

# Progettazione e sviluppo di piattaforma per la gestione di risorse linguistiche

Facoltà di Ingegneria dell'informazione, informatica e statistica

Corso di Laurea in Informatica

Andrea Gasparini

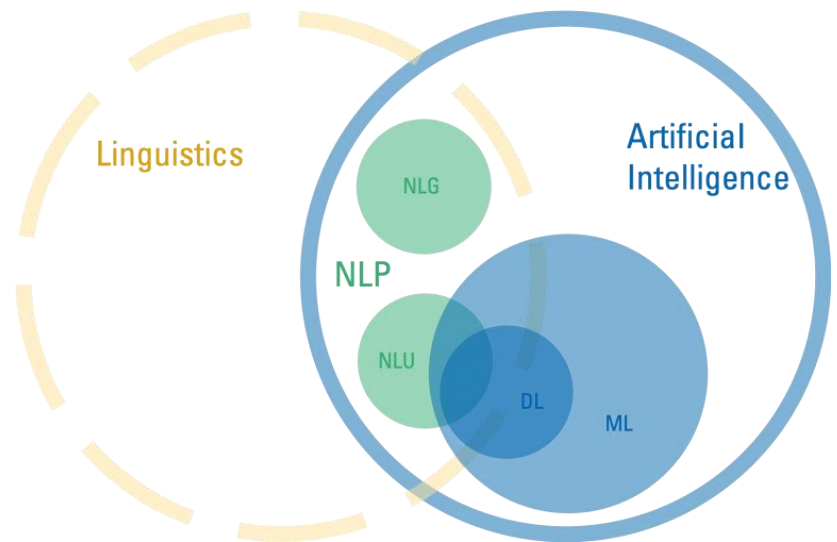


SAPIENZA  
UNIVERSITÀ DI ROMA

# Scenario di riferimento

# Scenario di riferimento: le risorse linguistiche

- **Natural Language Understanding (NLU):**  
argomento specifico dell'NLP relativo alla capacità delle macchine di leggere e comprendere il significato del linguaggio naturale
- Ambiguità del linguaggio naturale:  
è importante tenere in considerazione struttura sintattica e analisi semantica



# Scenario di riferimento: le risorse linguistiche

- Per elaborare un testo basandosi su sintassi e semantica, un algoritmo di NLP deve poter apprendere a partire da dati linguistici strutturati
- Una risorsa contenente questa tipologia di dati e che ne consente trattamento e analisi da parte di una macchina è detta **risorsa linguistica**
- **WordNet**, costruito manualmente, si basa sull'idea di un dizionario in cui la ricerca avviene concettualmente, raggruppando le parole in gruppi di sinonimi detti *synset*
- **BabelNet** è una rete semantica multilingue che integra automaticamente altre risorse come Wikipedia e WordNet, collegando lemmi e concetti tramite relazioni semantiche e raggruppando le parole in *Babel synsets*

# Scenario di riferimento: le licenze software

- Licenze closed source, *copyright* tradizionale, solitamente consentono il solo utilizzo e sotto alcune condizioni
- Licenze **Free and Open Source**, introducono il concetto di *copyleft*, concedendo utilizzo, copia, modifica e ridistribuzione, solitamente vincolando il solo appropriamento
- Licenze **Creative Commons**, permettono di concedere solamente alcune autorizzazioni e riservarne altre, creando la possibilità di avere un “copyright flessibile”



# Scenario di riferimento: necessità della piattaforma

- Molteplici modalità di accesso e di approvazione per le risorse di **SapienzaNLP**:
  - Non sempre viene verificato che l'utente approvi esplicitamente le licenze d'uso
  - L'utente può trovarsi a dover gestire più credenziali di accesso per diverse risorse
  - Complessità nella gestione delle richieste di accesso effettuate e della relativa approvazione
- Lo scopo è di unificare in un unico portale web:
  - Richiesta di accesso alle risorse
  - Gestione dello stato di approvazione delle richieste
  - Modalità con cui vengono fornite le risorse
  - Approvazione esplicita delle modalità con cui è concesso l'utilizzo dei dati (licenze d'uso)

