Rheinische Friedrich-Wilhelms-Universität Bonn

Institut für Informatik IV

Master Thesis
in Computer Science

RDF2Résumé

A seamless platform to generate enriched Semantic Curriculum Vitæ

Enkeleda Elezi

Matriculation Number: 3055412

Examiners: Prof. Dr. Jens Lehmann & Dr. Damien Graux

March 3, 2020

Declaration of Authorship

I, Enkeleda Elezi, declare that this thesis, titled "RDF2Résumé - A seamless platform to generate enriched Semantic Curriculum Vitæ", and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have quoted from the work of others, the source is always given. Except for such quotations, this thesis is entirely my own work. I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date, Place

Signature

Abstract

Résumés are fundamental during the recruitment process. A résumé is a document used and created by a person to present their background, skills, and accomplishments. Résumé parsers used in software such as Applicant Tracking Systems may correctly interpret some parts of the content of the résumé but not other parts because of the lack of a proper standard in résumés. Résumés written in a standard format are more likely to be correctly interpreted by résumé parsers and thereby may make the candidate more findable. The standardization of résumés and the improvement of the recruitment process can be achieved by using semantic web technologies because they add domain based semantic information in order to be easily discoverable and reusable by machines, as well as humans. This Master Thesis focuses on designing a domain ontology for the résumé field in order to fully grasp the structure of it. As a consequence, résumés can then be easier processed and analyzed, as compared to today, where there exist a lot of unstructured documents. Furthermore, we present the RDF2Résumé platform that converts text/RDF data to semantically annotated HTML for a personal website and enriched PDFs, regardless of the user's amount of knowledge in semantic technologies.

Acknowledgements

First, I would like to thank Prof. Dr. Jens Lehmann for his insights on the topic. Furthermore, I would like to express my appreciation to Dr. Damien Graux for his valuable and constructive suggestions during the planning and development of this master thesis. His willingness to give his time so generously has been very much appreciated.

Additionally, I would like to thank my friends for their encouragement.

Last but not least, I would like to thank my beloved parents and brother for their love and unconditional support throughout my studies.

Contents

List of Figures

Listings

List of Tables

Chapter 1

Introduction

1.1 Motivation

During the past years, we have witnessed the accelerated evolution of the recruitment process from traditional job fairs to web-based e-recruiting platforms. The recruitment process is a process of identifying the job vacancy, analyzing the job requirements, reviewing applications, screening, shortlisting, and selecting the right candidate. Screening candidates involves reviewing résumés and cover letters, conducting video or phone interviews, and then identifying the top candidates. Therefore it is reasonable to say that résumés are fundamental in the process of hiring.

A résumé is a document used and created by a person to present their background, skills, and accomplishments. The curriculum vitae (CV) used for employment purposes in the UK (and in other European countries) is more akin to the résumé—a shorter, summary version of one's education and experience—than to the longer and more detailed CV that is expected in US academic circles [49]. Although CVs are in different formats, the majority of them have some shared information sections. These sections include information from areas such as personal information, work experience, education and training, list of skills, and competencies acquired, patents and honors, and other interests.

Currently, for human resource recruiting, the Internet is being mainly used to place online job advertisements, to perform résumé search, and to acquire information about skills and competencies of individuals [13]. As the search for employment has become more electronic, it is common for employers to only accept résumés digitally, either out of practicality or preference. This has changed much about the manner in which résumés are written, read, and processed. Many employers now find candidates' résumés through search engines, which makes it more important for candidates to use appropriate keywords when writing a résumé. Larger employers use Applicant Tracking Systems [4] to search, filter, and manage high volumes of résumés.

Another consideration for electronic résumé documents is that they are parsed with natural language processors. Résumé parsers may correctly interpret some parts of the content of the résumé but not other parts. The best résumé parsers capture a high percentage of information regarding location, names, titles, but are less accurate with skills, industries, and

other less structured or rapidly changing data. Résumés written in a standard format are more likely to be correctly interpreted by résumé parsers and thereby may make the candidate more findable. Despite having different web applications to create and collect résumés, these web applications suffer mainly from a common standard data model, data sharing, and data reusing [28].

Online recruitment processes can be efficient using semantic web technologies (Bizer et al., 2005) [9]. To effectively locate and match individuals and positions within or from outside an organization, it is important to use semantic technology [12] [31]. Semantic web technologies add domain based semantic information in order to be easily discoverable and reusable by machines, as well as humans. The use of semantic descriptions of job offers and applicant profiles allows for qualitative and quantitative reasoning about matchings between available and required skills and competencies, which is needed to improve the process of deciding who to hire and assigning individuals to tasks and teams [6]. A formal description of knowledge as a set of concepts within a domain and the relationships that hold between them is an ontology. As one of the building blocks of Semantic Technologies, ontologies are part of the W3C standards stack for the Semantic Web. They provide users with the necessary structure to link one piece of information to other pieces of information on the Web of Linked Data. Because they are used to specify common modeling representations of data from distributed and heterogeneous systems and databases, ontologies enable database interoperability, cross-database search, and smooth knowledge management.

1.2 Research Questions

Naturally, the researching questions that come to mind are:

1. Are there standard formats of résumés and maybe ontologies?
2. Are there platforms or frameworks that create résumés based on a semantic web standard?

All the above questions define the objective of our thesis work.

1.3 Contributions

In this master thesis, we design the Résumé2RDF ontology which defines the structures of classes and relations that occur in a common résumé document via Web Ontology Language(OWL) [34] and the Resource Description Framework Schema (RDFS) [42]. As a result, résumés can then be easier processed and analyzed, as compared to today, where there exist a lot of unstructured documents.

Furthermore, we present the RDF2Résumé platform that converts text/RDF data to semantically annotated HTML for a personal website and to enriched PDFs and their corresponding Latex files, regardless of the user's amount of knowledge in semantic technologies.

Annotations create a relationship between URIs and build up a network of data [33]. This means that a further search process of job candidates by employers can then be performed on a semantic-based process.

1.4 Thesis Outline

The remainder of this thesis is structured as follows. In Chapter 2, we address the background information necessary for our work. After we are familiar with the domain, we discuss in Chapter 3 related works associated with our research questions. Chapter 4 describes the first thesis contribution, the updated and improved Résumé2RDF ontology for modeling CV in data on the semantic web. The defined data structure is then used in the RDF2Résumé platform. Chapter 4 explains more on how this platform is implemented and how it exploits ontological information for seamless generation of an enriched CV and semantically annotated personal website. Afterward, Chapter 6 summarizes the work done, and we give some future work notes. Finally, the appendix contains the source texts of the developed Résumé2RDF ontology and images of the implemented platform.

