 months 75 % then 825 % we’ll pay a marketing profile in LATAM, to obtein users, doctors and charities these years in Mexico, Brasil and Argentina.
4. **User Acquisition Cost (assuming only paid campaigns)**

<table>
<thead>
<tr>
<th></th>
<th>Cost</th>
<th>% total</th>
<th>Users</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Marketing</td>
<td>16.000</td>
<td>43 %</td>
<td>5000</td>
<td>3,20 €</td>
</tr>
<tr>
<td>Campaigns (aggregated)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Adwords</td>
<td>8.000</td>
<td>14 %</td>
<td>750</td>
<td>6,67 €</td>
</tr>
<tr>
<td>High-Profile</td>
<td>16.000</td>
<td>43 %</td>
<td>15000</td>
<td>1,07 €</td>
</tr>
<tr>
<td>endorsements</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>40.000</td>
<td></td>
<td>20750</td>
<td></td>
</tr>
<tr>
<td><strong>ACPU</strong></td>
<td></td>
<td></td>
<td></td>
<td>4,23 €</td>
</tr>
</tbody>
</table>

We will prove with campaigns in Facebook, Google Adwords, Radio and TV. Each month, we will pay around 1,000€.

5. **Customer Lifetime Value**

*Everything is in months.*

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average Life Users</strong></td>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ARPAU (Average Revenue per active users)</strong></td>
<td>0,33 €</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Churn Rate</strong></td>
<td>10 %</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Customer Retention</strong></td>
<td>90 %</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Discount Rate</strong></td>
<td>12 %</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(Months x Revenue) x Span</strong></td>
<td>7,92 €</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ACPU</strong></td>
<td>4,23 €</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Net</strong></td>
<td>3,69 €</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FORMULA**

Margin x % Retention Rate /1 + Discount Rate – Retention Rate

**AVERAGE CLV**

19 €

*ARPAU-*> Each heavy user make win 0,33€ to HealthEye.

*Churn Rate-*> Users than leave HealthEye.

6. **B2B Details:**

This is for the first year

<table>
<thead>
<tr>
<th></th>
<th>Psychologists</th>
<th>Phisios</th>
<th>Aparatology</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Estimated sales</strong></td>
<td>2500</td>
<td>3000</td>
<td>3</td>
</tr>
<tr>
<td><strong>Annual increase</strong></td>
<td>10,00%</td>
<td>50,00%</td>
<td>50,00%</td>
</tr>
<tr>
<td><strong>sales</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Price</strong></td>
<td>8 €</td>
<td>12 €</td>
<td>2.250 €</td>
</tr>
<tr>
<td><strong>Annual increase</strong></td>
<td>5,00%</td>
<td>5,00%</td>
<td>5,00%</td>
</tr>
<tr>
<td><strong>price</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>30.000 €</td>
<td>24.000 €</td>
<td>6.750 €</td>
</tr>
</tbody>
</table>

*Each number is estimated by year.*
7. **B2C Details:**

Suposed a 10% of users click in our publicity, we will obtain, everything with the 0,062 € cost of the click:

<table>
<thead>
<tr>
<th></th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Revenues</td>
<td>70.000 €</td>
<td>132.300 €</td>
<td>250.047 €</td>
</tr>
</tbody>
</table>

8. **B2B and B2C Details**

<table>
<thead>
<tr>
<th></th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2B</td>
<td>60.750 €</td>
<td>85.601 €</td>
<td>123.179 €</td>
</tr>
<tr>
<td>B2C</td>
<td>70.000 €</td>
<td>132.300 €</td>
<td>250.047 €</td>
</tr>
<tr>
<td>TOTAL</td>
<td>130.750 €</td>
<td>217.091 €</td>
<td>373.226 €</td>
</tr>
</tbody>
</table>

Everything is justified in the next point.