# Analisi dei requisiti

# Analisi dei requisiti: requisiti della piattaforma

- Selezione delle risorse
- Richiesta di accesso alle risorse
- Pannello di amministrazione
- Cronologia di tutte le richieste
- Dettaglio e accettazione/rifiuto richiesta
- Generazione di link per il download

Funzionalità essenziali per permettere accesso alle risorse e gestione delle richieste d'accesso



# Analisi dei requisiti: raffinamento dei requisiti

## **Autenticazione**

- Registrazione
- Recupero password
- Autenticazione

## **Gestione utenti**

- Lista utenti esistenti
- Dettaglio utente esistente
- Disabilitazione utente
- Creazione utente

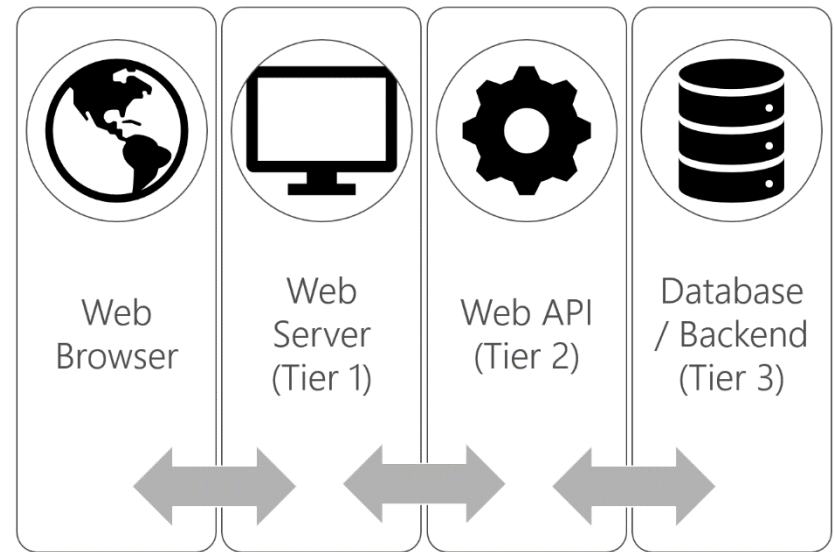
## **Gestione Risorse**

- Lista risorse esistenti
- Dettaglio risorsa esistente
- Eliminazione risorsa
- Creazione nuova risorsa
- Creazione e modifica versione