Chapter 2

Background

On this chapter, we will review all the background concepts necessary for our work. First, we begin by explaining the general concept of the Semantic Web and how RDF, OWL and SPARQL play a role in creating and querying semantically annotated documents. Afterwards, we describe a bit more the standards RDFa and JSON-LD that played a crucial part in our thesis

2.1 The Semantic Web

The Semantic Web is an extension of the World Wide Web through standards set by the World Wide Web Consortium (W3C) [8]. The goal of the Semantic Web is to make internet data machine-readable. The use of technologies such as Resource Description Framework (RDF) and Web Ontology Language (OWL) enable the encoding of semantics with the data

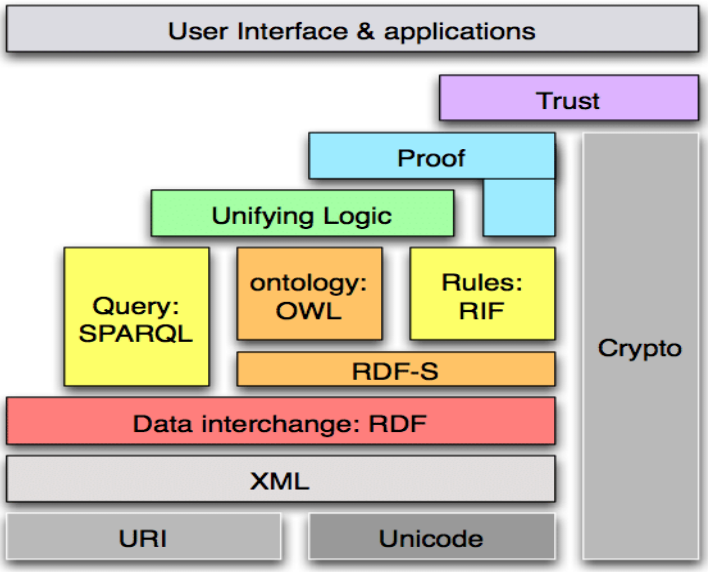

Figure 2.1.1: Semantic Web Architecture

From the figure 2.1.1, we can see how XML, RDF, RDF-S, OWL and SPARQL play a role in Semantic Web.

2.2 Resource Description Framework (RDF)

The Semantic Web is a Web of data — of dates and titles and part numbers and chemical properties and any other data one might conceive of. RDF provides the foundation for publishing and linking data [50].

The Resource Description Framework (RDF) is a framework for representing information in the Web [40]. It is based on the idea of making statements about resources (in particular web resources) in expressions of the form subject–predicate–object, known as triples. The subject denotes the resource. The predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. For example, one way to represent the notion "The sky has the color blue" in RDF is as the triple: a subject denoting "the sky", a predicate denoting "has the color", and an object denoting "blue".

RDF is an abstract model with several serialization formats (i.e., file formats), so the particular encoding for resources or triples varies from format to format. The following are some of the most popular serialization formats:

1. Turtle (Terse RDF Triple Language) [39], a serialization format focusing on human readability.
2. TriG [38], extension of Turtle.
3. N-Triples [37], a line-based, plain text format for encoding an RDF graph.
4. N-Quads [36], similiar to N-Triples
5. JSON-LD [23], a JSON-based serialization.
6. RDFa [44], an extension of HTML5 with semantic annotations.
7. RDF/XML [41], an XML-based syntax that was the first standard format for serializing RDF.

2.3 RDFS

RDF Schema (Resource Description Framework Schema), is a set of classes with specific properties using the RDF extensible knowledge representation data model, providing essential elements for the description of ontologies, otherwise called RDF vocabularies, intended to structure RDF resources [42]. These resources can be saved in a triplestore to reach them with the query language SPARQL.

2.4 OWL

The W3C OWL 2 Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things [34]. OWL is a computational logic-based language such that knowledge expressed in OWL can be reasoned with by computer programs either to verify the consistency of that knowledge or to make implicit knowledge explicit.

While RDFS offers taxonomic relations, object relations, and datatype properties, OWL provides a richer vocabulary and is more expressive:

- with OWL it is possible to specify restrictions such as cardinalities of object relations and datatype properties.
- it is possible to use logical operators in definitions (e.g., use the union of classes as a range of relation)

2.5 JSON-LD

JSON-LD [23] is a lightweight Linked Data format. It is easy for humans to read and write. It is based on the already successful JSON format and provides a way to help JSON data interoperate at Web-scale. JSON-LD is a World Wide Web Consortium Recommendation.

JSON-LD is designed around the concept of a "context" to provide additional mappings from JSON to an RDF model. The context links object properties in a JSON document to concepts in an ontology. In order to map the JSON-LD syntax to RDF, JSON-LD allows values to be coerced to a specified type or to be tagged with a language. JSON-LD is an ideal data format for programming environments, REST Web services, and unstructured databases such as CouchDB and MongoDB.

```
{
  "@context": {
    "name": "http://xmlns.com/foaf/0.1/name",
    "homepage": {
      "@id": "http://xmlns.com/foaf/0.1/workplaceHomepage",
      "@type": "@id"
    },
    "Person": "http://xmlns.com/foaf/0.1/Person"
  },
  "@id": "https://me.example.com",
  "@type": "Person",
  "name": "John Smith",
  "homepage": "https://www.example.com/"
}
```

Listing 2.1: JSON-LD Example

The example above describes a person based on the FOAF vocabulary [19]. First, the two JSON properties name and homepage and the type Person are mapped to concepts in the FOAF vocabulary, and the value of the homepage property is specified to be of the type @id, i.e., it is specified to be an IRI in the context definition. Based on the RDF model, this allows the person described in the document to be unambiguously identified by an IRI. The use of resolvable IRIs allows RDF documents containing more information to be transcluded, which enables clients to discover new data by simply following those links; this principle is known as Follow Your Nose [20].

By having all data semantically annotated as in the example, an RDF processor can identify that the document contains information about a person (@type). If the processor understands the FOAF vocabulary, it can determine which properties specify the person's name and homepage.

2.6 RDFa

RDFa [44] (or Resource Description Framework in Attributes) is a W3C Recommendation that adds a set of attribute-level extensions to HTML, XHTML and various XML-based document types for embedding rich metadata within Web documents. The RDF data-model mapping enables its use for embedding RDF subject-predicate-object expressions within XHTML documents. It also enables the extraction of RDF model triples by compliant user agents.

The essence of RDFa is to provide a set of attributes that can be used to carry metadata in an XML language (hence the 'a' in RDFa).

These attributes are:

- **about** – a URI or CURIE specifying the resource the metadata is about.
- **rel** and **rev** – specifying a relationship and reverse-relationship with another resource, respectively.
- **src**, **href** and **resource** – specifying the partner resource.
- **property** – specifying a property for the content of an element or the partner resource.
- **content** – optional attribute that overrides the content of the element when using the property attribute.
- **datatype** – optional attribute that specifies the datatype of text specified for use with the property attribute.
- **typeof** – optional attribute that specifies the RDF type(s) of the subject or the partner resource (the resource that the metadata is about).