9. **Profit and Lost**

Everything is in €

<table>
<thead>
<tr>
<th></th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenues</td>
<td>130.750 €</td>
<td>217.091 €</td>
<td>373.226 €</td>
</tr>
<tr>
<td>Purchase</td>
<td>0 €</td>
<td>0 €</td>
<td>0 €</td>
</tr>
<tr>
<td>Inventories</td>
<td>0 €</td>
<td>0 €</td>
<td>0 €</td>
</tr>
<tr>
<td>Personal</td>
<td>36.400 €</td>
<td>57.330 €</td>
<td>80.262 €</td>
</tr>
<tr>
<td>Rental</td>
<td>2.400 €</td>
<td>2.448 €</td>
<td>2.496 €</td>
</tr>
<tr>
<td>Other</td>
<td>700 €</td>
<td>721 €</td>
<td>742 €</td>
</tr>
<tr>
<td>Amortization</td>
<td>250 €</td>
<td>250 €</td>
<td>250 €</td>
</tr>
<tr>
<td>EBIT</td>
<td>91.000 €</td>
<td>157.152 €</td>
<td>289.474 €</td>
</tr>
<tr>
<td>Financial costs</td>
<td>0 €</td>
<td>0 €</td>
<td>0 €</td>
</tr>
<tr>
<td>Profit before TAX</td>
<td>91.000 €</td>
<td>157.152 €</td>
<td>289.474 €</td>
</tr>
<tr>
<td>Income TAX</td>
<td>27.300 €</td>
<td>47.145 €</td>
<td>86.842 €</td>
</tr>
<tr>
<td>Profit after TAX</td>
<td>63.700 €</td>
<td>110.007 €</td>
<td>202.632 €</td>
</tr>
</tbody>
</table>
10. CashFlow

<table>
<thead>
<tr>
<th></th>
<th>2014</th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Investment</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-current</td>
<td>1.000 €</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>assets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Current Assets</strong></td>
<td>2.000 €</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>3.000 €</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Charges</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sales</td>
<td>120.003 €</td>
<td>210.738 €</td>
<td>360.460 €</td>
<td></td>
</tr>
<tr>
<td><strong>Total Charges</strong></td>
<td>120.003 €</td>
<td>210.738 €</td>
<td>360.460 €</td>
<td></td>
</tr>
<tr>
<td><strong>Payments</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personal</td>
<td>36.400 €</td>
<td>57.330 €</td>
<td>80.262 €</td>
<td></td>
</tr>
<tr>
<td>Household</td>
<td>2.400 €</td>
<td>2.448 €</td>
<td>2.496 €</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>700 €</td>
<td>721 €</td>
<td>742 €</td>
<td></td>
</tr>
<tr>
<td>EBIT</td>
<td>27.300 €</td>
<td>47.145 €</td>
<td>86.842 €</td>
<td></td>
</tr>
<tr>
<td><strong>Total Payments</strong></td>
<td>66.800 €</td>
<td>107.644 €</td>
<td>170.344 €</td>
<td></td>
</tr>
<tr>
<td><strong>CashFlow</strong></td>
<td>-3.000 €</td>
<td>53.203 €</td>
<td>103.093 €</td>
<td>190.116 €</td>
</tr>
</tbody>
</table>

Discount Rate: 8.00%

NVA: 285.568 €

IRR: 18.65%
CONCLUSIONS

We’ve been able to see and show how to create a web application based on the MVC philosophy through the Symfony framework, PHP, HTML, CSS, Yaml, and Javascript.

HealthEye is not just a final degree project, but a reality that actually has helped more than 400 users currently living in Spain. We started the last February of 2014, with just 10 rare diseases.

Now, we have more than 55 diseases and growing each week. That is why HealthEye is not just a final project to become an engineer, but one example of a good use of the technology by simply creating a website.

If we talk about the technology, we have seen how to create a MVC application with all the steps together. We made something similar in the subject CORE, of the third grade and we have changed a little bit the languages used, but with the same steps we could say.

