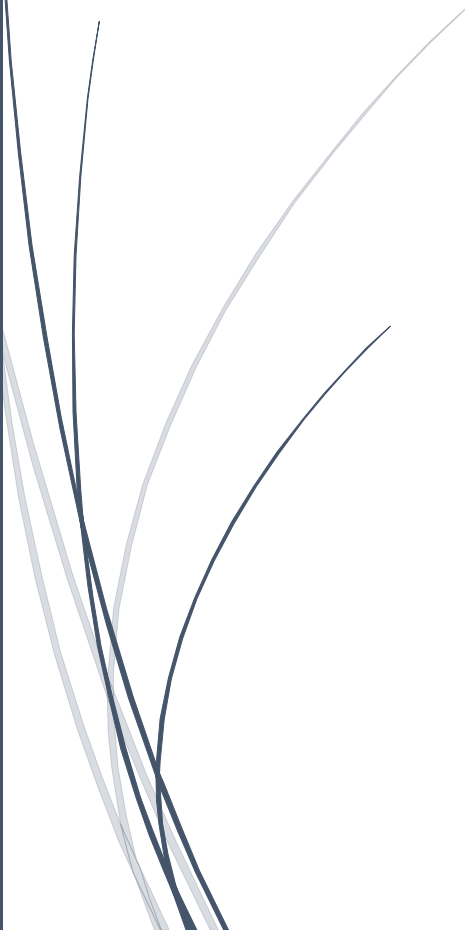




07/12/2015

# OpenIdeas

Documento architettuale



Ciro Amati, Stefania Cardamone, Amedeo Leo,  
Simone Romano  
PROGETTO WEB SEMANTICO

## Contents

Introduzione .....	2
Analisi dei requisiti .....	4
Gestione Accessi.....	4
Gestione Utenti .....	4
Gestione Idee.....	4
Gestione notifiche e email.....	4
Requisiti funzionali .....	5
Requisiti non funzionali .....	6
Casi d’uso.....	7
Gestione accessi .....	7
Gestione utente.....	10
Gestione idea.....	14
Manuale utente .....	19
Architettura e tecnologie .....	26
Architettura del sistema.....	26
Gestione dati .....	28
Tecnologie utilizzate .....	30
Template Responsive .....	34
Semantic Web.....	36
Sentiment analysis.....	36
Linked data .....	38
RDFa.....	38
Sicurezza .....	40
Email e notifiche .....	40
Workflow .....	41
Visione globale .....	41
Inserimento idea.....	42
Gestione autenticazione.....	43
Follow idea .....	44
Finanziamento idea .....	45
Commento idea .....	46
Conclusioni .....	47
References.....	48

## Introduzione

Lo scopo del portale *OpenIdeas* è dare la possibilità ad utenti che hanno voglia di realizzare un progetto (di qualsiasi tipologia) di

- Renderlo pubblico;
- Avere un feedback sul giudizio degli utenti;
- Trovare eventuali finanziatori.

Il sistema prevede la necessità di registrarsi per poter inserire un'idea, ma non per poterle visualizzare. L'idea dovrà appartenere ad una categoria già esistente del portale. La pubblicazione di un'idea è necessaria per poter cercare un finanziatore, ma allo stesso tempo per avere dei commenti da parte degli altri utenti, in modo da capire la validità della stessa.

Sono diverse le funzionalità del sistema:

- Gestione degli utenti con le classiche funzionalità di login (con convalida);
- Login con l'account facebook, logout, recupero password;
- Gestione notifiche;
- Interazione con la knowledge base del sistema contenente alcune informazioni pubbliche;
- Gestione delle idee.

La gestione delle idee rappresenta il core della applicazione web presentata. Infatti OpenIdeas nasce per offrire l'opportunità agli utenti di rendere pubblica una propria idea progettuale e per trovare eventuali finanziatori.

Tra le varie tecnologie utilizzate, un focus maggiore è stato dato ad alcune tecnologie del semantic web molto recenti ed attuali. Come accennato in precedenza, esiste una parte di dati pubblica. Tale parte contiene proprio delle informazioni riguardanti le idee inserite nel portale. In particolare è stata realizzata una knowledge base che contiene gli elementi chiave di una idea generica; è stata caricata su un webservice Fuseki ed è stata realizzata una interfaccia con cui è possibile interrogare (tramite query SPARQL) l'endpoint per ottenere risultati in diversi formati (es. xml).

Il sistema è in grado di classificare ogni idea (assegnandole una serie di punteggi) sulla base dei commenti inseriti dagli utenti. Per fare ciò sono state utilizzate delle tecniche di sentiment analysis.

Altra caratteristica interessante è data dalla presenza, all'interno dell'html, di codice RDFa. In questo modo le pagine html risultano comprensibili non solo all'utente che le legge ma anche a motori di ricerca o a software di terze parti.



Inserimento



Finanziamento



Commento



Segui

I dettagli di quanto introdotto sono disponibili nelle sezioni successive del documento che sono così strutturate:

- La sezione **Analisi dei requisiti** spiega i possibili utilizzi del sistema tramite una serie di scenari reali;
- La sezione **Manuale utente** guida l'utente nell'utilizzo del sistema (dal login all'inserimento di una idea);
- La sezione **Architetture e tecnologie** illustra con una visione di alto livello la architettura del sistema focalizzandosi sulla parte di gestione dati;
- La sezione **Semantic Web** raccoglie i concetti principali della applicazione, ovvero quelli legati alle tecnologie del semantic web utilizzate;
- Infine, le opportune conclusioni.

## Analisi dei requisiti

### Gestione Accessi

Comprende le attività di gestione degli account per l'accesso al sistema. Gli utenti utilizzano la funzione di log-in per autenticarsi e la funzione log-out per uscire dal sistema. In particolare, la funzione di login può essere effettuata sia tramite le credenziali inseriti al momento della registrazione che tramite le API fornite da Facebook. La registrazione è possibile sia attraverso la compilazione dei campi richiesti sia con Facebook.

### Gestione Utenti

Comprende le possibili attività degli utenti. Il sistema permette le seguenti azioni:

- Visualizzazione delle proprie informazioni.
- Visualizzazione delle idee.
- Visualizzazione delle migliori idee.
- Visualizzazione delle notifiche.
- Registrazione al sistema.

### Gestione Idee

È il fulcro del sistema. Il sistema permette di gestire i seguenti aspetti:

- Inserimento di un'idea, con relative informazioni.
- Inserimento di un commento ad un'idea non proprietaria; il sistema ricaverà il punteggio tramite sentiment analysis.
- Possibilità di seguire un'idea per ricevere gli aggiornamenti.
- Possibilità di finanziare un'idea.
- Visualizzazione di tutte le idee.
- Visualizzazione delle informazioni di un'idea.
- Visualizzazione delle idee tramite query SPARQL.

### Gestione notifiche e email

Il sistema permette all'utente di essere in continuo aggiornamento sui cambiamenti delle idee; in particolare:

- Se si ha commentato, si tiene traccia di commenti successivi al proprio commento.
- Se si ha scelto di seguire un'idea, si tiene traccia di nuovi commenti e finanziatore.
- Se si ha inserito un'idea, si tiene traccia di commenti, finanziatore e followers.
- All'atto della registrazione, all'utente viene inviata una email per conferma autenticazione.

## Requisiti funzionali

Di seguito verranno elencati i requisiti funzionali in dettaglio, suddivisi per attori, specificando tra parentesi la relativa priorità (*ALTA*: la funzione verrà implementata; *MEDIA*: la funzione verrà documentata e implementata a seconda del tempo a disposizione; *BASSA*: la funzione non verrà implementata).

- **Gestione accessi:**
  - Tutti gli utenti
    - Login Facebook(ALTA): Ogni utente può effettuare l'accesso alla piattaforma tramite le proprie credenziali di Facebook.
    - Login (ALTA): Ogni utente può effettuare l'accesso alla piattaforma con le proprie credenziali.
    - Logout (ALTA): Ogni utente può effettuare il logout dalla piattaforma.
  
- **Gestione utente:**
  - Tutti gli utenti
    - Visualizzazione dati personali (ALTA): Qualsiasi utente può visualizzare le proprie informazioni.
    - Registrazione al sistema (ALTA): Qualsiasi utente esterno può registrarsi al sistema.
    - Visualizzazione delle idee (ALTA): Qualsiasi utente può visualizzare ogni idea.
    - Modifica informazioni (BASSA): Qualsiasi utente può modificare le proprie informazioni personali.
  
- **Gestione idee:**
  - Tutti gli utenti:
    - Inserimento di un'idea (ALTA): Qualsiasi utente può inserire un'idea con relative informazioni.
    - Inserimento di un commento (ALTA): Qualsiasi utente può inserire un commento ad un'idea non proprietaria. Il sistema ricaverà il punteggio tramite sentiment analysis.
    - Seguire un'idea (ALTA): Qualsiasi utente può seguire un'idea non proprietaria.
    - Finanziare un'idea (ALTA): Qualsiasi utente può finanziare un'idea non proprietaria.
    - Visualizzazione di un'idea (ALTA): Qualsiasi utente può visualizzare ogni idea.
    - Modifica di un'idea (BASSA): L'autore può modificare la propria idea.

## Requisiti non funzionali

Il sistema vedrà come fondamentali i requisiti non funzionali di usabilità, performance, affidabilità e sicurezza:

- **Usabilità:** il sistema dovrà essere utilizzabile da un gruppo molto vasto di utenti.
- **Performance:** il sistema dovrà soddisfare le esigenze di ogni utente nel minor tempo possibile.
- **Affidabilità:** il sistema dovrà essere affidabile e mantenere inalterati tutti i dati contenuti in esso in caso di un funzionamento anomalo o di un errato utilizzo.
- **Sicurezza:** il sistema utilizzerà una funzione di autenticazione in modo da negare l'accesso da parte di utenti non registrati rispettando la tutela della privacy. È prevista una cifratura della password inserita all'atto della registrazione.

## Casi d'uso

## Gestione accessi

ID	UC_AC_01
<b>Preambolo</b>	
L'utente si trova nella pagina relativa al login con l'intenzione di accedere al sistema.	
<b>Caso D'uso</b>	<b>Login</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	L'utente si trova nella schermata di login
<b>Punti di estensione</b>	
<b>Generalizzazione di</b>	
<b>Scenario Principale</b>	
1	"OpenIdeas" mostra la pagina dove inserire i dati necessari al login.
2	L'utente inserisce i dati necessari: <ul style="list-style-type: none"> <li>• Email</li> <li>• Password</li> </ul> E clicca sul bottone per la conferma.
3	"OpenIdeas" legge i dati dal database e mostra la pagina iniziale.
<b>Scenario Secondario</b>	
2.a	L'utente clicca sul bottone per accedere tramite Facebook.
3.a	"OpenIdeas" legge le informazioni dal database relative alla mail di Facebook e mostra la pagina iniziale.



<b>ID</b>	UC_AC_02
<b>Preambolo</b>	
L'utente si trova in qualsiasi pagina di "OpenIdeas" con l'intenzione di effettuare il logout.	
<b>Caso D'uso</b>	<b>Logout</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	L'utente ha effettuato l'accesso al sistema.
<b>Scenario Principale</b>	
1	L'utente clicca sul bottone per effettuare il logout.
2	"OpenIdeas" procede al logout e visualizza la pagina di login.

ID	UC_AC_03
<b>Preambolo</b>	
L'utente si trova nella pagina di login di "OpenIdeas" con l'intenzione di registrarsi.	
<b>Caso D'uso</b>	<b>Registrazione</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	
<b>Scenario Principale</b>	
1	<p>L'utente inserisci i campi:</p> <ul style="list-style-type: none"> <li>• Nome</li> <li>• Cognome</li> <li>• Email</li> <li>• Password</li> <li>• Immagine profilo</li> <li>• Pagina web</li> <li>• Data di nascita</li> </ul> <p>Quindi clicca su "Registrati".</p>
2	"OpenIdeas" procede con l'inserimento nel database delle informazioni e visualizza la pagina iniziale.
<b>Scenario Secondario</b>	
1.a	L'utente clicca sul bottone per effettuare il login con Facebook.
2.a	"OpenIdeas" procede con l'inserimento nel database delle informazioni ottenute tramite le API di Facebook e visualizza la pagina iniziale.

Gestione utente

<b>ID</b>	UC_GU_01
<b>Preambolo</b>	
L'utente si trova in qualsiasi pagina e vuole visualizzare le sue informazioni	
<b>Caso D'uso</b>	<b>Visualizza informazioni</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	
<b>Scenario Principale</b>	
1	L'utente clicca sul proprio username. Quindi clicca su "Profilo".
2	"OpenIdeas" mostra le informazioni dell'utente.

ID	UC_GU_02
<b>Preambolo</b>	
L'utente si trova in qualsiasi pagina e vuole visualizzare la pagina di about del sistema.	
<b>Caso D'uso</b>	<b>Visualizza informazioni sistema</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	
<b>Scenario Principale</b>	
1	L'utente clicca sul bottone per visualizzare le informazioni del sistema.
2	"OpenIdeas" mostra le informazioni del sistema.