  The following 2.2 is an example of adding Dublin Core metadata to an XML element in an XHTML file. Dublin Core data elements are data typically added to a book or article (title, author, subject, and so on.)

```
<div xmlns:dc="http://purl.org/dc/elements/1.1/"
  about="http://www.example.com/books/wikinomics">
  <span property="dc:title">Wikinomics</span>
  <span property="dc:creator">Don Tapscott</span>
  <span property="dc:date">2006-10-01</span>
</div>
```

Listing 2.2: RDFa Example

2.7 SPARQL

SPARQL (a recursive acronym for SPARQL Protocol and RDF Query Language) is an RDF query language—that is, a semantic query language for databases—able to retrieve and manipulate data stored in Resource Description Framework (RDF) format [54]. It was made a standard by the RDF Data Access Working Group (DAWG) of the World Wide Web Consortium and is recognized as one of the key technologies of the semantic web.

SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns. Triple Patterns are written as a whitespace-separated list of a subject, predicate, and object; there are abbreviated ways of writing some common triple pattern constructs.

An example of a SPARQL query for selecting the title of the instance book1 using an example ontology can be seen on listing 2.3.

```
PREFIX  dc: <http://purl.org/dc/elements/1.1/>
PREFIX  : <http://example.org/book/>

SELECT  $title
WHERE   { :book1  dc:title  $title }
```

Listing 2.3: SPARQL example

Chapter 3

Related Work

In this chapter, we will explore further the research questions described in 1.2. Each of the subsections goes further into our objectives and review relevant work. Subsection 3.1 reviews standard models and ontologies that are based upon describing résumés and noting the advantage and disadvantages of existing ones. Subsection 3.2 deals with researching available platforms or frameworks that generate CVs or personal websites through a semantic-enabled process.

3.1 Standard models/ontologies about résumés

In 2004, the European Parliament and European Commission promoted a standard CV model known as **Europass** in the European Union [56]. This standard format eases the migration of skilled workers among countries. It aims to make skills and qualifications easy to identify. Europass followed extensive changes up until 2012 when it launched its revamped website, which included a new CV template and online editor. Those improvements included:

- more user-friendly online interface with direct preview of the final document;
- new headings: Personal website, Instant messaging, Language certificate(s);
- improved tutorials for better structured information: description of projects, conferences, publications, etc.;
- new graphical identity for better legibility of Europass CV generated: new font, use of colours, simplified headings, etc.

Between February 2005 and November 2019, 150 million Europass curriculum vitae (CV) have been created online. Visits to the Europass portal (30,099,551) increased by 13% in 2019, CVs generated online (25,082,536) by ca 13%, and cover letters (2,268,666) by 11% compared to 2018 [16]. These kinds of data statistics imply that this standard has been popular and adopted by a large population. Europass helps in the idea of standardization of résumés but is not the only standard in the domain. Moreover, Europass does not include

a relevant ontology, and the website is not semantically-annotated. Our thesis contribution, Résumé2RDF, is an ontology that defines the information inside the Curriculum Vitae. Europass structure is still worth keeping and improving, and that is why our ontology was also defined in such a way our ontology can also describe that information expressed by Europass.

The application of ontologies in the résumé/recruitment domain is relevant because the search process of job candidates by employers (and dually of job offers by workers) can be performed on a semantic-based process. **ResumeRDF** is an ontology designed to express information contained within a personal resume or Curriculum Vitae (CV) on the Semantic Web. It allows describing professional work experience, education details, skills, certification and courses attended, and other relevant information [10]. This information is expressed using two namespaces:

- http://rdfs.org/resume-rdf/cv.rdfs - Resume ontology
- http://rdfs.org/resume-rdf/base.rdfs - Property value taxonomy

There is also an extension to FOAF [17] to support description of resume such as foaf:publication property to describe information about publications. Unfortunately, ResumeRDF [47] has not been developed or updated since 2007. Since then other similar ontologies have been created [30], [11], [1], [7] and [18]. ResumeRDF is able to describe more semantic relations and provide a richer source of information than the other ontologies. Even though Europass was created in 2004, ResumeRDF did not take this model into account in its design. As a result, there are inconsistencies between them, that should be updated. The foundation that ResumeRDF has defined is enough for us not to start the research from scratch. We had a starting point, and in our final point, we reached Résumé2RDF ontology. This ontology is the updated and improved second version of ResumeRDF, also pointed out by the title of the ontology.

Something else interesting worth exploring is also social networks for connecting career and business professionals. These types of connections grew exponentially in the last couple of years. Examples of such networks are **Linkedin** [26] and **Xing** [58]. These types of platforms now offer not just describing the person's background and qualifications, but also search possibilities of job postings, connecting with other professionals in different fields and offering different courses. During the definement of a user's information, we learn that these networks require more information compared to the ResumeRDF Ontology or Europass. Such information may include publications, patents, honors or awards won, and so on. Since these networks are now a normal part of the recruitment process, ResumeRDF should be updated to incorporate the necessary changes. Résumé2RDF, the second improved version, includes these changes and is up to date with nowadays recruitment process.

Moreover, there is another interesting ontology that should be considered, the **ESCO ontology** [52]. The ontology of the taxonomy "European Skills, Competences, qualifications and Occupations" considers three pillars (or taxonomy) and two registers. The three pillars are:

- Occupation

- Skill (and competences)
- Qualification

These pillars also exist in the ResumeRDF ontology, and such entities should be equally classified. The updated version of ResumeRDF, Résumé2RDF, defines these equal classifications and further improves the ontology by doing so.

3.2 Systems that generate résumés using a standard model

As mentioned earlier, Europass has a CV template and an online editor. The data required to fill in the forms match the defined Europass model. The process involves manual text input in different forms that match the corresponding sections of the model. The forms are not semantically-annotated, and the generation of pdfs is not semantic-enabled. User profiles in social apps such as Linkedin and Xing are similar in that regard to the Europass editor. RDF2Résumé platform also provides a modern user interface where the user has the option to enter manually information describing their background and skills. On the other side, the process inside the RDF2Résumé platform is semantic-enabled, and generated pdfs can be enriched via DBpedia. Furthermore, the platform provides the option of downloading a personal website created by the user data that is semantically annotated via RDFa.

Have there been systems implemented before that integrated semantics for résumé creation? In 2009, Mirizzi et al. developed a semantic web-enabled system for résumé composition and publication [30]. The system automatically produces a semantically annotated résumé exploiting domain knowledge modeled with respect to an ontology designed by them. The user is guided through the composition and, exploiting ontology-based suggested information. They are helped in eliciting and making explicit their skills and competences. The system uses semantically enriched XHTML forms, via RDFa, and semantic-based tags to output the semantic-enabled version of a résumé. The aim of the system is two-fold: on the one hand, a user obtains a document that can be stored in specific semantic-based systems used by recruiting agencies, and exposed on the web, where, thanks to RDFa annotations can be indexed by emerging semantic-search engines.

Unfortunately, this system is no longer available and can not be used for further development. The use of semantic tagging for semantic-based autocompletion suggestion is not a feature included in our platform. However, the concept is worth exploring further, perhaps as future work. RDF2Résumé just like Mirizzi outputs a semantically annotated HTML form (personal website with data regarding résumés) via RDFa.