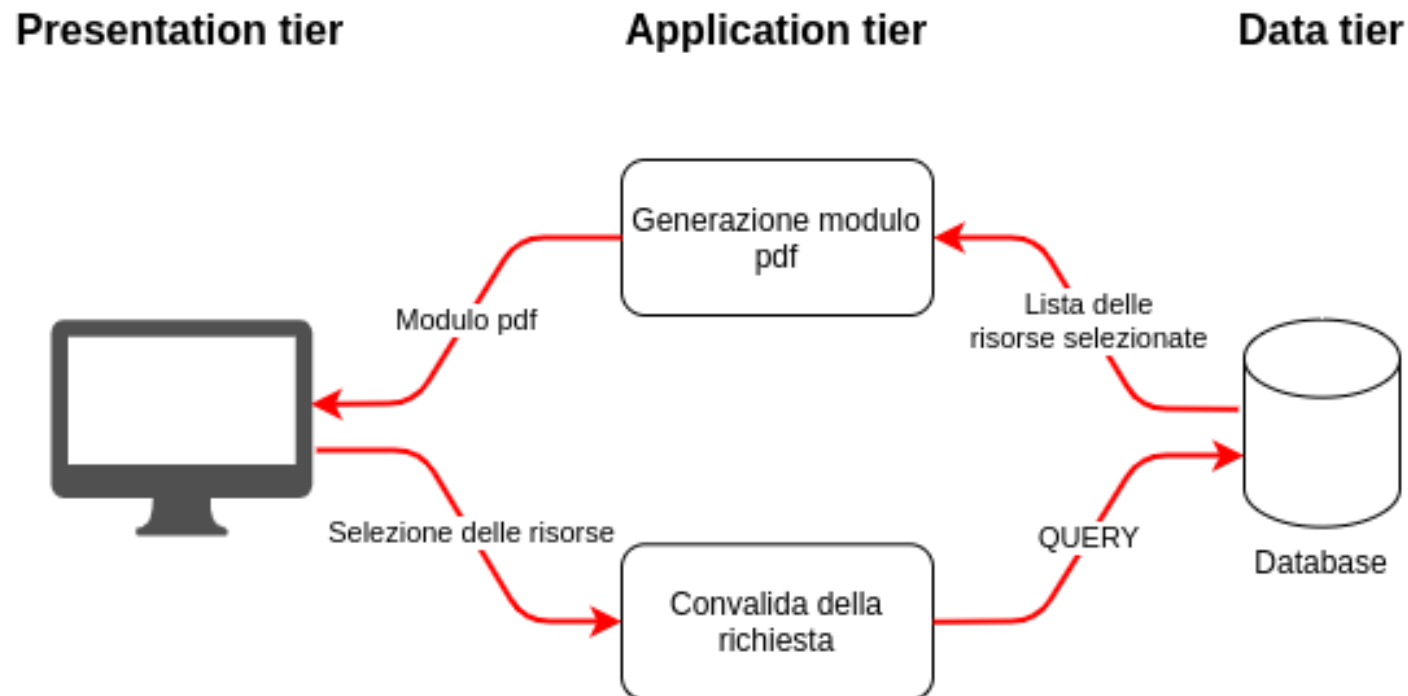
# Progettazione

# Progettazione: architettura

- **Presentation tier**, interfaccia utente, consente di utilizzare le funzionalità della piattaforma
- **Application tier**, parte applicativa e logica, convalida dei dati e comunicazione fra gli altri tier
- **Data tier**, database e servizi necessari per gestione e interrogazione
- Scalabilità, modularità e flessibilità