<b>ID</b>	UC_GU_03
<b>Preambolo</b>	
L'utente si trova nella pagina iniziale e vuole visualizzare solo le idee di una determinata categoria	
<b>Caso D'uso</b>	<b>Visualizza idee di una categoria</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	
<b>Scenario Principale</b>	
1	L'utente clicca sulla categoria "Software"
2	"OpenIdeas" mostra le idee afferenti alla categoria Software.

ID	UC_GU_04
<b>Preambolo</b>	
L'utente si trova in qualsiasi e vuole visualizzare tutte le idee	
<b>Caso D'uso</b>	<b>Visualizza idee.</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	
<b>Scenario Principale</b>	
1	L'utente clicca su "OpenIdeas".
2	"OpenIdeas" mostra le idee inserite.

## Gestione idea

<b>ID</b>	UC_GI_01
<b>Preambolo</b>	
L'utente si trova nella sezione "Profilo" e vuole inserire un'idea.	
<b>Caso D'uso</b>	<b>Inserimento idea</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	
<b>Scenario Principale</b>	
1	L'utente inserisce il nome della descrizione, quindi clicca su "Step 2" e inserisce un'immagine e l'url del video contenente un riepilogo della propria idea; infine clicca su "Summary" e conferma.
2	"OpenIdeas" inserisce l'idea nel database.

<b>ID</b>	UC_GI_02
<b>Preambolo</b>	
L'utente si trova nell'idea "SmartAPI" e vuole inserire un commento.	
<b>Caso D'uso</b>	<b>Inserimento commento</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	L'utente non è l'autore dell'idea
<b>Scenario Principale</b>	
1	L'utente inserisce il testo del commento, quindi clicca su "Inserisci Commento".
2	"OpenIdeas" inserisce il commento dell'utente nel database; quindi aggiorna i punteggi dell'idea, ottenuti tramite sentiment analysis.



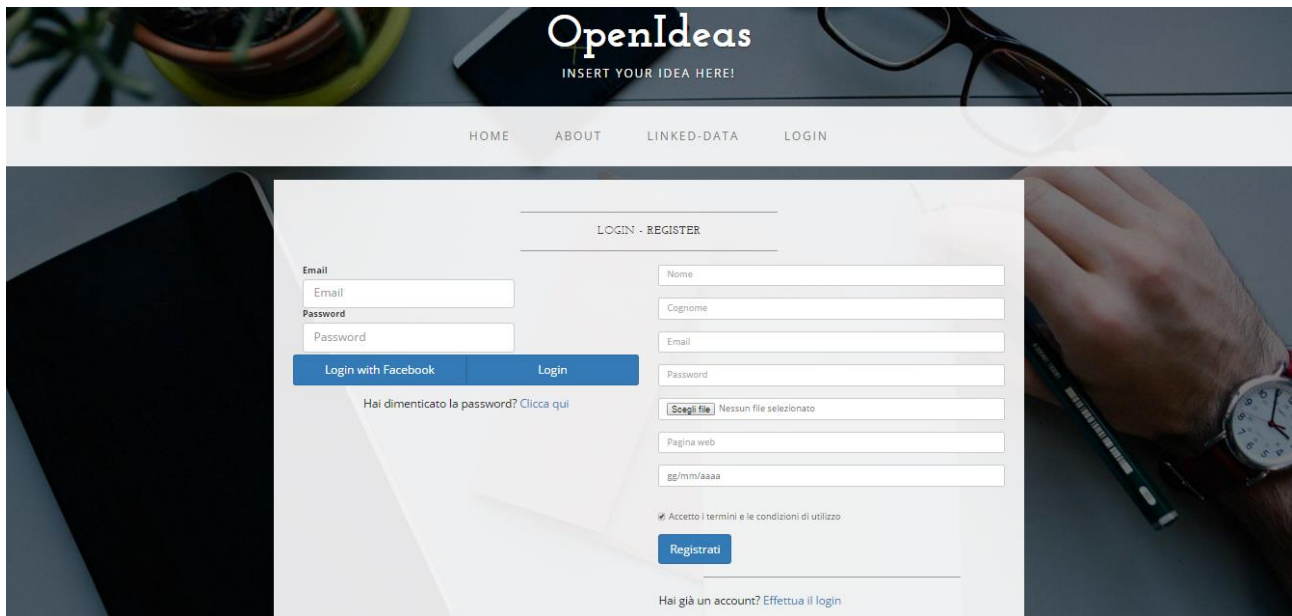
ID	UC_GI_03
<b>Preambolo</b>	
L'utente si trova nell'idea "SmartAPI" e vuole seguirla.	
<b>Caso D'uso</b>	<b>Follow di idea</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	L'utente non è l'autore dell'idea
<b>Scenario Principale</b>	
1	L'utente clicca su "Segui".
2	"OpenIdeas" inserisce l'utente tra i follower dell'idea "SmartAPI" e lo informerà ad ogni successivo aggiornamento.

ID	UC_GI_04
<b>Preambolo</b>	
L'utente si trova nell'idea "SmartAPI" e vuole finanziarla.	
<b>Caso D'uso</b>	<b>Finanziamento idea</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	L'utente non è l'autore dell'idea
<b>Scenario Principale</b>	
1	L'utente clicca su "Finanzia".
2	"OpenIdeas" assegna l'idea all'utente.

ID	UC_GI_05
<b>Preambolo</b>	
L'utente si trova in qualsiasi pagina e clicca su "Linked data".	
<b>Caso D'uso</b>	<b>Query SPARQL</b>
<b>Attori</b>	Utente
<b>Pre-Condizioni</b>	
<b>Scenario Principale</b>	
1	L'utente inserisce il testo della query. Quindi sceglie il formato in output desiderato e clicca su query.
2	"OpenIdeas" mostra all'utente il risultato della query e scarica il relativo file creato.

## Manuale utente

L'accesso al portale prevede una fase iniziale di login. Esso presenta una parte di registrazione ma anche un accesso immediato tramite Facebook.

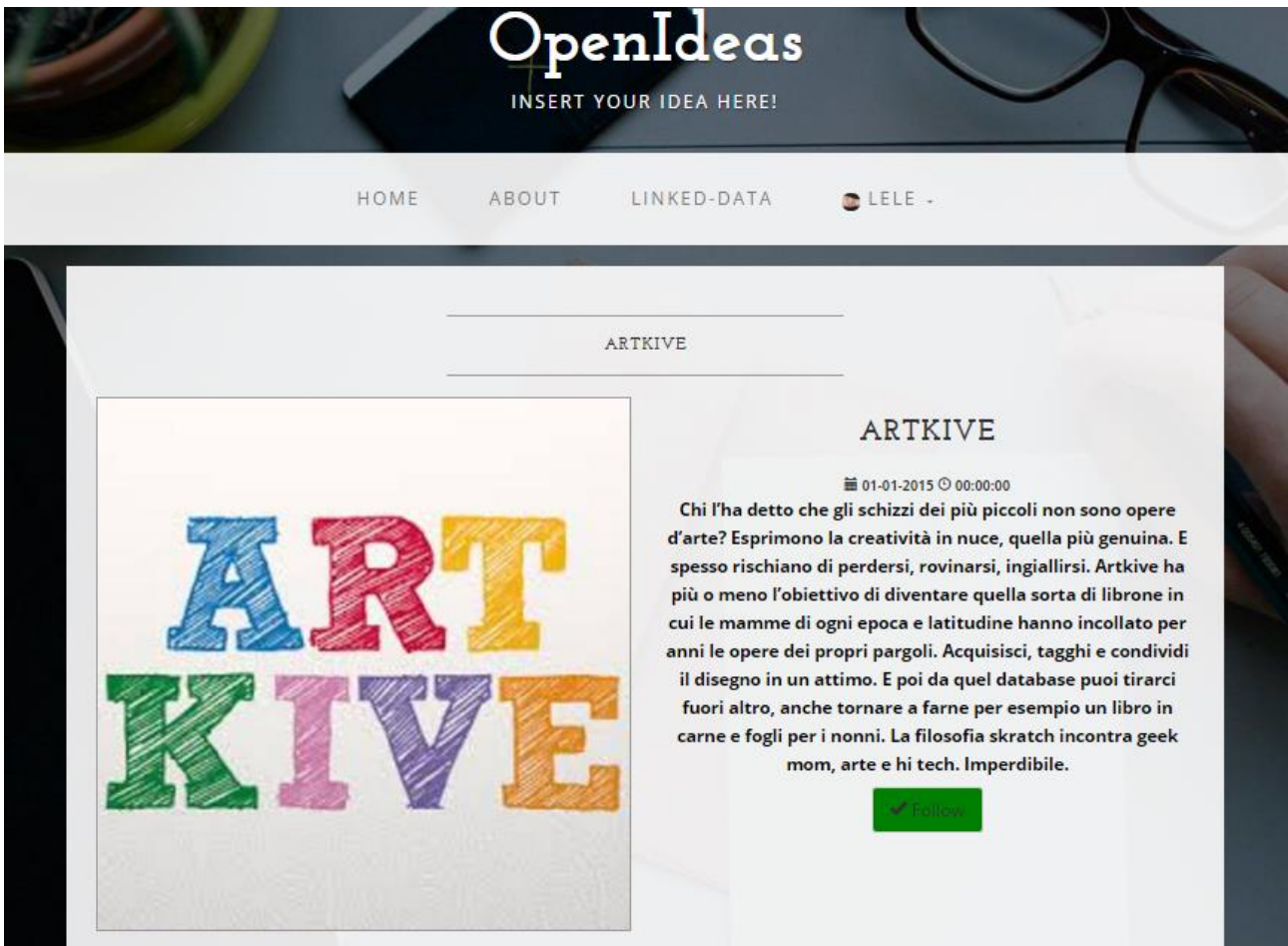


The screenshot displays the OpenIdeas website interface. At the top, the logo "OpenIdeas" is accompanied by the tagline "INSERT YOUR IDEA HERE!". Below the logo is a navigation menu with links for "HOME", "ABOUT", "LINKED-DATA", and "LOGIN". The main content area features a "LOGIN - REGISTER" form. On the left side of the form, there are input fields for "Email" and "Password", a "Login with Facebook" button, and a "Login" button. A link "Hai dimenticato la password? Clicca qui" is positioned below the login buttons. On the right side, there are input fields for "Nome", "Cognome", "Email", and "Password", a file upload field labeled "Scegli file" with the text "Nessun file selezionato", a "Pagina web" field, and a "gg/mm/aaaa" field. A checkbox "Accetto i termini e le condizioni di utilizzo" is checked, and a "Registrati" button is located below it. At the bottom of the form, a link "Hai già un account? Effettua il login" is visible. The background of the form area shows a hand holding a pen and a watch.

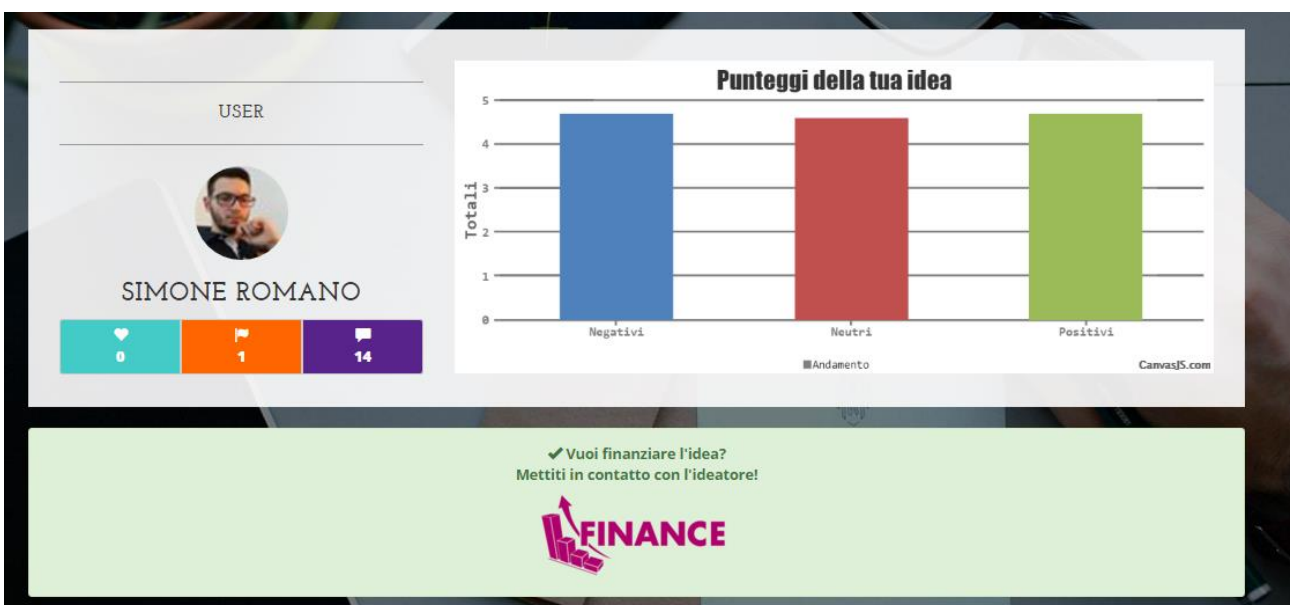
Nella Home, sarà possibile visualizzare tutte le idee inserite, ordinate per categoria. Sarà visibile la valutazione degli utenti e il numero di follower.