Another research worth mentioning is hresume [22]. hResume is a microformat for publishing résumé or Curriculum Vitae (CV) information using (X)HTML on web pages. Like many other microformats, hResume uses HTML classes and rel attributes to make an otherwise non-semantic document more meaningful. On its website, it is mentioned that the social networking sites Linkedin and Xing use hresume in their public profile pages, but in a quick inspection of the HTML code, one can see that the relevant tags and classes are missing. Additionally, the hresume creator tool is no longer available. The contradiction might also come from the fact that the hResume website has been last modified on June 23rd, 2013.

Most of the available systems, such as CV parsers developed by Sovren [53] or Faviki [29], are devoted to extracting and structuring semantic information from a semi-structured document as a résumé is. The idea is that once the information is extracted, then it is possible to semantically enrich and tag it with semantic-based annotation. Other systems embed semantic information within a résumé during the writing process, such as hresume [2] or Mirizzi [30]. On the other hand, these systems seem to be no longer available and functioning. We combined these approaches and developed the RDF2Résumé platform. RDF2Résumé platform provides a modern user interface to output a semantic-enabled CV. Based on the Résumé2RDF Ontology, the application outputs JSON-LD résumé files, personal websites semantically annotated via RDFa and enriched PDFs via DBpedia.

Chapter 4

Résumé2RDF Ontology

As previously mentioned, the starting base ontology for our research is ResumeRDF [10]. This ontology is provided only in English, but our improved and updated version of it is available in 5 languages: English, German, Italian, French and Albanian. The ontology name is Résumé2RDF, and its document specification can be accessed in the linked repository[1]. The name derives from being the second improved and updated version of ResumeRDF.

During the designing of the ontology, other sources of information about common CV data were taken into consideration. The sources are social networks such as Linkedin [26] and Xing [26] and the CV model created by the European Parlament, Europass [56].

The base ontology has one primary entity **CV** that contains various types of CV resources pointing to classes such as WorkHistory, Education, Course, Skill, Reference, Target and Other. During the improvement and update process of the ontology, other subclasses were added such as Publication, Patent, Project and HonorAward.

The CV property **aboutPerson** provides the link between the CV and the person it describes. We associate the main class with the subclasses of the CV resources with the following corresponding properties: hasWorkHistory, hasEducation, hasCourse, hasSkill, hasReference, hasTarget, hasOther, hasPublication, hasPatent, hasProject, hasHonorAward. A visualization of these links can be viewed in figure 4.0.1. This visualization and further ones are created using WebVowl [57][27].

The main CV class also holds information that describes the date of its last change, title and copyright. The class also states whether the CV is active and whether the CV is confidential.

The following describes the modelling of CV data for each type of CV resource. It should be noted that only the most significant properties and decisions made are mentioned. For a more precise and detailed specification, the complete source text of the ontology should be accessed in the repository[2].

- **Data about a Person (class Person)**

[1] https://github.com/e-elezi/rdf2resume/blob/master/Ontologies/OntologyDescription/OntologyDescription.pdf

[2] https://github.com/e-elezi/rdf2resume/blob/master/Ontologies/resume2rdf_ontology.ttl

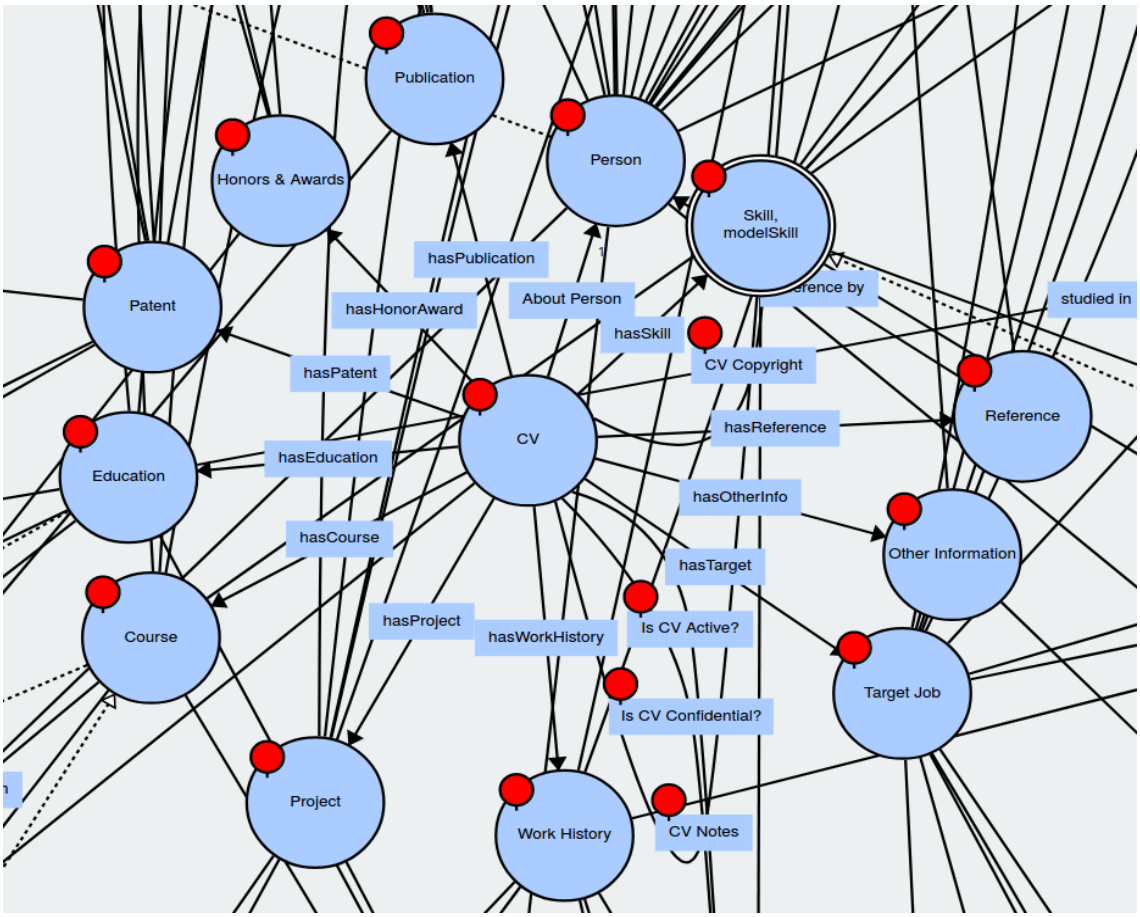

Figure 4.0.1: Visualization of CV Class

Class **Person** defines necessary personal/contact information about a person. Below the table 4.0.1 refers to the comparative study between the data shown in ResumeRDF Ontology versus the ones in Europass, Linkedin and Xing.

✓- the property exists in that domain.

X - the property does not exist in that domain.

Visualization of the final Person class included in Résumé2RDF ontology can be seen in figure 4.0.2.

The properties that are included in Résumé2RDF are:

- **firstName**, **lastName**, **formerName**, **title**
- **email**, **dateOfBirth**, **gender**, **phone**, **address**

The properties that were removed and did not make it to Résumé2RDF are: **birthPlace**, **Citizenship**, **maritalStatus**, **noOfChildren**. Modern day CVs do not need to display such information.