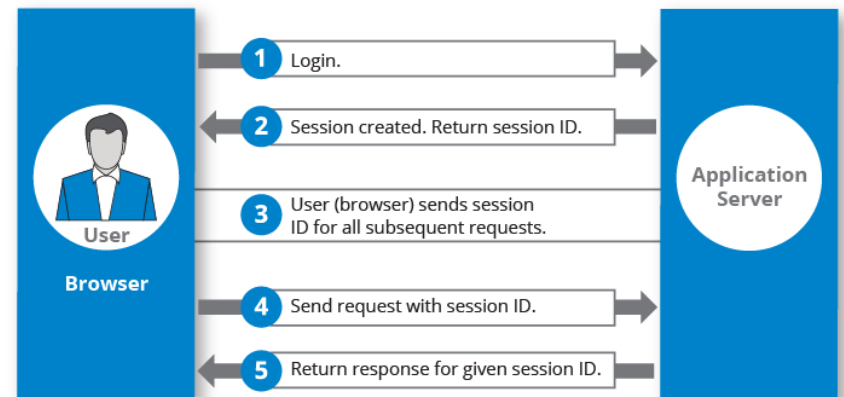


# Progettazione: architettura

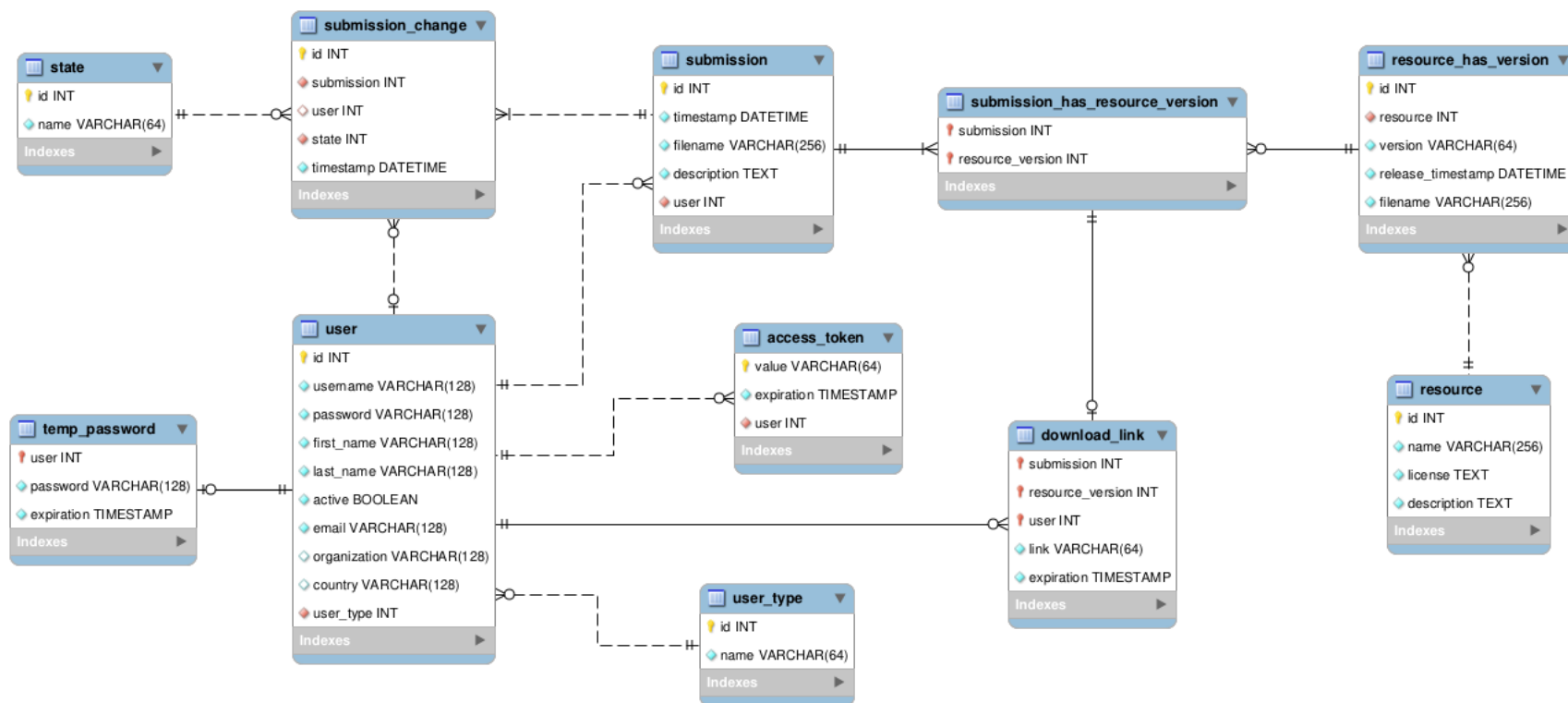


# Progettazione: sessione utente

- Possibilità per un utente di usufruire di un servizio continuativamente, senza doversi autenticare per ogni operazione
- Approccio basato su **token**, codice identificativo associato all'utente per la relativa sessione
- Validità per un breve periodo di tempo, rinnovato ad ogni operazione dell'utente, garantendo continuità della sessione



# Progettazione: schema concettuale



# Implementazione

# Implementazione: data tier

- **API** volte a stabilire una connessione con il database ed effettuare modifiche e interrogazioni
- **Pattern DAO (Data Access Object)**
  - Una classe DTO (Data Transfer Object) che rappresenta un'entità del database
  - Un'interfaccia DAO che definisce i metodi per l'interazione con la relativa entità
  - Almeno una classe che implementi il DAO, responsabile dell'effettiva connessione e dell'azione sul DB

```
1 @Getter
2 @Builder
3 public class User
4 {
5     @Setter
6     private Integer id;
7     private String username;
8     private String password;
9     private String firstName;
10    private String lastName;
11    private String email;
12    private String organization;
13    private String country;
14    private int type;
15    private boolean active;
16 }
```

```
1 public interface UserDao
2 {
3     void insertUser(User user);
4
5     void updateUser(User user);
6     void updateUserDetails(User user);
7     void updateUserPassword(int id, String password);
8
9     void deleteUser(int id);
10
11    Optional<User> getUser(int id);
12    Optional<User> getUser(String username, String password);
13    Optional<User> getUserByEmail(String email);
14
15    List<User> getUsers();
16 }
```