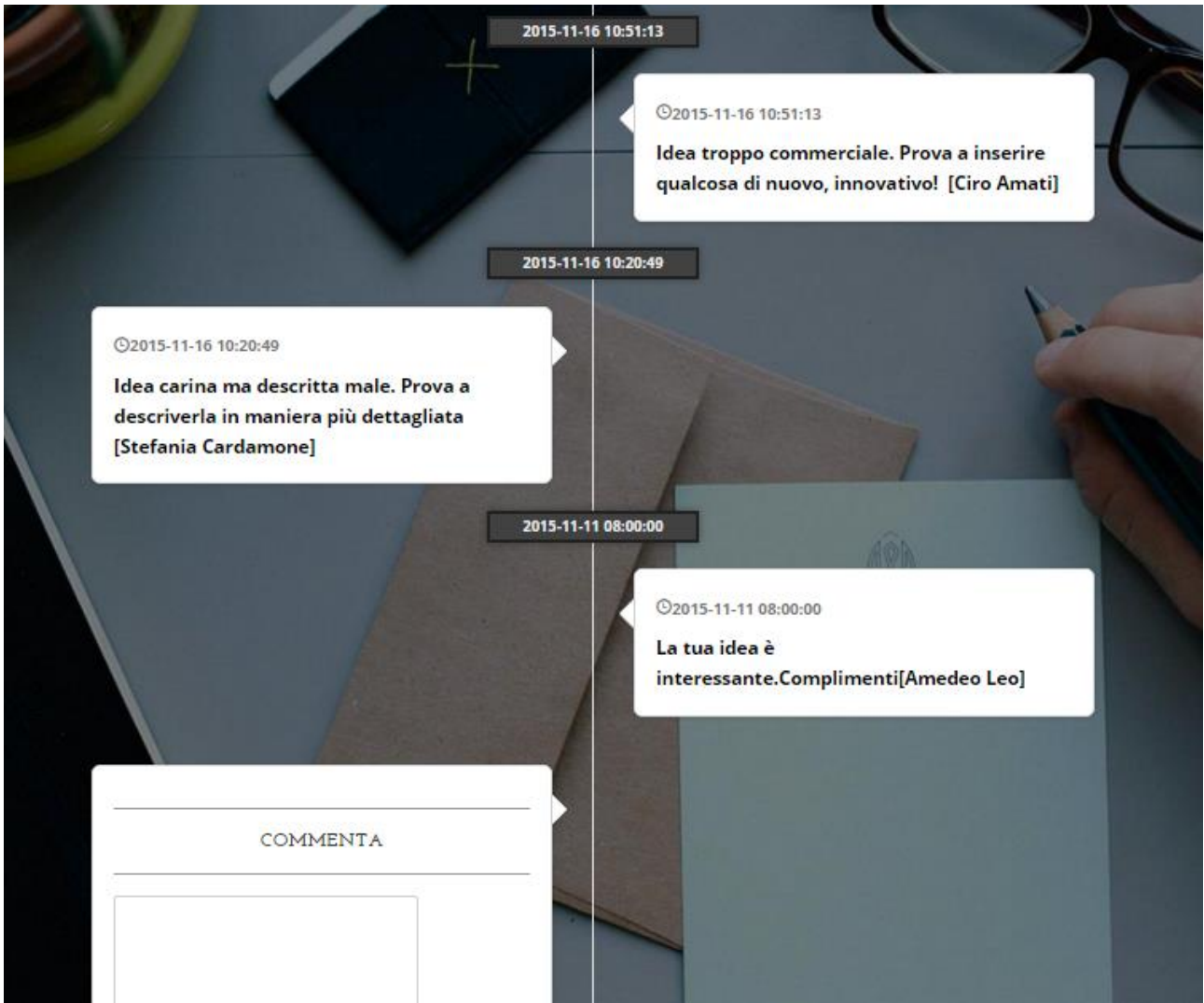


Vediamo in dettaglio la pagina dell'idea. Essa presenta una prima parte descrittiva dell'idea stessa. È possibile "seguirla" effettuando un click sull'apposito tasto "follow".



Grazie alla sentiment analysis è possibile visualizzare il punteggio dell'idea analizzando i commenti inseriti dagli utenti. Essi saranno classificati come "negativi", "neutri" e "positivi". Il portale, inoltre, permette ad un qualsiasi utente di finanziare l'idea, se la ritiene interessante.





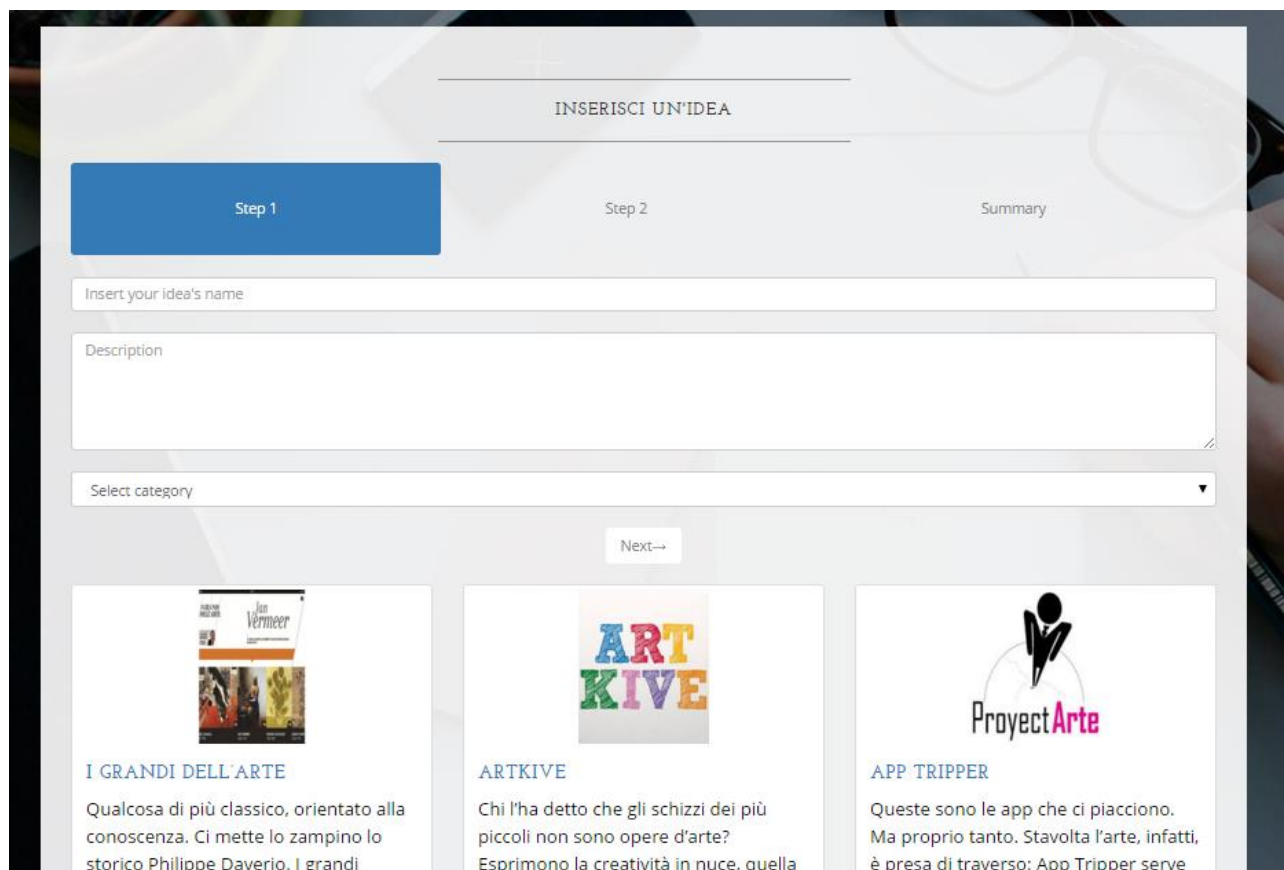
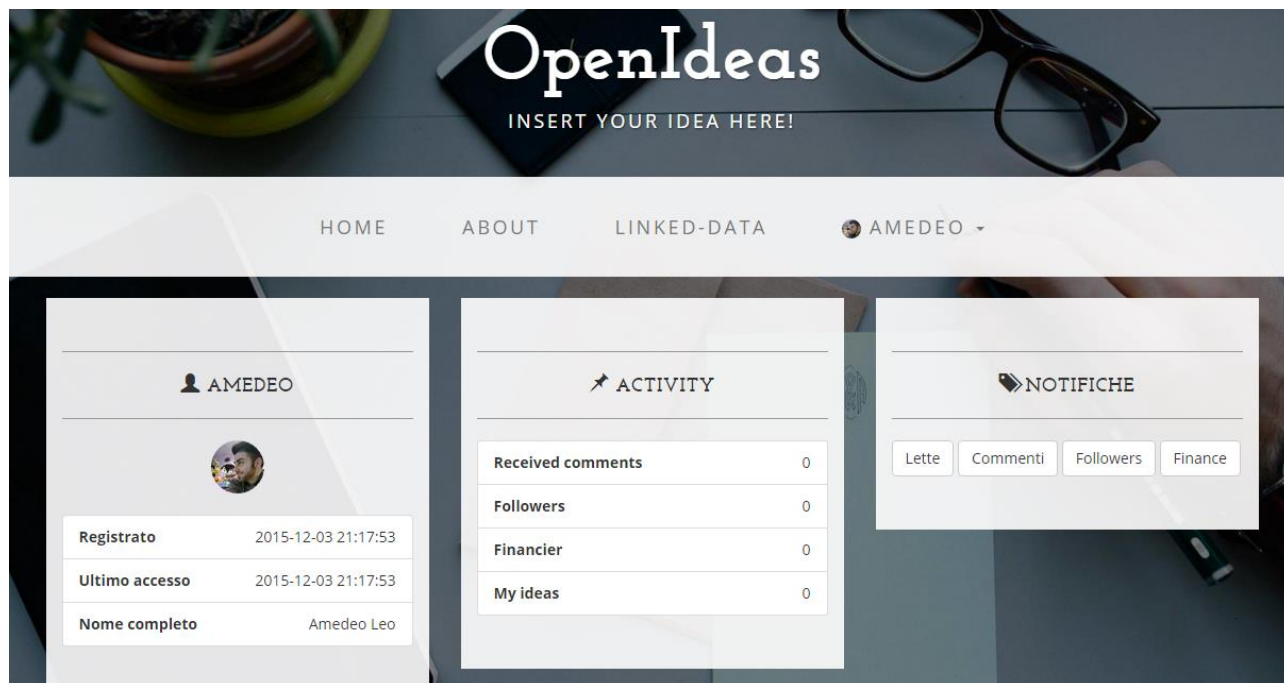
L'utente potrà esprimere una propria opinione commentando l'idea. I commenti, come detto in precedenza, saranno utili per la sentiment analysis.





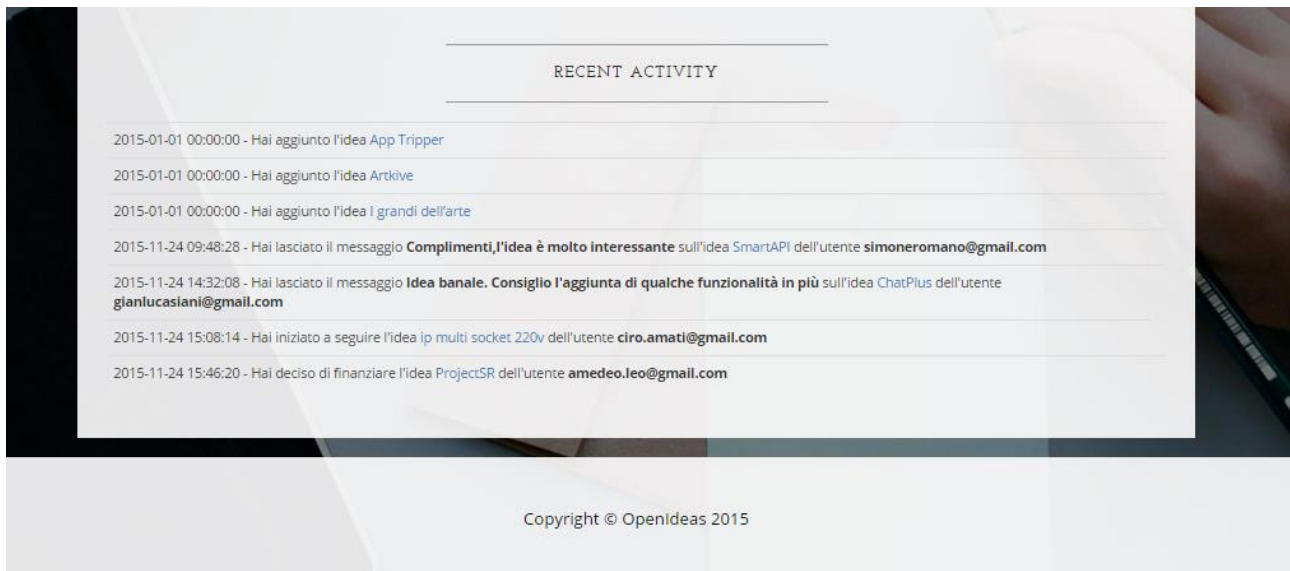
È prevista anche una sezione, all'interno della pagina Idea, in cui l'utente può visualizzare i dati relativi all'ultima settimana.