In addition, nowadays, a person can have multiple websites for different purposes, and the ontology should make it possible for this type of information to be saved. That is

ResumeRDF	Property type	Europass	Linkedin	Xing
cv:firstName	Literal/String	✓	✓	✓
cv:surname	Literal/String	✓	✓(lastname)	✓(lastname)
cv:otherName	Literal/String	X	✓(former name)	✓(birthname)
cv:prefix	Object	X	X	✓(title)
cv:email	Literal/String	✓	✓	✓
cv:website	Literal/String	✓	✓(multiple)	✓
cv:phone - home	Literal/String	✓	✓	X(phone)
cv:phone - work	Literal/String	✓	✓	X(phone)
cv:phone - fax	Literal/String	X	X	✓
cv:phone - mobile	Literal/String	✓	✓	✓
cv:gender	Object	✓(+ do not indicate)	X	X
cv:birthPlace	Literal/String	X	X	X
cv:hasCitizenship	Literal/String	X	X	X
cv:hasNationality	Literal/String	✓	X	X
cv:hasDriversLicense	Literal/Boolean	✓(String)	X	X
cv:maritalStatus	Literal/String	X	X	X
cv:noOfChildren	Literal/String	X	X	X
cv:Address - street	Literal/String	✓	✓(1 literal address)	✓
cv:Address - city	Literal/String	✓	X	✓
cv:Address - district	Literal/String	X	X	✓(state*)
cv:Address - country	Object	✓	X	✓
cv:Address - postcode	Literal/String	✓	X	✓
cv:dateOfBirth	Literal/Date	✓	✓(only month & date)	✓

Table 4.0.1: Comparing Person data between ResumeRDF, Europass, Linkedin and Xing

why the website property was changed from Literal to Object Property. We created a new class **Website** that holds the website URL (string) and the website type. The latter values are derived from another ontology, named **value** ontology. This ontology was designed for properties that have distinct, recognisable values. The value ontology source text can also be found in here[3]. The website values mentioned are Twitter, Facebook, Github, Instagram, Personal, Linkedin, Xing, Company, Blog, RSS Feed, Portfolio and Other from the value ontology.

One further change in the ontology is the shift of **Drivers license** property from a boolean type property into a string property to correctly point to the specific license name.

[3] https://github.com/e-elezi/RDF2Résumé/blob/master/Ontologies/resume2rdf_value_ontology.ttl

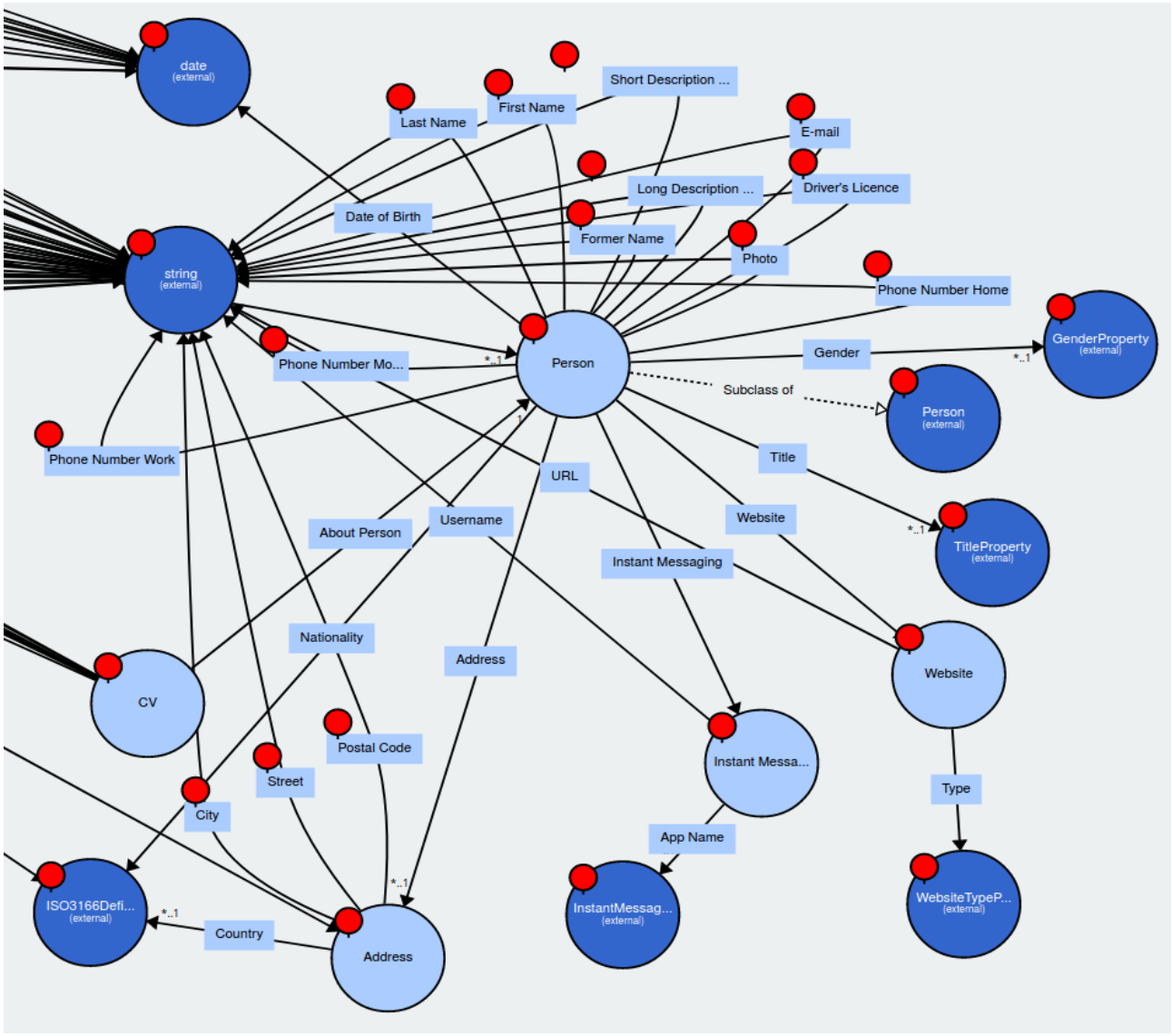

Figure 4.0.2: Visualization of Person Class

Moreover, we have added new properties such as **photo**, **personShortDescription**, **personLongDescription** and **InstantMessaging**. The photo property makes it possible for a user to upload a photo in the platform. Short and long description refer to quotes/motivation texts that a person might like to include in their resumes. Instant Messaging describes the applications' name and usernames that a recruiter could use to connect further/communicate with the job seeker. The application names are derived from the entity **InstantMessagingTypeProperty** of the value ontology with the corresponding values: Skype, Google Hangouts, WeChat.