# Implementazione: application tier

- Parte logica ed applicativa della piattaforma: controllo della sessione utente, vincoli derivanti dai requisiti e implementazione delle funzionalità
- **API** volte a rendere disponibili le funzionalità tramite **richieste HTTP**, utilizzabili tramite browser web e quindi dal presentation tier
- **Java + Spring Boot**, framework che include Dependency Injection e permette di ottenere un web server stand-alone con la sola compilazione



# Implementazione: application tier

- Classi **Controller** espongono dei metodi per eseguire le funzionalità applicative
- I metodi vengono mappati a degli **endpoint** che ne permettono l'invocazione tramite richieste HTTP
- **UserController**
  - signup
  - login
  - getUsers
  - ...

```
1 @PostMapping(path = "/signup")
2 public ResponseEntity signUp(@RequestParam String username,
   @RequestParam String email, @RequestParam String firstName,
   @RequestParam String lastName, @RequestParam String organization,
   @RequestParam String country)
3 {
4     if (StringUtils.isEmpty(username, email, firstName, lastName,
5         organization, country))
6         return RestMessage.badRequest()
7             .body("All fields are mandatory.")
8             .toResponseEntity();
9
10    UserDao userDao = new UserSQL();
11    AccessTokenDAO tokenDAO = new AccessTokenSQL();
12
13    Optional<RestMessage> constraintViolations =
14        getConstraintsViolations(userDao, username, email);
15    if (constraintViolations.isPresent())
16        return constraintViolations.get()
17            .accessToken(accessToken)
18            .toResponseEntity();
19
20    User user = User.builder()
21        .username(username)
22        .email(emailService.checkEmailFormat(email))
23        .firstName(firstName)
24        .lastName(lastName)
25        .password(StringUtils.generatePassword())
26        .organization(organization)
27        .country(country)
28        .type(UserType.DEFAULT.getId())
29        .active(true)
30        .build();
31
32    emailService.sendNewUserEmail(user);
33
34    userDao.insertUser(user);
35
36    Timestamp newExp = TokenUtils.getNewTimestamp(millisRefresh);
37    AccessToken newToken = AccessToken.random(newExp, user.getId());
38
39    tokenDAO.insertAccessToken(newToken);
40
41    return RestMessage.ok()
42        .body(user)
43        .accessToken(newToken)
44        .toResponseEntity();
45 }
```

# Implementazione: application tier

- Classi **Filters** per introdurre funzionalità ulteriori prima dell'esecuzione via HTTP dei metodi dei Controller
- Controllo sulla validità della sessione e sull'accesso alle funzionalità di amministrazione tramite **pattern** negli endpoint


```
1 @Component
2 public class AdminFilter extends OncePerRequestFilter
3 {
4     private final Pattern pattern = Pattern.compile("/api/admin/.*");
5
6     @Override
7     protected void doFilterInternal(HttpServletRequest request,
8                                     HttpServletResponse response, FilterChain chain)
9     {
10         AccessToken accessToken = TokenUtils.getAttribute(request);
11
12         User user = new UserSQL().getUser(accessToken.getUser()).get();
13         int userType = user.getType();
14         int adminUserType = UserType.ADMIN.getId();
15
16         if (userType == adminUserType) chain.doFilter(request, response);
17         else response.setStatus(HttpServletResponse.SC_FORBIDDEN);
18     }
19
20     @Override
21     protected boolean shouldNotFilter(HttpServletRequest request)
22     {
23         return !pattern.matcher(request.getServletPath()).matches();
24     }
25 }
```

# Implementazione: presentation tier



- **Interfaccia utente**, pagine web che consentono la comunicazione tra l'utente finale e le funzionalità dei precedenti moduli
- Approccio basato su componenti, applicato al paradigma object-oriented
- **TypeScript**, linguaggio open source che estende la sintassi di JavaScript, aggiungendo forte tipizzazione, generici, interfacce, classi enum, ...
- **Bootstrap**, framework CSS per design responsive e template di stile
- **SASS**, estensione di CSS che aggiunge variabili, funzioni e una migliore struttura