Profilo dell'utente: I dati relativi alle informazioni sull'idea dell'utente, possono essere visibili nella propria pagina. Prevede appunto una sezione per le notifiche e il conteggio delle attività.

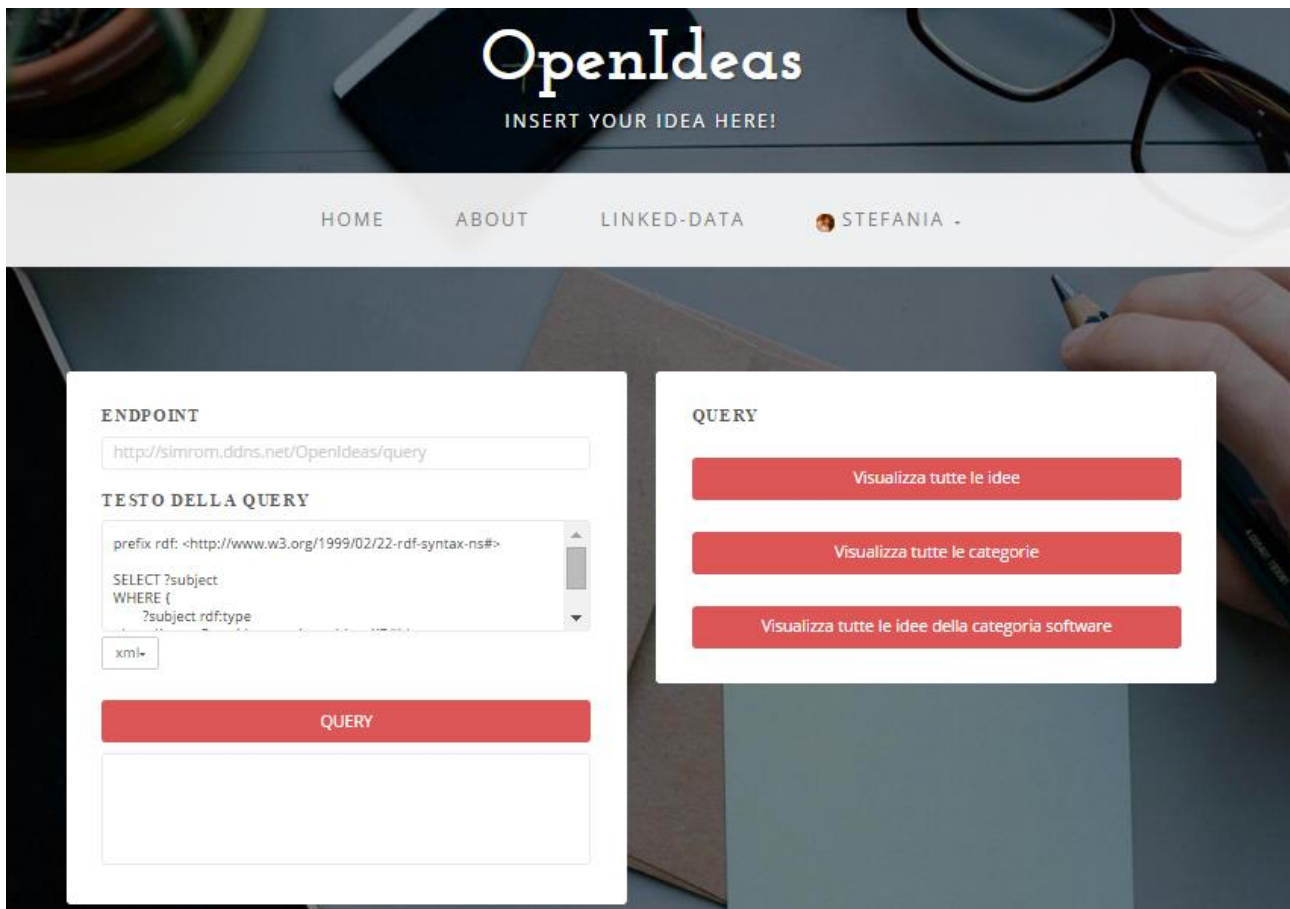




L'utente inserirà l'idea nella propria user page grazie ad una form dedicata. Visualizzerà, inoltre, l'history delle sue attività.



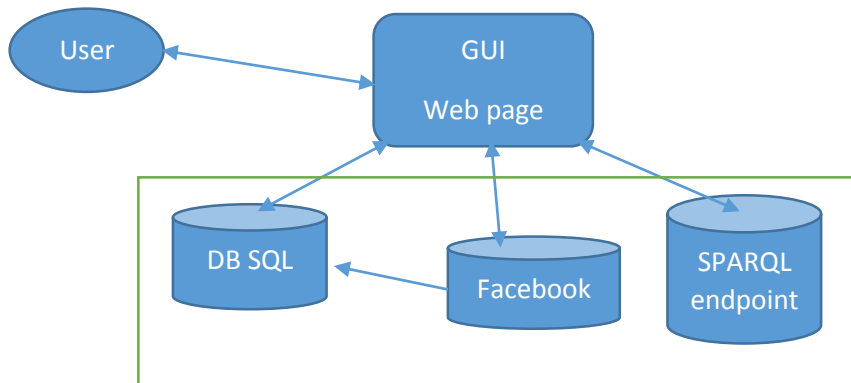
Il portale presenta anche una sezione “Linked data” come descritta nella sezione [Tecnologie utilizzate](#):



## Architettura e tecnologie

### Architettura del sistema

L'architettura del sistema è mostrata di seguito:



L'utente del sistema interagisce con una Web Application per visualizzare i contenuti o aggiungerne altri. L'applicazione utilizza più fonti per lo storage dei dati. In particolare è stato realizzato un database SQL per lo storage di tutte le informazioni (vedere sezione seguente). I **dati relativi alle informazioni degli utenti** possono essere popolati

- Dall'utente effettuando una registrazione al sito
- Con le informazioni prelevate dall'account Facebook dell'utente (se l'utente utilizza la funzionalità *login with facebook*)

Inoltre, i **dati relativi alle idee inserite** nel sistema sono memorizzati sia nel database SQL che nel triple store appositamente realizzato. Quindi, ogni volta che un utente inserisce una nuova idea nel sistema, questa viene inserita nel database SQL e viene caricata nel grafo del dataset pubblico realizzato (vedere sezione

Sentiment analysis algorithm

La libreria “phpInsight” (disponibile a <https://github.com/JWHennessey/phpInsight/tree/master/lib/PHPInsight> ) utilizza un **classificatore Naive Bayes** per calcolare il “sentimento” della frase in input. Il learning del classificatore è fatto utilizzando tre dataset di parole per le categorie *positivo*, *negativo* e *neutro*. Di seguito sono riportati i **dizionari**:

<b>Pos</b>	%-) (-: (: (^ (^-^ ) (^.^) (^_ ^) (o: *\o/* --^--@ 0:) 8) :) :-) :-* :-d :-p :-} :3 :9 :'de :] :b) :d :o) :p :x ;^ ) <3 <3 =) =] >:) >:d >=d @}->-- ^) ^_ ^ abundance abide abilities ability abound above-average absolve abundance abundant accede accept acceptable ...
<b>Neg</b>	%-( )-: )o: 38* 8 8-0 8/ 8c : :# : ( :*( : ( : - :& :- ( :-(o) :-/ :-s :-  / : [ : _ ( :e :f :o :o( :s :  = [ > >/ >:( >:  >:o >[ >o> ^o) abandon abandoned abandonment abase abasement abash abate abdicate aberration abhor ...
<b>Neu</b>	(o; *) 8-) :- :-\ :-o :0->-<[: :l :  ;) ;o) <:} absolute absolutely absorbed accentuate activist actual actuality adolescents affect affected air alert all-time allegorize alliance alliances allusion allusions alright altogether amplify analytical apparent apparently appearance apprehend assess assessment assessments assumption astronomic astronomical astronomically attitude attitudes average aware ...

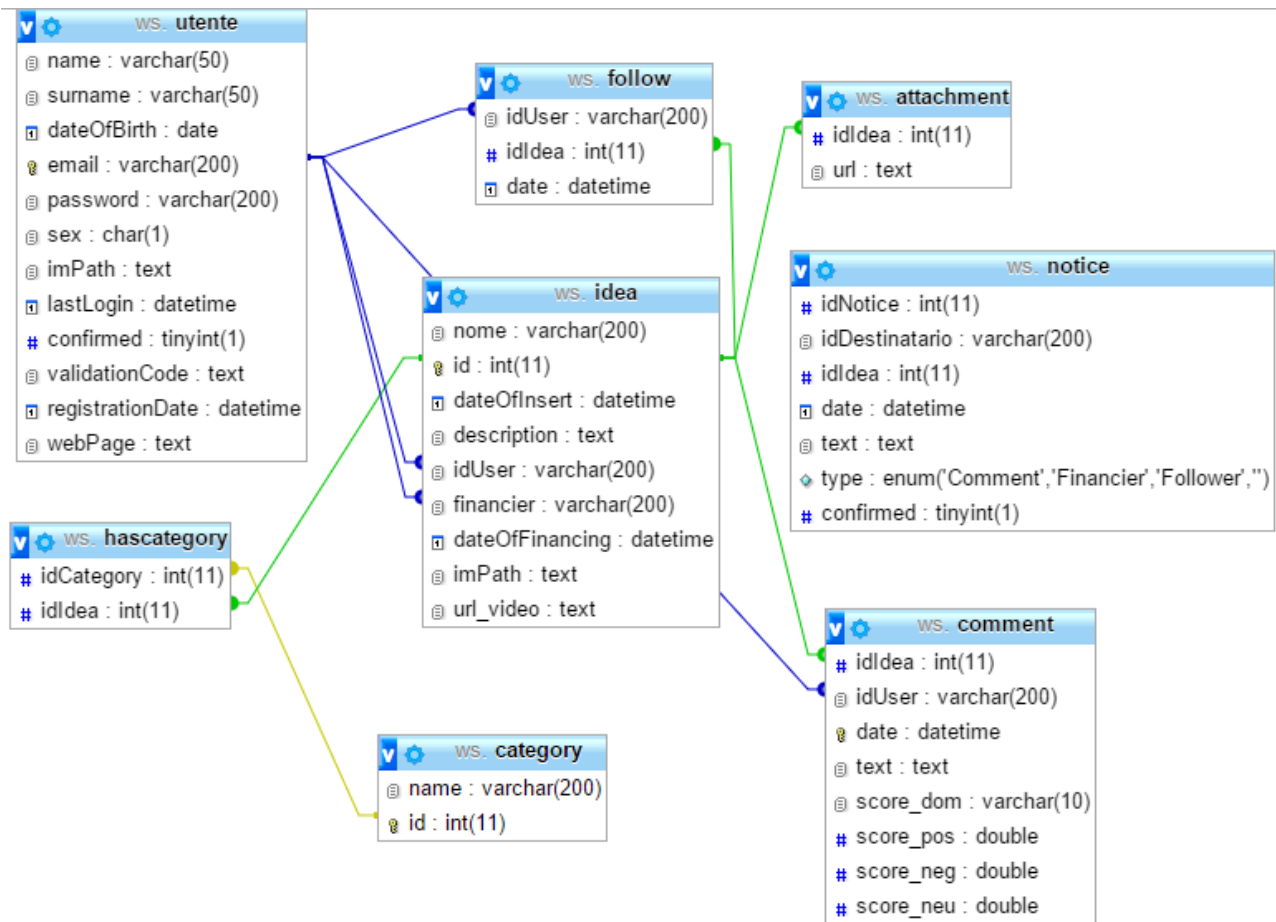
Creato il dataset, è stato addestrato il classificatore Naive Bayes. Il training di tale classificatore è molto veloce ma assume l’indipendenza tra le features.

Linked data). Il grafo RDF può essere interrogato tramite query SPARQL per ottenere le informazioni desiderate.

## Gestione dati

### Database Sql

La figura seguente mostra il diagramma ER della base di dati realizzata.