It should also be noted that the country value is an object property that points to the specific country entity in the ontology `https://www.w3.org/Consortium/Offices/Presentations/RDFTutorial/rdfs/Countries.owl` and the **Gender** points to the entity **GenderProperty** with values: Female, Male and Not Indicated.

- **Target data (class Target)**

Class **Target** describes the CV information for the target of the job application.

The table 4.0.2 refers to the comparative study between the target data shown in ResumeRDF Ontology versus Europass, Linkedin and Xing.

ResumeRDF	Property type	Europass	Linkedin	Xing
cv:targetCareerLevel	Object	X	✓	✓
cv:targetCompanyDescription	Literal/String	X	X	X
cv:targetCompanyIndustry	Object	X	✓	✓
cv:targetCompanySize	Object	X	X	X
cv:targetCountry	Object	X	✓	✓
cv:targetJobDescription	Literal/String	X	X	X
cv:targetJobMode	Object	X	X	X
cv:targetJobType	Object	X	✓	✓
cv:targetSalary	Literal/Number	X	X	✓
cv:targetSalaryCurrency	Literal/String	X	X	X
cv:weeksNoticePeriod	Literal/String	X	X	X
cv:conditionWillRelocate	Literal/Boolean	X	X	X
cv:conditionWillTravel	Literal/Boolean	X	X	X

Table 4.0.2: Comparing Target data between ResumeRDF, Europass, Linkedin and Xing

✓- the property exists in that domain.

X - the property does not exist in that domain.

Visualization of the final Target class included in Résumé2RDF ontology can be seen in figure 4.0.3.

The properties that are included in Résumé2RDF are:

- **targetCareerLevel**, **targetCompanyDescription**, **targetCompanySize**
- **targetCountry**, **targetJobDescription**, **targetJobMode**, **targetWeeksNoticePeriod**, **targetConditionWillRelocate**, **targetConditionWillTravel**

The property **targetSalaryCurrency** was removed and the **targetSalary** was renamed to **targetSalaryRange**(string). The property **targetJobMode** was removed as well since targetJobType already provides the required information.

Additionally, we have added new properties such as **targetCompanyField**(fields/sectors of business the company belongs to), **targetRegion** and **targetJobTitle**(for a specific job title). **TargetCareerLevel** points to the entity CVCareerLevel with values: Student, Entry Level, Professional, Management, Executive and SeniorExecutive.
TargetCompanyIndustry points to the entity IndustryType. The values are influenced from Linkedin/Xing values. **TargetCompanySize** points to the entity CompanySize. The values are: Small, Medium and Large.
TargetCountry value is an object property that points to the specific country entity in this ontology https://www.w3.org/Consortium/Offices/Presentations/RDFTutorial/

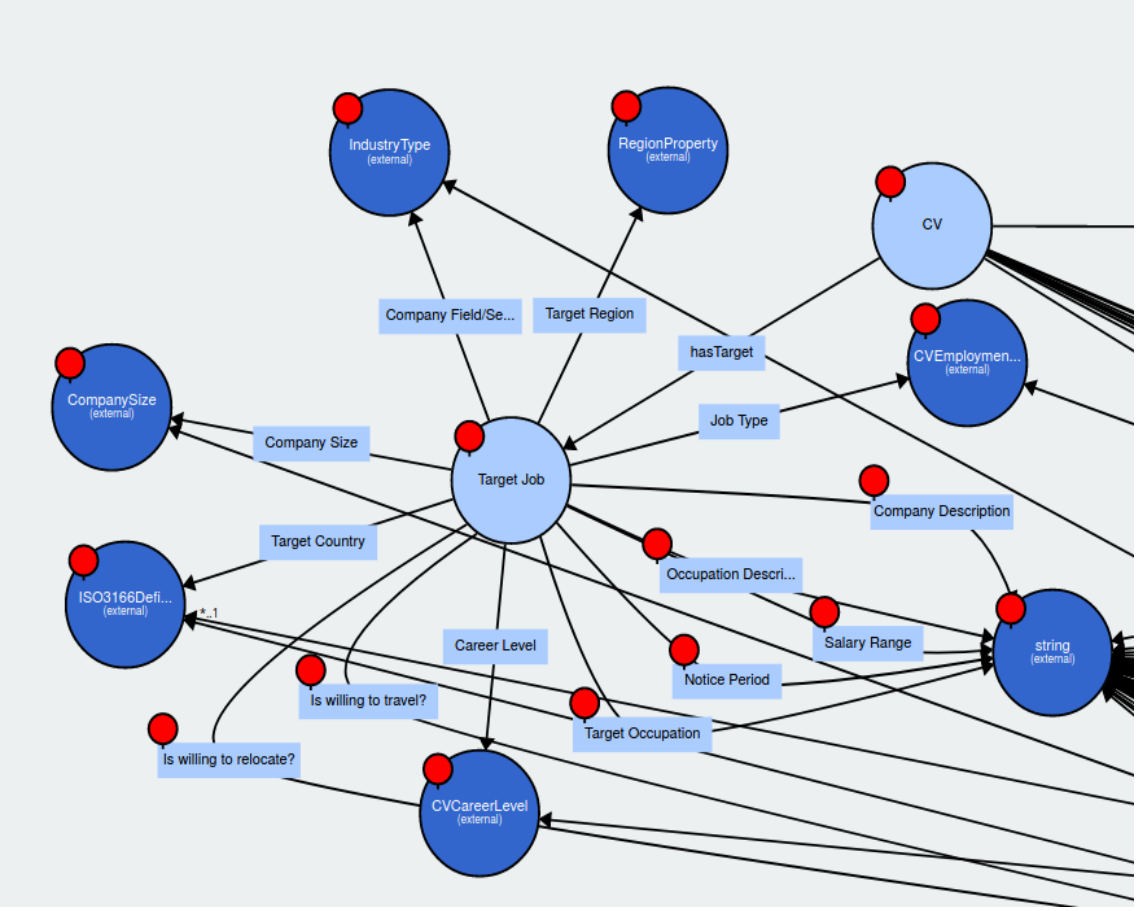

Figure 4.0.3: Visualization of Target Class

rdfs/Countries.owl.
TargetJobType points to the entity CVEmploymentType. The values are: Employee full time, Employee part time, Contractor, Intern, Self-employed, Freelance and Apprenticeship.

- **Work Experience data (class WorkHistory)**

  Class **WorkHistory** describes a past/current work experience. The organization where this experience took place is important and that is why the class **Organization** and its subclasses **Company** and **EducationalOrg** were created. A visualization of the Organization Class can be seen in figure 4.0.4.

  Below the table 4.0.3 refers to the comparative study between the work data shown in ResumeRDF Ontology versus Europass, Linkedin and Xing.

  ✓- the property exists in that domain.

  X - the property does not exist in that domain.

  Visualization of the final WorkHistory class included in Résumé2RDF ontology can be seen in figure 4.0.5.