Talking about the execution of the idea to turn it on into a real project with real users, I certainly have no doubts that I couldn’t do that without the help of many people in the path.

The first thing we did, was be clear with the idea, by just creating what was necessary in the platform and make it different from the competence. Then we made the first contacts with charities and specialized doctors to get their crucial support.

That is why, I consider HealthEye the most important thing that I’ve done in college: to create something new, from nothing, so thank you very much those who has helped me in this long and hard path.

FUTURE WORK

In this section, we will talk about the future of HealthEye and how to make it a most usefull tool for the final users.

The most important thing, is the internationalization of the project. To do so, we will need an invest to create internet campaigns and have economics benefits to start in Latin America at the finish of the year.

We have to improve too, some things of the website, to make it more easier to the users. That’s a very interesting point and maybe we will need a developer just dedicated to this task.
1. **BOOKS**

- “Desarrollo Ágil con Symfony 2.1” – Javier Aguiluz
- “Finanzas para no financieros” – Barbara Soriano

2. **WEBS**

http://stackoverflow.com

To resolve all the technical issues I had while coding the platform.

http://symfony.es/

The oficial doc in Spain for Symfony.

http://librosweb.es/libro/symfony_2_x/capitulo_2.html

Awesome book to discover the Symfony framework in a deep way that others.

http://www.maestrosdelweb.com/curso-symfony2-introduccion-instalacion/

Introduction to all the secrets of Symfony

https://www.youtube.com/watch?v=jPxPtVBboDI

Essencial video to the installation of Symfony in Ubuntu 12.04.

https://hbr.org/2011/01/how-to-design-a-winning-business-model

How to create a business model for a successful investor
ANNEX I: AWARDS AND SCHOLARSHIPS

   
   http://blogthinkbig.com/proyectos-talentum-startups-ericsson/

2. October 2013 - Third Prize in “Concurso Idea Emprendedora” between more than 1.500 projects - IEBS – Business School Barcelona
   
   http://comunidad.iebschool.com/concurso-emprendedores/ganadores-2013/

3. October 2013 – START PFC Scholarship – Orange Chair in ETSIT – UPM
   
   http://catedra-orange.upm.es/59-healtheye-una-historia-de-emprendimiento-enprimera-persona

   
   http://www.ueia.com/project/health-eye/

5. October 2014 - Global Management Programme Telefónica & IESE

   Thanks to HealthEye I was selected for Telefonica to join this Master in BCN.

   

7. December 2014 - First Prize to the best Social Project – Fundación MasHumano & Minister of Health and Social Affairs in Spain
   
   http://www.injuve.es/empleo/noticia/9-premio-jovenes-emprendedoresmashumano
   

   
   http://www.revolucionalasalud.es/
ANNEX II: PRESS CLIPPING


2. February 2014 – RTVE

http://www.rtve.es/alacarta/audios/tendencias/tendencias-health-eye-27-02-14/2419853/

3. August 2014 – Revista Emprendedores

http://www.emprendedores.es/ideas-de-negocio/red-social-enfermedades-raras


http://www.ueia.com/project/health-eye/

5. April 2014 – Instituto de la Ingeniería de España

http://www.iies.es/Los-ingenieros-de-telecomunicacion-de-la-UPM-presentansus-ideas-emprendedoras_a3615.html

6. July 2014 – Expansión


7. September 2014 - Think Big Telefonica Blog

http://blogthinkbig.com/healtheye/

8. October 2014 – RSC Telefónica

http://www.rcysostenibilidad.telefonica.com/blog/2014/10/06/nace-un-nuevoreto-por-la-discapacidad-nace-el-reto-ability/#.VSEWENysWcw

9. December 2014 – Actua UPM Blog

http://actuaupm.blogspot.com.es/2014/12/lista-de-premiados-de-la-xiactuaupm.html

10. December 2014 – EFE Emprende