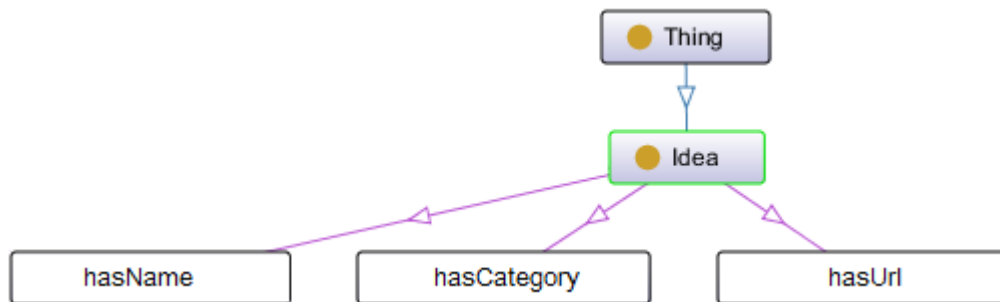
Di seguito è spiegata brevemente ciascuna tabella:

- **utente**: contiene le informazioni relative all'utente del sistema; tali informazioni potranno essere inserite dall'utente in fase di registrazione o potranno essere prelevate da facebook nel caso di login con facebook; la password dell'utente non è memorizzata in chiaro
- **follow**: tale tabella esprime la relazione *follow* presente tra un utente ed una idea; l'utente può *seguire* una idea per essere aggiornato sulle news (commenti, finanziatori)
- **attachment**: ogni idea avrà sicuramente una immagine ed eventualmente altri allegati; tale tabella lega gli allegati all'idea corrispondente
- **idea**: contiene le informazioni sull'idea
- **notice**: contiene le informazioni sulle notifiche per ogni utente; in particolare le tipologie di notifiche possono essere
  - commento su una idea inserita dall'utente o su una idea a cui l'utente è interessato
  - un utente ha deciso di finanziare una tua idea
  - un utente ha cominciato a seguire una tua idea
- **hascategory**: ogni idea appartiene ad una categoria; tale tabella lega una idea ad una categoria
- **category**: contiene le categorie
- **comment**: contiene le informazioni sui commenti inseriti dagli utenti e sull'output della *sentiment analysis* effettuata su di essi

*Triple-store*

I dati principali presenti nel sistema sono quelli relativi alle idee inserite dall'utente. Tali dati, come visto nella sezione precedente, sono già raccolti nella tabella *idea* del database SQL. Oggigiorno si sta notevolmente diffondendo il concetto di *Linked-data*, che propone la pubblicazione di dati strutturati e collegati fra loro. È per questo motivo che è stato realizzato un endpoint SPARQL che contiene i dati essenziali dell'idea. Tale dataset è pubblico ed interrogabile tramite la sezione **Linked-data** presente sull'applicazione web con linguaggio **SPARQL**.

Di seguito è riportato il grafo della semplice ontologia realizzata



Vediamo dunque il codice Turtle che realizza lo schema appena mostrato:

```

1 @prefix : <http://www.OpenIdeas.org/openIdeasKB#> .
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3 @prefix owl: <http://www.w3.org/2002/07/owl#> .
4 @prefix xml: <http://www.w3.org/XML/1998/namespace> .
5 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
6 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
7
8 <http://www.OpenIdeas.org/openIdeasKB>
9   a owl:Ontology .
10
11 :hasName a owl:DatatypeProperty ;
12         rdfs:domain :Idea ;
13         rdfs:range xsd:string .
14
15 :Idea a owl:Class .
16
17 :hasUrl a owl:DatatypeProperty ;
18        rdfs:domain :Idea ;
19        rdfs:range xsd:string .
20
21 :8 a :Idea , owl:NamedIndividual ;
22   :hasCategory "Software" ;
23   :hasName "idea13" ;
24   :hasUrl "http://localhost/WebSemantico/OpenIdeas/idea.php?id=13" .
25
26 :11 a :Idea .
27
28 :hasCategory a owl:DatatypeProperty ;
29            rdfs:domain :Idea ;
30            rdfs:range xsd:string .
31
32 :13 a :Idea , owl:NamedIndividual ;
33   :hasCategory "Software" ;
34   :hasName "idea13" ;
35   :hasUrl "http://localhost/WebSemantico/OpenIdeas/idea.php?id=13" .
  
```

## Tecnologie utilizzate

Come già evidenziato nelle sezioni precedenti, il prodotto descritto consiste in una Web Application. Per lo sviluppo dell'applicazione si è scelto di utilizzare le seguenti tecnologie/servizi:

- php
- html
- JavaScript
- jQuery
- SQL
- SPARQL
- Fuseki
- AJAX

Come già evidenziato precedentemente, i commenti inseriti dagli utenti per le varie idee sono utilizzati per abbinare un punteggio all'idea. Sono state utilizzate 2 librerie php per effettuare la sentiment analysis

1. [google-translate-php](#)
2. [phpInsight](#)

La seconda libreria permette, dato un testo in lingua inglese, di ottenere 3 punteggi relativi a

1. Positività
2. Neutralità
3. Negatività

del testo inserito. Dal momento che gli utenti possono essere di diversa nazionalità, il testo dei commenti viene prima tradotto con la prima libreria e poi dato in input alla seconda per la sentiment analysis. Vediamo dunque il frammento di codice che utilizza entrambe le librerie appena citate per realizzare la sentiment analysis:

```
$scores = getScore($content);  
insertComment($idUser, $idIdea, $content, $scores);
```

Quindi si calcola lo score del commento e si inseriscono i dati del commento nel database. In particolare vediamo la funzione realizzata per il calcolo dello score del commento:

```

function getScore($comment) {
    require __DIR__ . '/vendor/autoload.php';
    require_once __DIR__ . '/libs/sentimentAnalysis/autoload.php';

    //Translate first
    $str = new TranslateClient(); // Default is from 'auto' to 'en'
    //$str->setSource('it');
    //$str->setTarget('en');
    $comment = $str->translate($comment);

    //Calculate sentiment
    $sentiment = new \PHPInsight\Sentiment();
    // calculations:
    $scores = $sentiment->score($comment);
    $class = $sentiment->categorise($comment);

    $stoReturn['dom'] = $class;
    $stoReturn['pos'] = $scores['pos'];
    $stoReturn['neg'] = $scores['neg'];
    $stoReturn['neu'] = $scores['neu'];
    return $stoReturn;
}

```

È stato inoltre utilizzato l'**sdk di Facebook** per consentire agli utenti di effettuare il login utilizzando le credenziali facebook. In questo caso vengono recuperate le informazioni necessarie a riempire la tabella *utente* del database dall'account Facebook dell'utente e viene effettuata una registrazione (se l'email dell'utente non è già presente nel database) prima di effettuare il login. Il seguente codice JavaScript carica l'sdk Facebook:

```

window.fbAsyncInit = function() {
    FB.init({
        appId      : ██████████,
        cookie     : true, // enable cookies to allow the server to access
        // the session
        xfbml      : true, // parse social plugins on this page
        version    : 'v2.2' // use version 2.2
    });
}

```



La seguente funzione è invocata al click del bottone *Login with Facebook*:

---

```
// This function is called when someone finishes with the Login
// Button. See the onlogin handler attached to it in the sample
// code below.
function checkLoginState() {
  FB.login(function(response) {
    FB.getLoginStatus(function(response) {
      console.log('statusChangeCallback');
      console.log(response);
      // The response object is returned with a status field that lets the
      // app know the current login status of the person.
      // Full docs on the response object can be found in the documentation
      // for FB.getLoginStatus().
      if (response.status === 'connected') {
        // Logged into your app and Facebook.
        loggedInWithFacebook();
      } else if (response.status === 'not_authorized') {
        // The person is logged into Facebook, but not your app
      } else {
        // The person is not logged into Facebook, so we're not sure if
        // they are logged into this app or not.
      }
    });
  });
}
```

Altra importante funzionalità introdotta nell'applicazione è la sezione *Linked-data*. Tale sezione espone un'interfaccia per interrogare la base ontologica mostrata nel paragrafo precedente tramite linguaggio SPARQL. Per realizzare tale funzionalità è stato utilizzato [Fuseki](#), un web-server che consente di gestire dataset e grafi tramite protocollo http. Su tale web-server è stata caricata l'ontologia che viene interrogata tramite richieste http con la libreria curl. In particolare vengono inserite nuove idee ogni volta che l'utente ne aggiunge una dalla pagina personale e vengono interrogate quando l'utente utilizza l'interfaccia della sezione linked-data.

Il seguente codice effettua la richiesta POST per eseguire upload di una nuova idea nella ontologia presente sul server Fuseki:

```

/**
 * @author Simone Romano
 * This php script manage a fuseki sparql endpoint to store
 * new idea added in OpenIdeas portal.
 */
$endpoint = "http://simrom.ddns.net:3030/OpenIdeas/update?";
$query = "http://simrom.ddns.net:3030/OpenIdeas/query?";
function uploadIdeaInSparqlEndpoint($id, $url, $category, $name){
    if (!function_exists('curl_init'))
        die('CURL is not installed!');
    // get curl handle
    $ch= curl_init();
    $url = getUrl($id, $url, $category, $name);
    // set request url
    curl_setopt($ch, CURLOPT_URL, $url);
    // return response, don't print/echo
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_POST, 1);
    /*
    Here you find more options for curl:
    http://www.php.net/curl_setopt
    */
    $response = curl_exec($ch);
    print_r($response);
    curl_close($ch);
    return $response;
}

```

In particolare è interessante vedere la funzione `getUrl` che costruisce l'url per la richiesta POST:

```

function getUrl($id, $url, $category, $name){
    global $endpoint;
    $format = 'json';

    $query = "
    prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    prefix :<http://www.OpenIdeas.org/openIdeasKB#>
    INSERT DATA{
        :{$id} rdf:type :Idea;
        :hasUrl \"{$url}\";
        :hasName \"{$name}\";
        :hasCategory \"{$category}\".
    }";

    $searchUrl = $endpoint
        . 'update=' . urlencode($query)
        . '&format=' . $format;

    return $searchUrl;
}

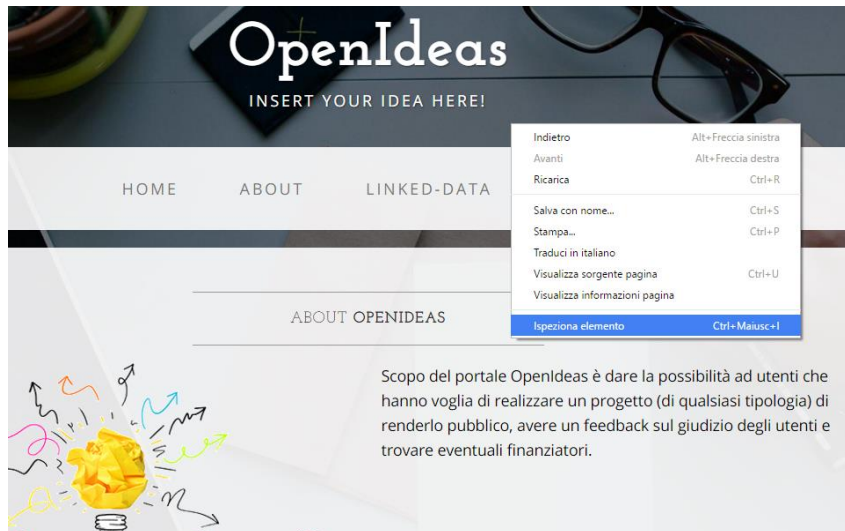
```

## Template Responsive

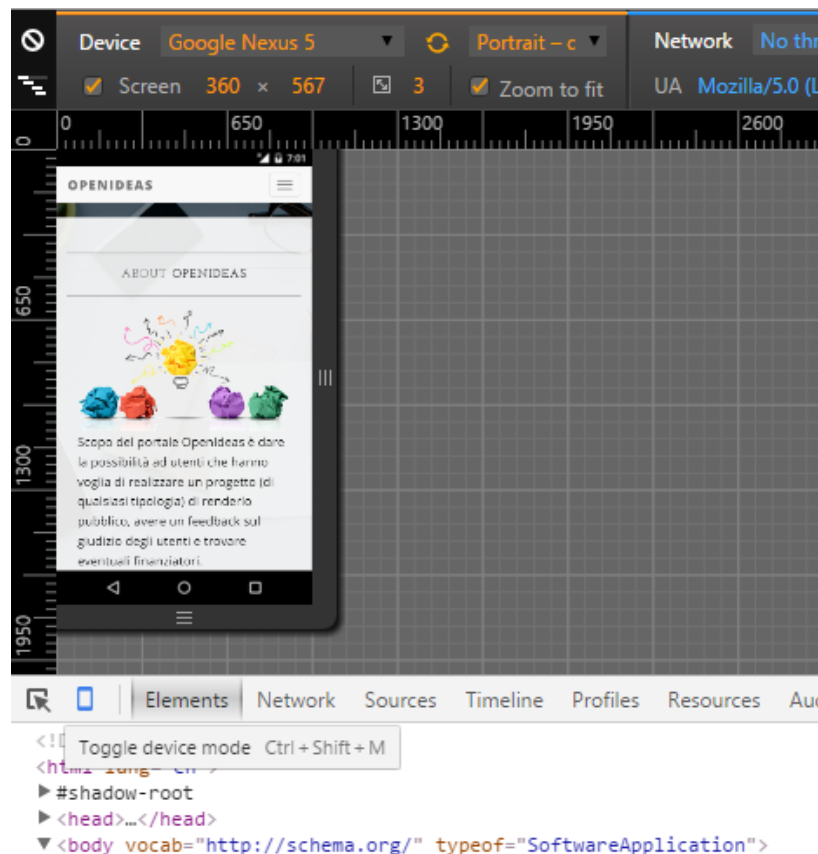
Il progetto inoltre, prevede un design così detto "responsive". Il design responsivo, o responsive web design (RWD), indica una tecnica di web design per la realizzazione di siti in grado di adattarsi graficamente in modo automatico al dispositivo coi quali vengono visualizzati (computer con diverse risoluzioni, tablet, smartphone, cellulari, web tv), riducendo al minimo la necessità dell'utente di ridimensionare e scorrere i contenuti.