# Implementazione: presentation tier



SAPIENZA  
NLP



SAPIENZA  
UNIVERSITÀ DI ROMA

DIPARTIMENTO  
DI INFORMATICA

1

Resources selection

2


Pdf upload

Select one or more resources to download


<input type="checkbox"/>	<b>BabelNet</b>	BabelNet is both a multilingual encyclopedic dictionary, with lexicographic and encyclopedic coverage of terms, and a semantic network which connects ...	<b>V 4.0.1</b>
<input type="checkbox"/>	<b>BabelNet</b>	BabelNet is both a multilingual encyclopedic dictionary, with lexicographic and encyclopedic coverage of terms, and a semantic network which connects ...	<b>V 5.0</b>
<input type="checkbox"/>	<b>VerbAtlas</b>	VerbAtlas is a hand-crafted lexical-semantic resource whose goal is to bring together all verbal synsets from BabelNet into semantically-coherent fram...	<b>V 1.0</b>
<input type="checkbox"/>	<b>VerbAtlas</b>	VerbAtlas is a hand-crafted lexical-semantic resource whose goal is to bring together all verbal synsets from BabelNet into semantically-coherent fram...	<b>V 1.1</b>


NEXT

# Implementazione: presentation tier



SAPIENZA  
NLP



 SAPIENZA  
UNIVERSITÀ DI ROMA

DIPARTIMENTO  
DI INFORMATICA


1

Resources selection

2

Pdf upload



Upload the compiled PDF to submit your download request




Choose a file or drag and drop here

SUBMIT

# Implementazione: presentation tier

 NLP Resource Manager  Logout

### Submission Details



**Name:** Luigi Verdi

**Email:** luigi\_verdi@gmail.com

**Organization:** La Sapienza

**Country:** Italy

**Submission Date:** 11/19/2020, 12:51:29 AM

**State:** APPROVED

**State selection:**

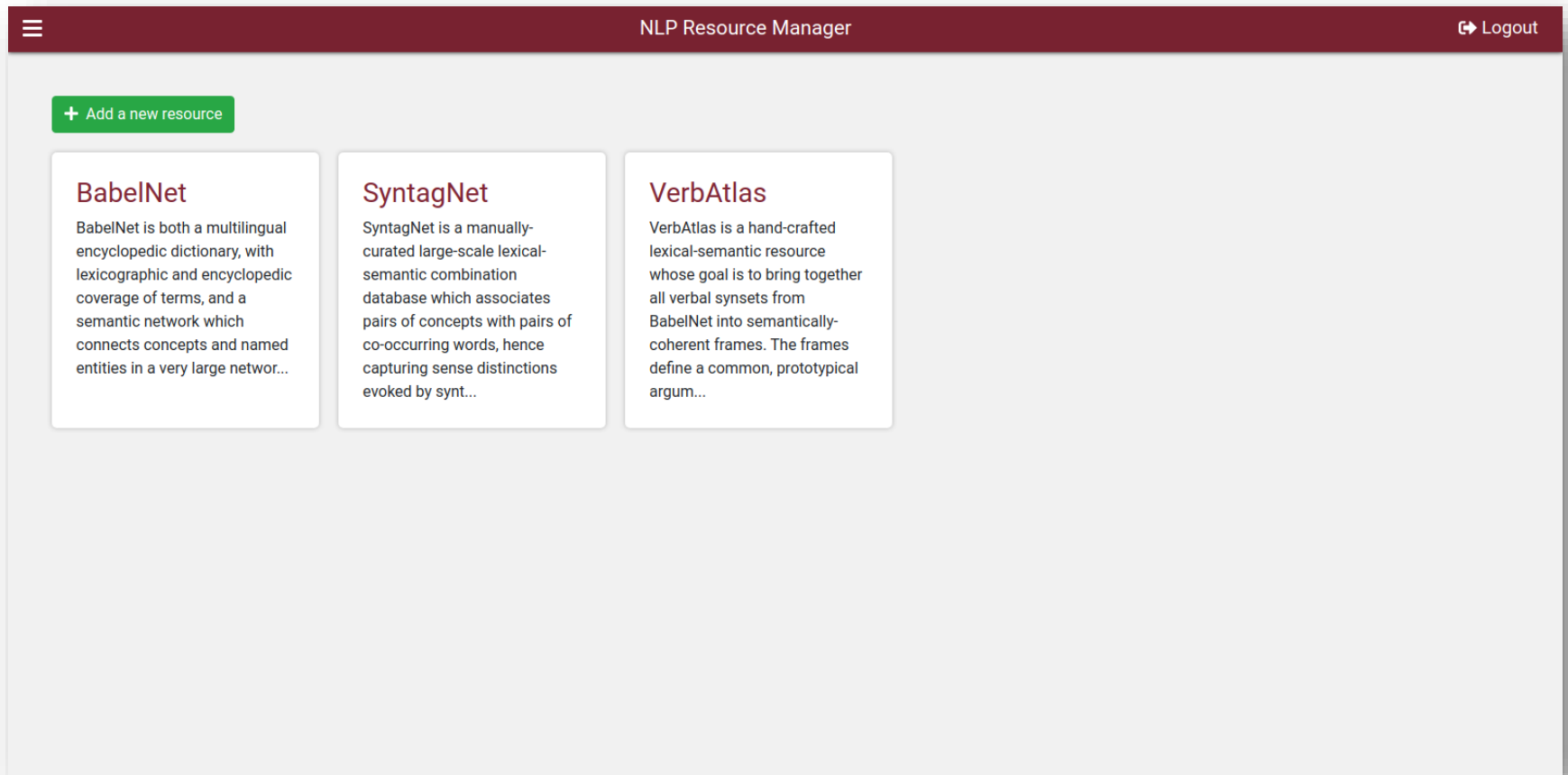
### Changes History

State	Date	User
<span>APPROVED</span>	11/19/2020, 12:53:03 AM	andrea.gasparini