  The properties that are included in Résumé2RDF are:

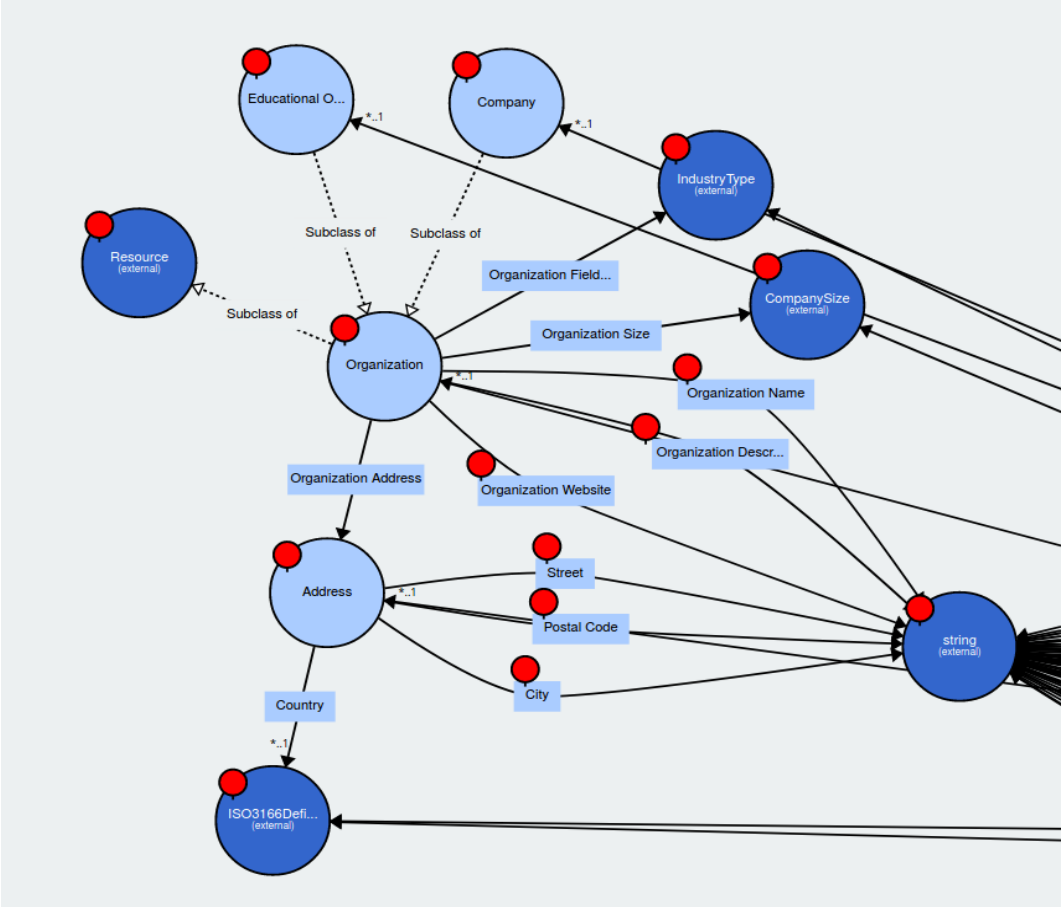

Figure 4.0.4: Visualization of Organization Class

ResumeRDF	Property type	Europass	Linkedin	Xing
cv:startDate	Literal/Date	✓	✓	✓
cv:endDate	Literal/Date	✓	✓	✓
cv:jobTitle	Literal/String	✓	✓	✓
cv:jobDescription	Literal/String	✓	✓	✓
cv:careerLevel	Object	X	X	✓
cv:numSubordinates	Literal/String	X	X	X
cv:jobType	Object	X	✓	✓
cv:orgName	Literal/String	✓	✓	✓
cv:orgLocality	Literal/String	✓(general location)	✓(general location)	X
cv:orgCountry	Object	✓	X	X
cv:orgNotes	Literal/String	X	X	X
cv:orgURL	Literal/String	X	X	✓
cv:orgIndustry	Object	✓(sector)	X	✓
cv:isCurrent	Literal/Boolean	✓	✓	✓

Table 4.0.3: Comparing WorkHistory data between ResumeRDF, Europass, Linkedin and Xing

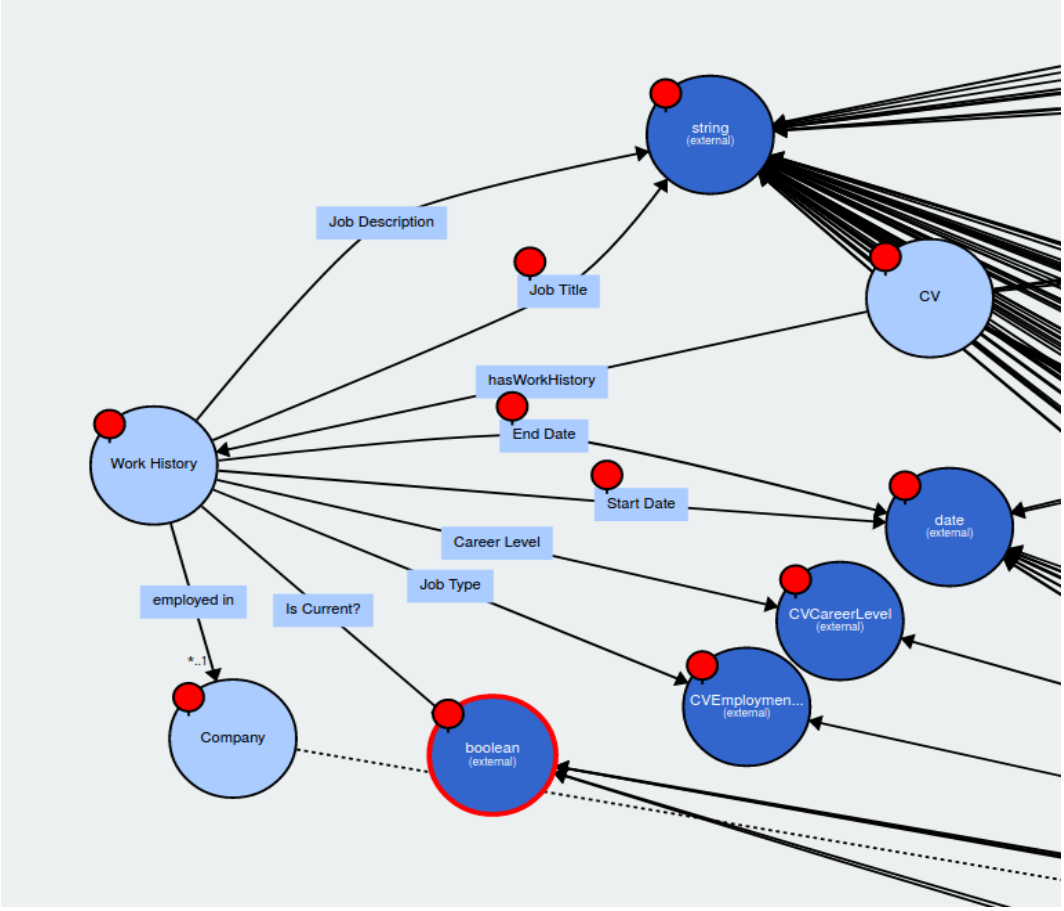

Figure 4.0.5: Visualization of WorkHistory Class

- **startDate**, **endDate**, **jobTitle**, **jobDescription**, **isCurrent**, **careerLevel**, **jobMode**
- **orgName**, **orgDescription**, **orgAddress**, **orgWebsite**

The properties that not included are: **numSubordinates** and **orgLocality**. Organization notes is updated into organization description and also URL into organization website.