Google, nel browser Chrome, permette grazie alla modalità "ispezione elemento", di poter visualizzare il template a seconda del device. Di seguito un semplice esempio nei quali saranno indicati i vari step:

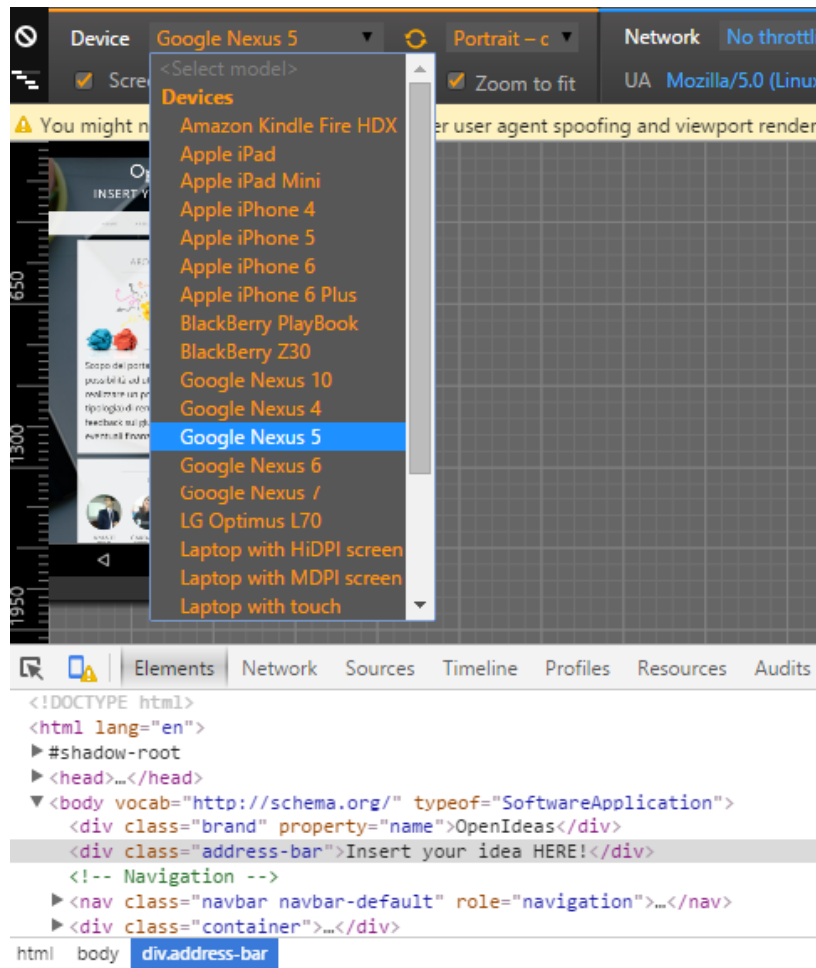
1. Con il tasto destro del mouse, è possibile accedere al seguente menu. Selezioniamo la voce "ispezione elemento"



2. Cliccando sull'icona del device sarà visualizzata la seguente interfaccia.



3. Infine, sarà possibile scegliere, attraverso un menu, il device sul quale visualizzare la pagina



## Semantic Web

Una parte delle tecnologie utilizzate nel progetto puntano ad utilizzare alcune delle tecnologie del Semantic web. In particolare sono stati analizzati i commenti inseriti dagli utenti con tecniche di **sentiment analysis**, è stata inserita una sezione **linked-data** che consente di interrogare un triple-store con informazioni sulle idee inserite tramite SPARQL ed è stato aggiunto codice RDFa alle pagine html.

## Sentiment analysis

Il portale realizzato rappresenta una sorta di *vetrina* dove un utente può inserire delle idee personali. Ci sono almeno 2 motivi per cui dovrebbe rendere pubblica una propria idea:

1. Trovare un finanziatore.
2. Avere un feedback da parte degli utenti sull'idea.

Capire qual è la valutazione degli utenti è un aspetto molto importante per l'utente che pubblica una propria idea. Esistono diversi modi per conoscere la valutazione dell'utente; un caso classico è quello di inserire una barra numerata (es. da 1 a 10) e chiedere esplicitamente all'utente di votare l'idea.

La cosa più naturale è invece far inserire un commento in linguaggio naturale all'utente ed elaborarlo al fine di ricavare uno score (o una serie di score) che danno una informazione sul contenuto sentimentale del commento. Raccolti questi score è dunque possibile ricavare un punteggio e dunque una valutazione dell'idea. Il paragrafo [Tecnologie utilizzate](#) spiega come è stata implementata tale funzionalità.

Ecco mostrato un semplice esempio con gli opportuni passi da seguire.

1. L'utente loggato visualizza un'idea.
2. Decide dunque di fornire un feedback positivo al suo autore.



The image shows a screenshot of a web interface for submitting a comment. At the top, the word "COMMENTA" is centered between two horizontal lines. Below this is a text input field with a blue border containing the text "Complimenti! Idea fantastica!". At the bottom of the form, the text "INSERISCI COMMENTO" is displayed in blue capital letters.

3. Il sistema assegna al commento dell'utente il seguente punteggio:

date	text	score_dom	score_neg	score_neu	score_pos
2015-11-30 09:08:09	Complimenti! Idea fantastica!	0	0.2	0.4	0.4

Tale punteggio è determinato:

- Dapprima traducendo il commento in inglese; ciò viene fatto con la libreria “google-translate-php” (disponibile a <https://github.com/Stichoza/google-translate-php>);
  - Successivamente, la libreria “phpInsight” (disponibile a <https://github.com/JWHennessey/phpInsight/tree/master/lib/PHPInsight> ) permette sia di assegnare al commento uno score dominante (in questo caso positivo) sia di determinare un punteggio ad ogni categoria (positivo, negativo, neutro).
4. Il sistema aggiorna quindi l'idea:
- Il grafico incrementerà ciascun valore a seconda del relativo score; in particolare sommerà ciascuno score di ogni categoria.
  - Nella sezione “Ultima settimana” saranno calcolati:
    - Media punteggi positivi ultima settimana;
    - Totale punteggi positivi ultima settimana;in base ai nuovi punteggi.

*Sentiment analysis algorithm*

La libreria “phpInsight” (disponibile a <https://github.com/JWHennessey/phpinsight/tree/master/lib/PHPInsight> ) utilizza un **classificatore Naive Bayes** per calcolare il “sentimento” della frase in input. Il learning del classificatore è fatto utilizzando tre dataset di parole per le categorie *positivo*, *negativo* e *neutro*. Di seguito sono riportati i **dizionari**:

<b>Pos</b>	%-) (-: (: (^ (^-^ ) (^.^) (^_ ^) (o: *\o/* --^--@ 0:) 8) :) :-) :-* :-d :-p :-} :3 :9 :de :] :b) :d :o) :p :x ;^ ) <3 <3 =) =] >:) >:d >=d @}->-- ^) ^_ ^ abidance abide abilities ability abound above-average absolve abundance abundant accede accept acceptable ...
<b>Neg</b>	%-( )-: ) :o: 38* 8 8-0 8/ 8c : :# :(: (* (: (- :-& :-(- (o) :-/ :-s :-  :/ :[ :_ ( :e :f :o :o( :s :  =[ > >/ >:( >:  >:o >[ >o> ^o) abandon abandoned abandonment abase abasement abash abate abdicate aberration abhor ...
<b>Neu</b>	(o; *) 8-) :- :-\ :-o :0->-<  :     ; )o) <:} absolute absolutely absorbed accentuate activist actual actuality adolescents affect affected air alert all-time allegorize alliance alliances allusion allusions alright altogether amplify analytical apparent apparently appearance apprehend assess assessment assessments assumption astronomic astronomical astronomically attitude attitudes average aware ...

Creato il dataset, è stato addestrato il classificatore Naive Bayes. Il training di tale classificatore è molto veloce ma assume l’indipendenza tra le features.

Linked data

Il concetto di Linked-data (dati collegati) è un concetto che si sta diffondendo molto sul web ed è attualmente una importante tematica di ricerca. I linked data rappresentano una modalità di pubblicazione di dati strutturati e collegati fra loro.

Nel caso specifico è stata creata una sezione del sito (Linked-data) che espone una interfaccia che consente di effettuare delle query in linguaggio SPARQL per ottenere informazioni sulle idee presenti nel database. Si può richiedere l’output in formato JSON o XML. Per i dettagli implementativi vedi il paragrafo [Tecnologie utilizzate](#).

RDFa

Le pagine Web scritte in XHTML contengono dati intrinsecamente strutturati: eventi di calendario, informazioni anagrafiche, didascalie di foto, titoli di canzoni, informazioni sul copyright, etc. Se gli autori delle pagine sono in grado di esprimere questi dati in maniera precisa, e i tool possono leggerli con affidabilità, si possono immaginare scenari di utilizzo del tutto nuovi. I dati strutturati possono essere trasferiti dal web alle applicazioni, in modo che, per esempio, i dati di calendario, o i diritti della licenza, possono essere comunicati

direttamente all'utente, per essere registrati nella sua agenda personale o per informarlo dei suoi diritti. I metadati possono essere inclusi nelle pagine HTML in vari modi (nel nostro caso, tramite RDFa).

In OpenIdeas utilizziamo due differenti approcci:

Il primo permetterà agli utenti di poter accedere alle informazioni nelle varie "property", senza dover adottare tecniche di tipo "scraping".

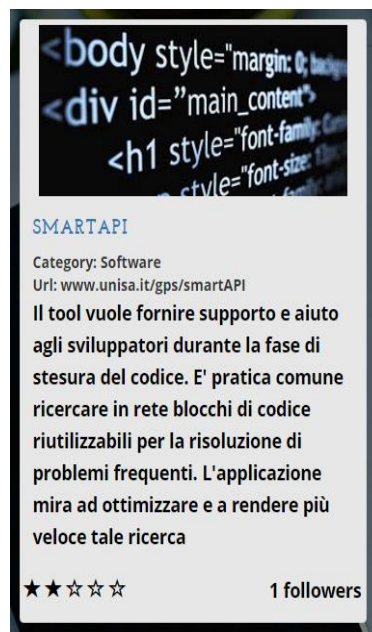
Come è possibile vedere dall'immagine, il prefix = "oi: http://www.OpenIdeas.org/openIdeasKB" specifica il vocabolario OpenIdeas. Questo vocabolario è utilizzato negli attributi "property" per specificare le proprietà del documento o gli elementi contenuti nel documento. Molti di questi vocabolari possono essere inclusi in un unico documento.

```
index.php
1 <!DOCTYPE html>
2 <html lang="en" xmlns="http://www.w3.org/1999/xhtml"
3   prefix="oi: http://www.OpenIdeas.org/openIdeasKB">
4 <head>
```

Grazie all'attributo "property" di RDFa, è possibile specificare: il nome del progetto, la categoria e l'url in cui risiede la risorsa.

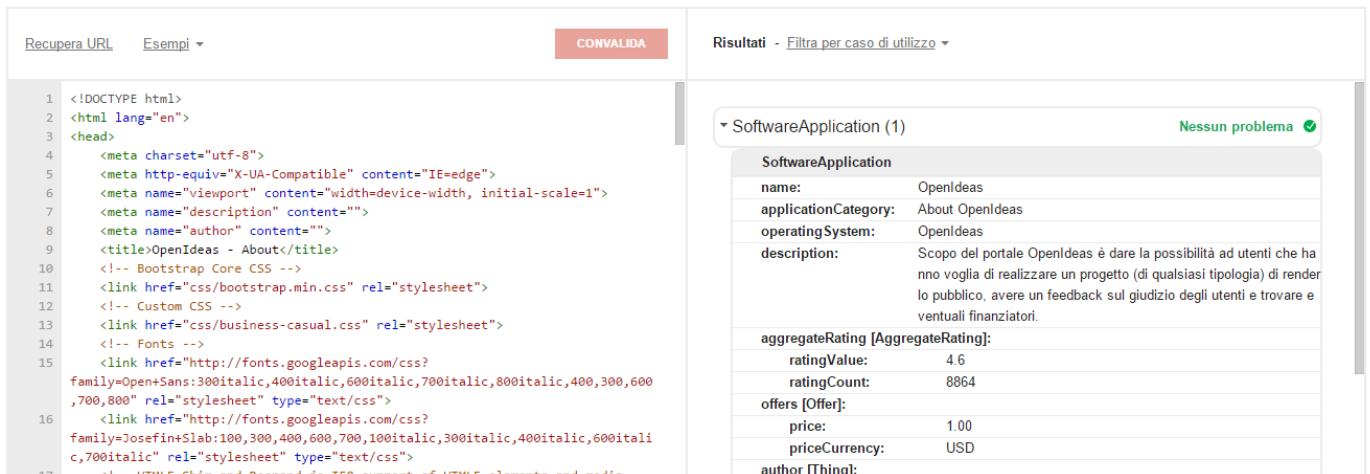
```
<div class="col-sm-4 col-lg-4 col-md-4">
  <div class="thumbnail" style="background-color: #e7e7e7;font-weight: 700; ">
    " style="width:320px;height:150px;" >
    <div class="caption">
      <h4 class="pull-right"></h4>
      <h4><a property="oi:hasName" href="<?php echo "idea.php?id={$idIdea};">"><?php echo "{$idea['Idea']['nome']}"></a>
      </h4>
      <span property="oi:hasCategory"><?php echo "Category: {$category}" ?> </span><br>
      <span property="oi:hasUrl"><?php echo "Url: {$url}" ?> </span>
      <p><?php echo "{$idea['Idea']['description']}"></p>
    </div>
```

L'output finale sarà il seguente:





Il secondo approccio è stato applicato per la sezione “About”, in cui è utilizzato il vocabolario standard `vocab=http://schema.org/` specificando il tipo di Rich Snippet `typeof="SoftwareApplication"`. In questo modo le informazioni saranno prelevate direttamente dal motore di ricerca e visualizzate nella pagina dei risultati. Grazie a un tool di Google, può essere testato il codice HTML.