<span>PENDING</span>	11/19/2020, 12:51:29 AM	

### Resources



<input checked="" type="checkbox"/>	Name	Release Date	Download Link
<input checked="" type="checkbox"/>	BabelNet 4.0.1	9/10/2020	
<input checked="" type="checkbox"/>	BabelNet 5.0	9/10/2020	
<input checked="" type="checkbox"/>	VerbAtlas 1.0	9/10/2020	
<input checked="" type="checkbox"/>	VerbAtlas 1.1	10/24/2020	

# Implementazione: presentation tier





# Implementazione: presentation tier

 NLP Resource Manager  Logout

## Resource Details


**Name:**



**Description:**

BabelNet is both a multilingual encyclopedic dictionary, with lexicographic and encyclopedic coverage of terms, and a semantic network which connects concepts and named entities in a very large network of

**License:**

BabelNet Non-Commercial License  
Release: 4.0.1  
This is a human-readable summary of (and

 Add a new version

Version	Release Date		File Name	
<input type="text" value="5.0"/>	<input type="text" value="09/10/2020"/>		<input type="text" value="BabelNet-5.0.zip"/>	<input type="button" value="Save"/>
<input type="text" value="4.0.1"/>	<input type="text" value="02/05/2020"/>		<input type="text" value="BabelNet-4.0.1.zip"/>	<input type="button" value="Save"/>

# Sviluppi futuri

# Sviluppi futuri

- Integrare un'area personale per gli utenti
- Possibilità di apporre Firma Digitale per i moduli pdf
- Possibilità di autenticarsi tramite Identità Digitale (SPID)
- Anonimizzare gli utenti disabilitati

# Progettazione e sviluppo di piattaforma per la gestione di risorse linguistiche

Facoltà di Ingegneria dell'informazione, informatica e statistica

Corso di Laurea in Informatica

Andrea Gasparini



SAPIENZA  
UNIVERSITÀ DI ROMA