The screenshot shows a tool interface for testing HTML code. On the left, there is a text area with HTML code for an 'About' page, including meta tags for charset, viewport, description, and author, as well as links for Bootstrap Core CSS and custom CSS. On the right, the tool displays the results of a structured data test for a 'SoftwareApplication' schema. The results show a single instance with the following details:

SoftwareApplication	
name:	OpenIdeas
applicationCategory:	About OpenIdeas
operatingSystem:	OpenIdeas
description:	Scopo del portale OpenIdeas è dare la possibilità ad utenti che hanno voglia di realizzare un progetto (di qualsiasi tipologia) di renderlo pubblico, avere un feedback sul giudizio degli utenti e trovare eventuali finanziatori.
aggregateRating [AggregateRating]:	
ratingValue:	4.6
ratingCount:	8864
offers [Offer]:	
price:	1.00
priceCurrency:	USD
author [Thing]:	

## Sicurezza

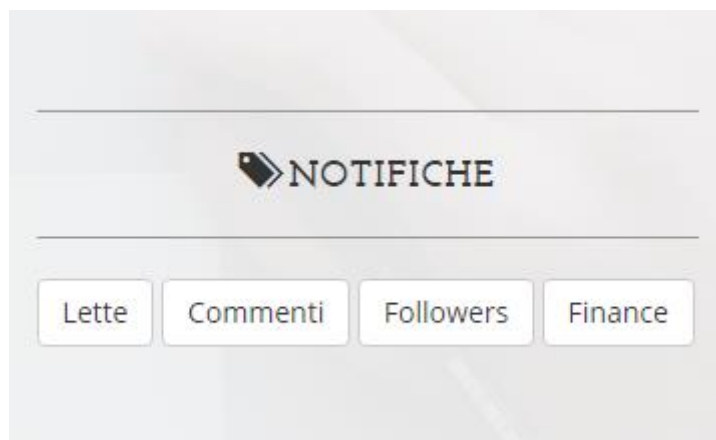
Il sistema dispone di un sistema di codifica per inserire la password all'interno del database. Qualora l'utente si registrasse tramite le API di Facebook, la password non verrà memorizzata; ciò è dovuto alla safety definita dalle stesse API. Al login dell'utente verrà automaticamente decodificata.

## Email e notifiche

Il sistema permette all'utente di essere in continuo aggiornamento sui cambiamenti delle idee; in particolare:

- Se si ha commentato, si tiene traccia di commenti successivi al proprio commento.
- Se si ha scelto di seguire un'idea, si tiene traccia di nuovi commenti e finanziatore.
- Se si ha inserito un'idea, si tiene traccia di commenti, finanziatore e followers.

Ciò viene fatto sia tramite l'invio di un'email all'utente sia attraverso le notifiche del sistema.



In particolare, tramite “Lette” si eliminano tutte le notifiche del sistema. Invece, se, ad esempio, si clicca su “Commenti” saranno mostrate tutte le notifiche dell'utente relative ai commenti. Al click di ogni notifica, il sistema porterà l'utente sulla relativa idea che ha generato quella notifica.

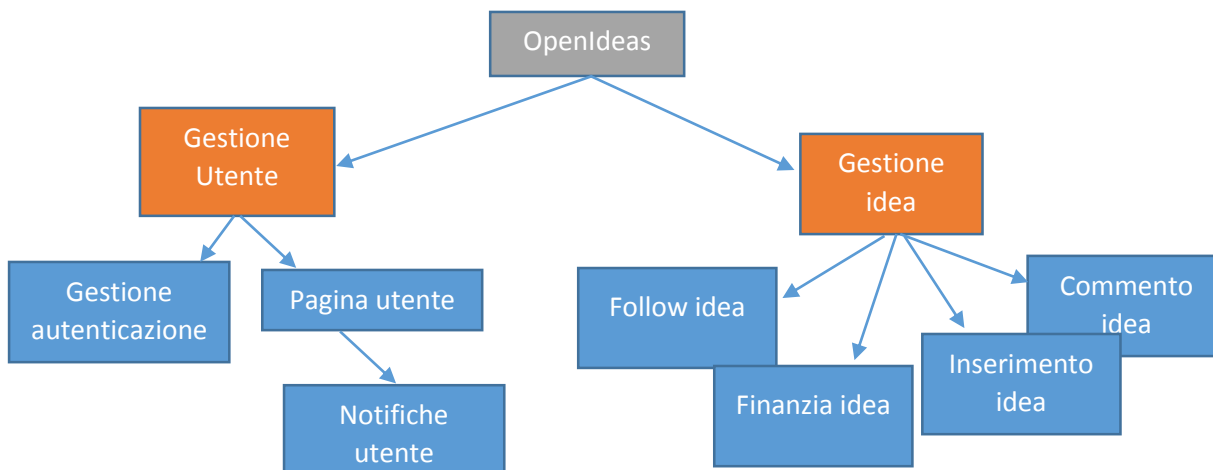
Inoltre, all'atto della registrazione, all'utente viene inviata una email per conferma autenticazione.

## Workflow

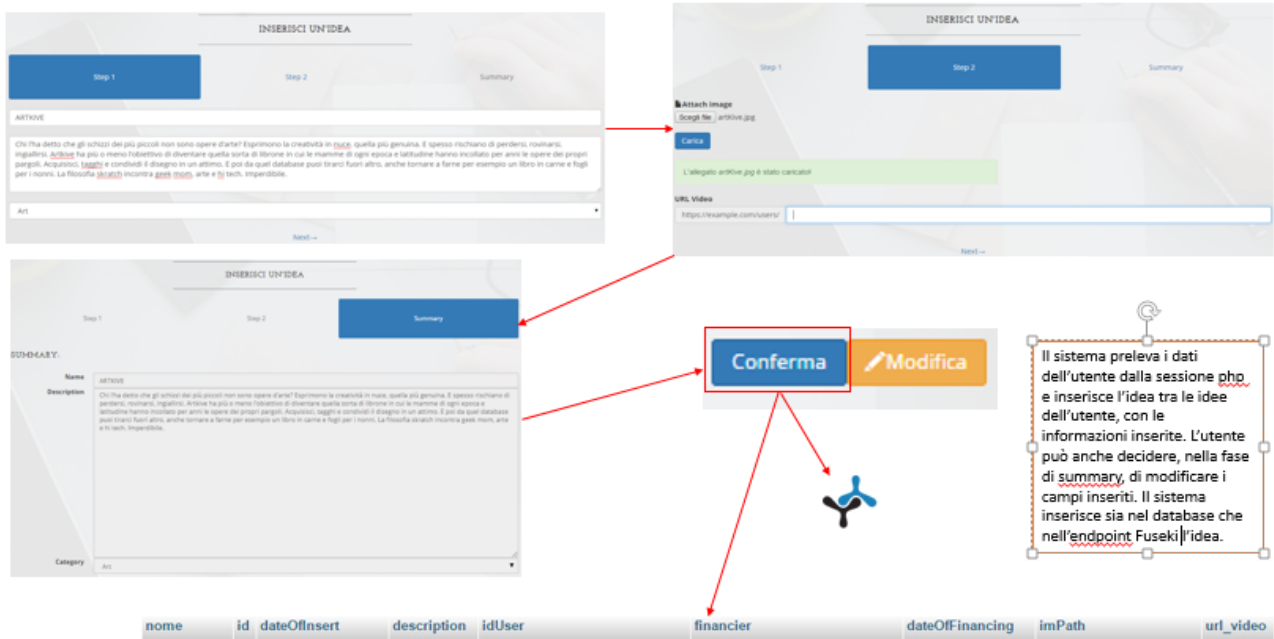
Descriviamo in questa sezione il workflow di alcune delle principali feature del sistema. In particolare andremo a dettagliare le funzionalità legate alla gestione dell'autenticazione (fase di registrazione, login, recupero password) e alle caratteristiche principali della gestione dell'idea.

## Visione globale

Si può scomporre il sistema OpenIdeas in due macrocomponenti, una legata alla gestione degli utenti e l'altra legata alla gestione dell'idea.



## Inserimento idea

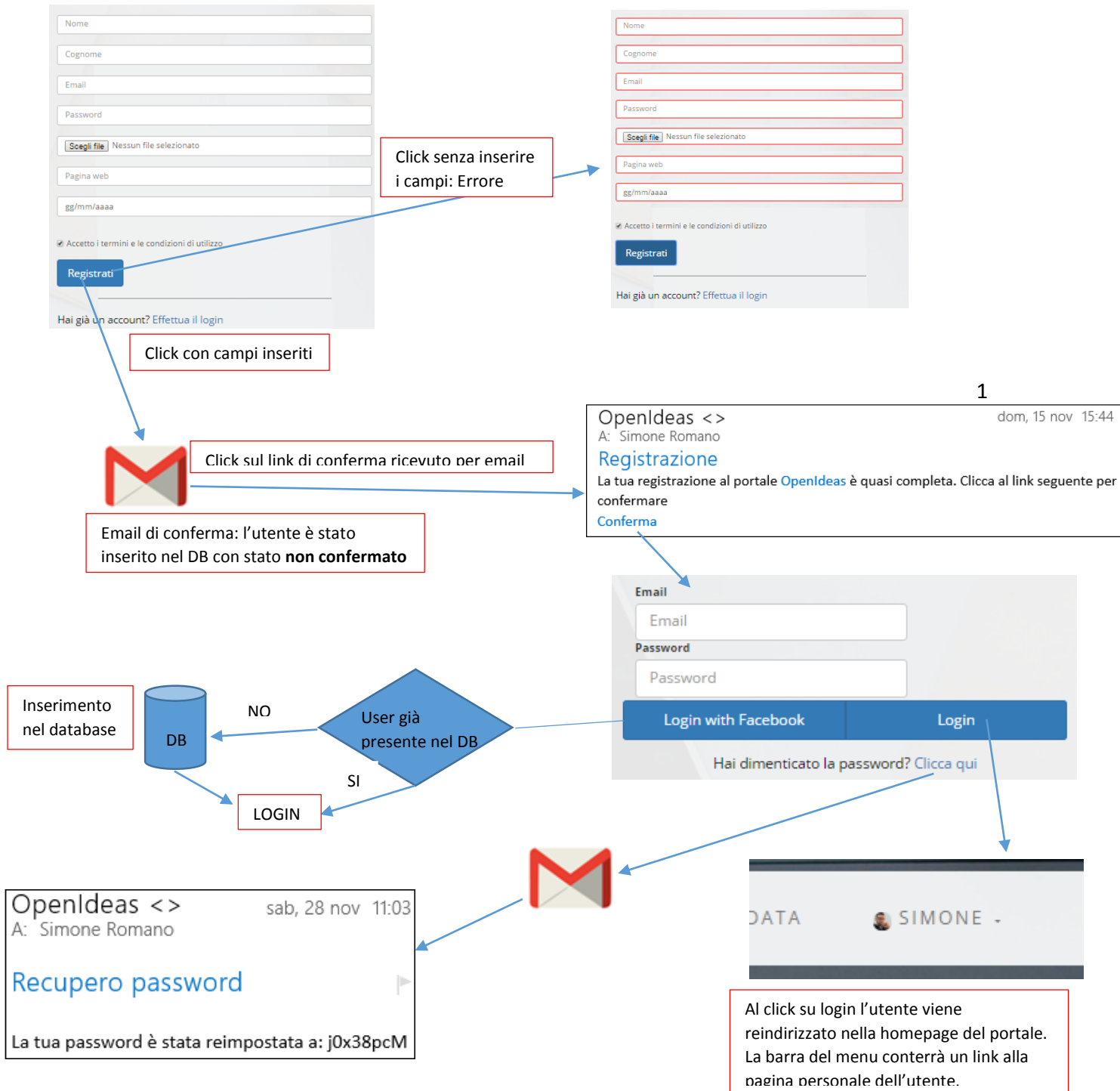


### Note:

- Solo gli utenti registrati possono inserire un'idea.
- É possibile modificare i campi inseriti.
- Il sistema non permette di continuare con l'inserimento se non sono stati compilati tutti i campi.

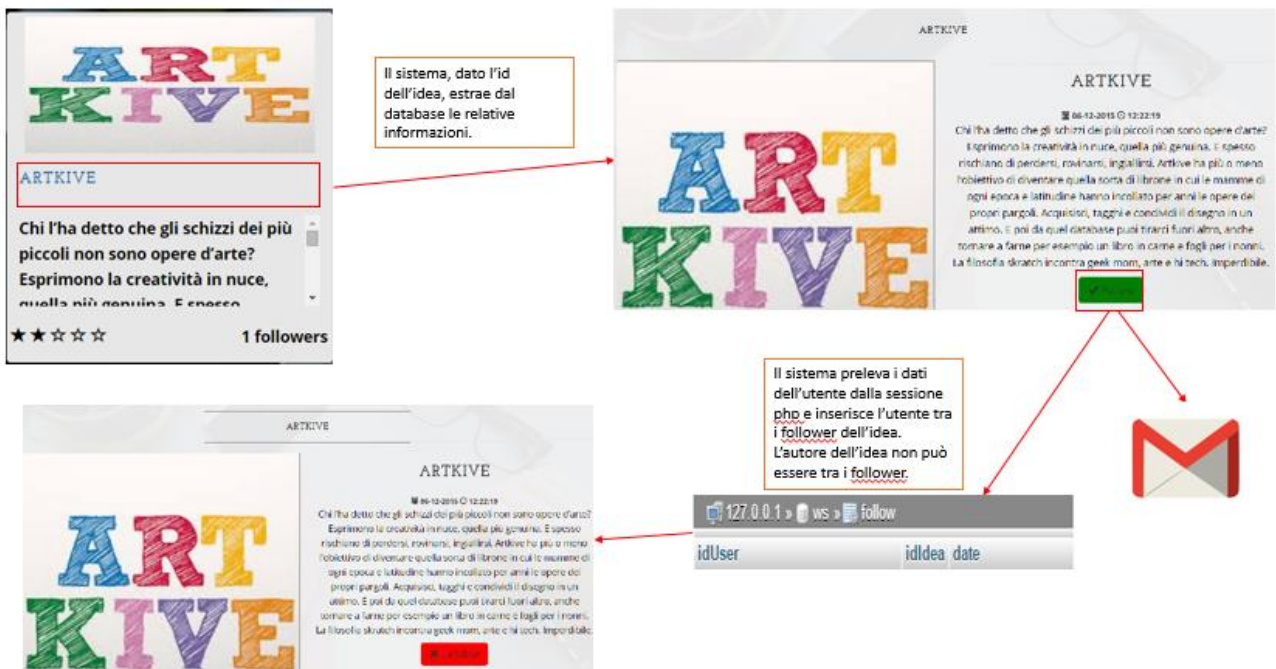
## Gestione autenticazione

Diamo dunque una visione ad alto livello della gestione dell'autenticazione degli utenti al sistema.



<sup>1</sup> Quando l'utente clicca su *Registrati* le sue informazioni vengono memorizzate nel DB in maniera temporanea e viene salvato un campo *ConfirmationCode* calcolato come md5 della stringa data dalla concatenazione dell'email e del nome; quando l'utente clicca su *Conferma* (nella email) viene creata una richiesta GET avente come parametri email e nome; il sistema può dunque verificare la identità dell'utente.

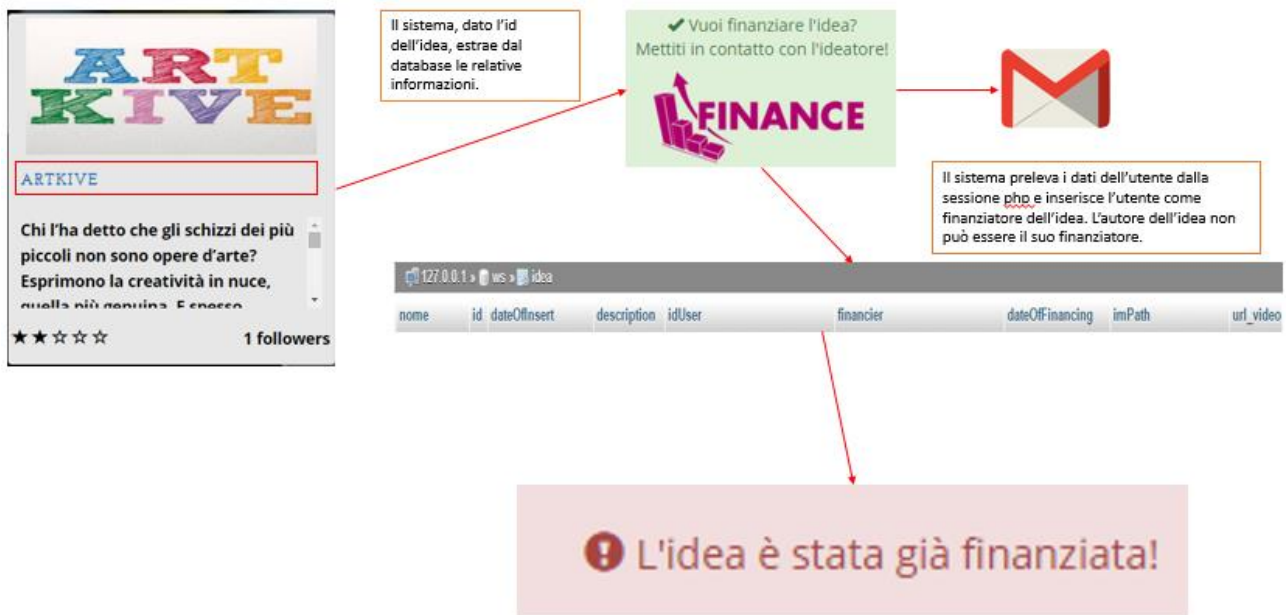
## Follow idea



### Note:

- Il sistema controlla che l'utente non sia già tra i follower dell'idea.
- Il sistema prevede la possibilità di fare "unfollow" dell'idea se si seguiva in precedenza.
- L'aggiornamento del button viene effettuato tramite uno script ajax, che invoca una funzione php per aggiornare il database.
- Il sistema invia sia una notifica che un'email all'autore dell'idea per informarlo del nuovo follower.

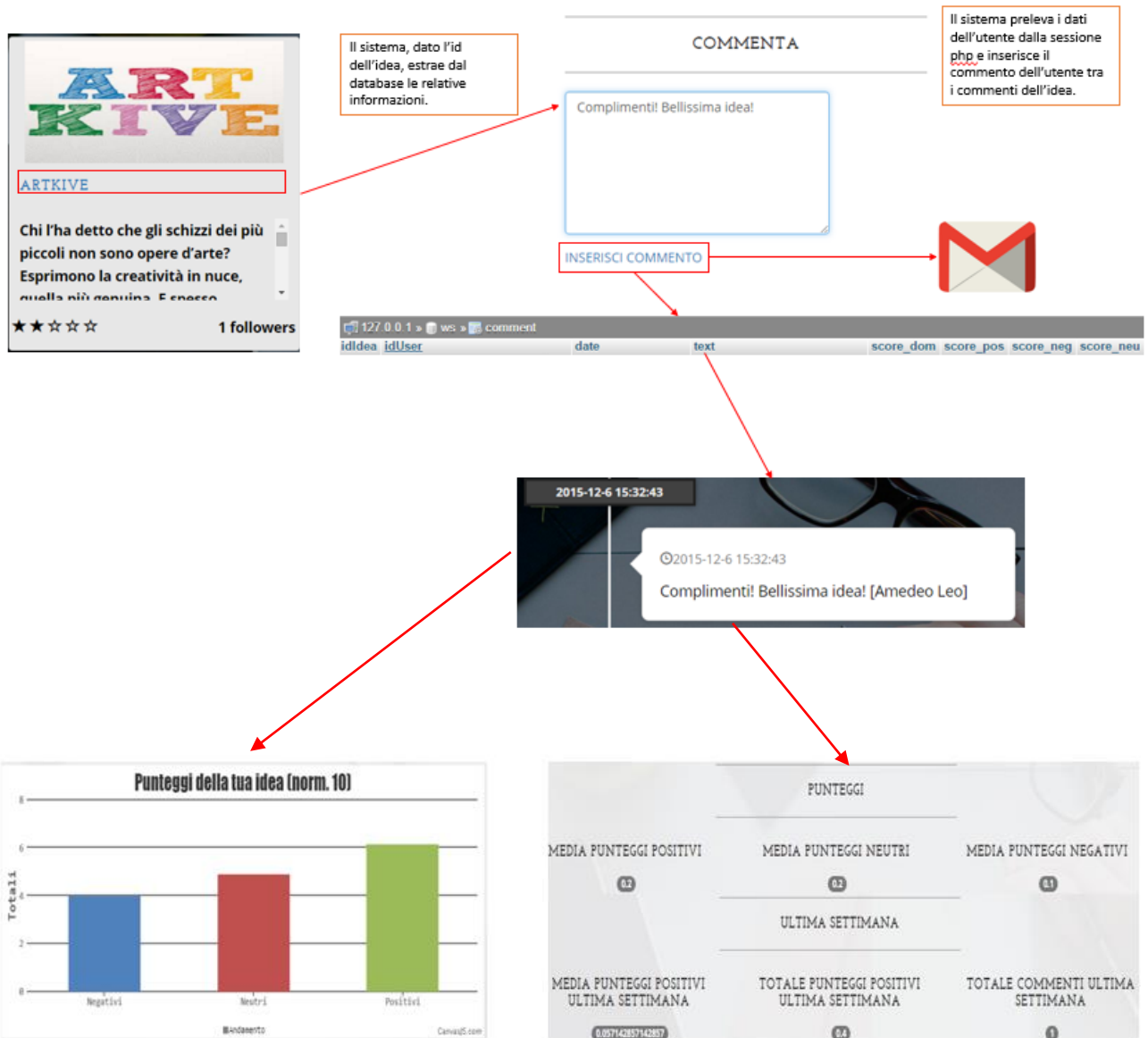
## Finanziamento idea



### Note:

- Il sistema controlla che non ci sia già un finanziatore per l'idea.
- L'aggiornamento del button viene effettuato tramite uno script ajax, che invoca una funzione php per aggiornare il database.
- Il sistema invia sia una notifica che un'email:
  - all'autore dell'idea per informarlo del finanziamento;
  - ai follower dell'idea;
  - al finanziatore.

## Commento idea



### Note:

- L'inserimento del commento viene effettuato tramite uno script ajax, che invoca una funzione php per aggiornare il database.
- Il sistema porta l'utente verso l'ultimo comment inserito. L'ordine è cronologico inverso.
- Il sistema invia sia una notifica che un'email:
  - all'autore dell'idea per informarlo del commento;
  - ai follower dell'idea;
  - agli utenti che avevano precedentemente commentato l'idea.
- Il sistema inferisce dal commento i punteggi relativi alla sentiment analysis e li inserisce nella tabella "comment" del database. Quindi aggiorna sia il grafico dell'idea, in cui i punteggi sono stati normalizzati a 10, sia la sezione relativa ai punteggi, suddivisa in media di ciascuna classe e ultima settimana.

## Conclusioni

Il sistema proposto raggiunge gli obiettivi fissati in fase di analisi dei requisiti. In particolare consente all'utente di inserire delle idee una volta effettuata la registrazione al portale e consente di visualizzare/commentare/finanziare altre idee presenti sul portale.

Abbiamo evidenziato che il sistema è capace di classificare le idee inserite tramite tecniche di sentiment analysis. La libreria utilizzata prende in input testo in lingua inglese. Per garantire il funzionamento del sistema di classificazione con tutti i linguaggi è stata introdotta una fase di traduzione (da qualsiasi linguaggio all'inglese) che è seguita dal calcolo dello score. Questo passaggio potrebbe compromettere la qualità dei risultati della sentiment analysis. Un possibile miglioramento può essere quello di utilizzare diversi dizionari per coprire più linguaggi.

Una delle possibili estensioni è quella di consentire agli utenti di interagire tra loro utilizzando il portale (es. tramite chat private).

La sezione linked-data contiene una innovativa interfaccia verso alcuni dei dati e consente di ottenere alcune importanti informazioni sulle idee presenti nel portale. Tali informazioni sono strutturate secondo una semplice ontologia creata ad-hoc per il contesto applicativo. Sviluppi futuri prevedono la estensione di tale knowledge-base per consentire all'utente di ricavare ulteriori informazioni. Eventualmente sarebbe possibile introdurre alcune relazioni che condurrebbero a sistemi di inferenza.



## References

Amati Ciro, C. S. (2015).

[https://drive.google.com/folderview?id=0ByKArlYxafLqR2N4RU5QSIUtdUE&usp=drive\\_web&tid=0ByKArlYxafLqDI1WkFENGE4cmc](https://drive.google.com/folderview?id=0ByKArlYxafLqR2N4RU5QSIUtdUE&usp=drive_web&tid=0ByKArlYxafLqDI1WkFENGE4cmc).

bootstrap. (2015). <http://getbootstrap.com/>.

Fuseki v2. (2015). [https://jena.apache.org/documentation/serving\\_data/](https://jena.apache.org/documentation/serving_data/).

google-translate. (2015). <https://github.com/Stichoza/google-translate-php>.

phpInsight. (2015). <https://github.com/JWHennessey/phpInsight>.

Rdfa. (2015). <http://www.w3.org/TR/xhtml-rdfa-primer/>.

Romano, S., & Leo, A. (2015). Fuseki: serving RDF data over HTTP.

[https://drive.google.com/folderview?id=0ByKArlYxafLqR2N4RU5QSIUtdUE&usp=drive\\_web&tid=0ByKArlYxafLqDI1WkFENGE4cmc](https://drive.google.com/folderview?id=0ByKArlYxafLqR2N4RU5QSIUtdUE&usp=drive_web&tid=0ByKArlYxafLqDI1WkFENGE4cmc).

Schema.org. (2015). <https://developers.google.com/structured-data/schema-org>.