We have also added a new property, **orgSize** referring to the size of the company and **orgField** referring to the field/sector of business. The second property helps recruiters get an idea of the company where the experience took place.

- **Education data (class Education)**

  Class **Education** describes a past/current education. The organization where this experience took place is important and that is why the class **EducationalOrg** was created.

  Below the table 4.0.4 refers to the comparative study between the education data shown in ResumeRDF Ontology versus Europass, Linkedin and Xing.

  ✓- the property exists in that domain.

ResumeRDF	Property type	Europass	Linkedin	Xing
cv:eduStartDate	Literal/Date	✓	✓	✓
cv:eduGradDate	Literal/Date	✓	✓	✓
cv:degree	Literal/String	✓	✓	✓
cv:degreeDescription	Literal/String	✓	✓	✓
cv:degreeType	Object	X	✓	✓
cv:orgName	Literal/String	✓	✓	✓
cv:orgLocality	Literal/String	✓	X	X
cv:orgCountry	Object	✓	X	X
cv:orgNotes	Literal/String	X	X	X
cv:orgURL	Literal/String	X	X	✓
cv:isEduCurrent	Literal/Boolean	✓	X	X

Table 4.0.4: Comparing Education data between ResumeRDF, Europass, Linkedin and Xing

X - the property does not exist in that domain.

Visualization of the final Education class included in Résumé2RDF ontology can be seen in figure 4.0.6.

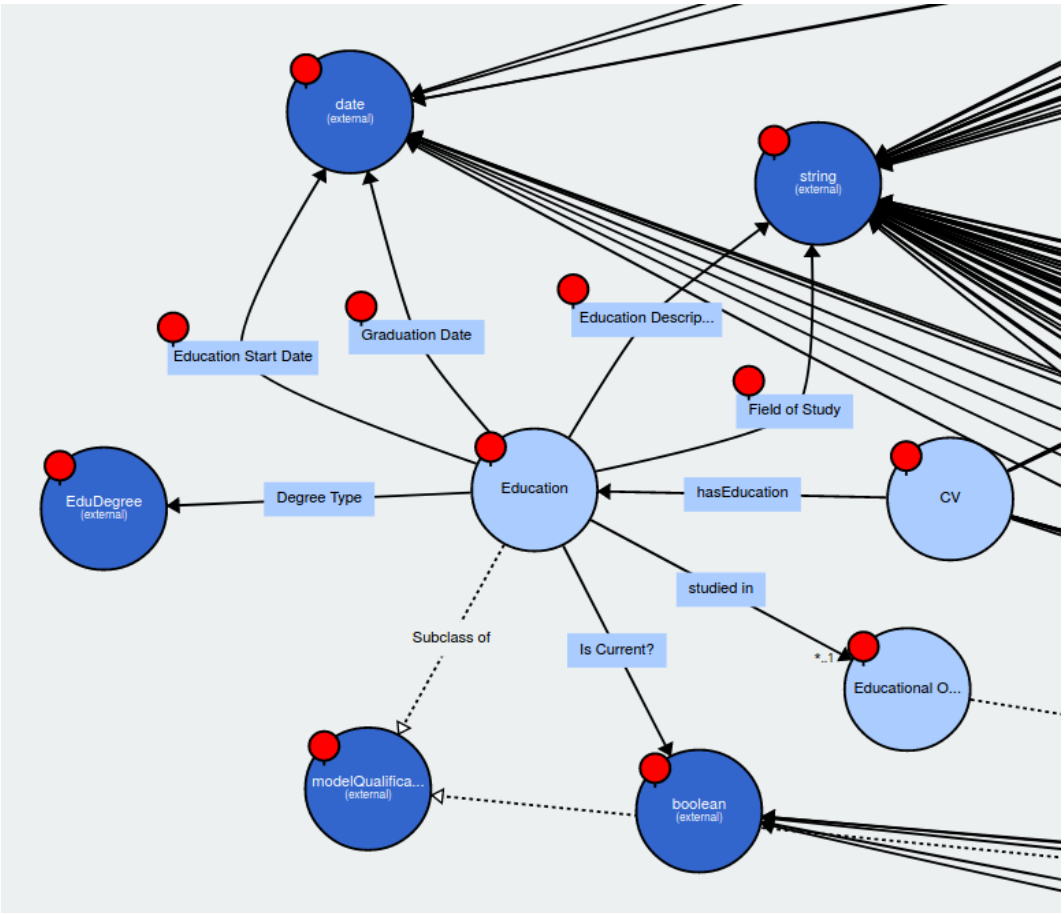

Figure 4.0.6: Visualization of Education Class

The properties that are included in Résumé2RDF are:

- **eduStartDate**, **eduGradDate**, **degree**, **degreeType**, **isEduCurrent**, **degreeDescription**
- **orgName**, **orgDescription**, **orgAddress**, **orgWebsite**

The property not included is **orgLocality**. This property was removed from the Organization Class in the previous class.

- **Course data (class Course)**

  Class **Course** describes a past/current course or training taken.

  The table 4.0.5 refers to the comparative study between the course data shown in ResumeRDF Ontology versus Europass, Linkedin and Xing.

ResumeRDF	Property type	Europass*	Linkedin	Xing
cv:courseStartDate	Literal/Date	✓	X	X
cv:courseFinishDate	Literal/Date	✓	X	X
cv:courseTitle	Literal/String	✓	✓	✓
cv:courseDescription	Literal/String	✓	X	X
cv:courseURL	Literal/String	X	X	X
cv:orgName	Literal/String	✓	X	X
cv:orgLocality	Literal/String	✓	X	X
cv:orgCountry	Object	✓	X	X
cv:orgNotes	Literal/String	X	X	X
cv:orgURL	Literal/String	X	X	X
cv:isCertified	Literal/Boolean	X	X	X

Table 4.0.5: Comparing Course data between ResumeRDF, Europass, Linkedin and Xing

✓- the property exists in that domain.

X - the property does not exist in that domain.

Visualization of the final Course class included in Résumé2RDF ontology can be seen in figure 4.0.7.

*Europass does not have a specific course adding feature. Education, course and training are handled the same way.

Xing does not have a specific course adding feature, but rather a qualification feature where one can add the name of a qualification/certification earned.

In Résumé2RDF, Course and Training are handled the same way because Training is perceived as a subclass of Course.

The properties that are included in Résumé2RDF are:

- **courseStartDate**, **courseFinishDate**, **courseTitle**, **courseDescription**, **courseURL**
- **orgName**, **orgDescription**, **orgAddress**, **orgWebsite**

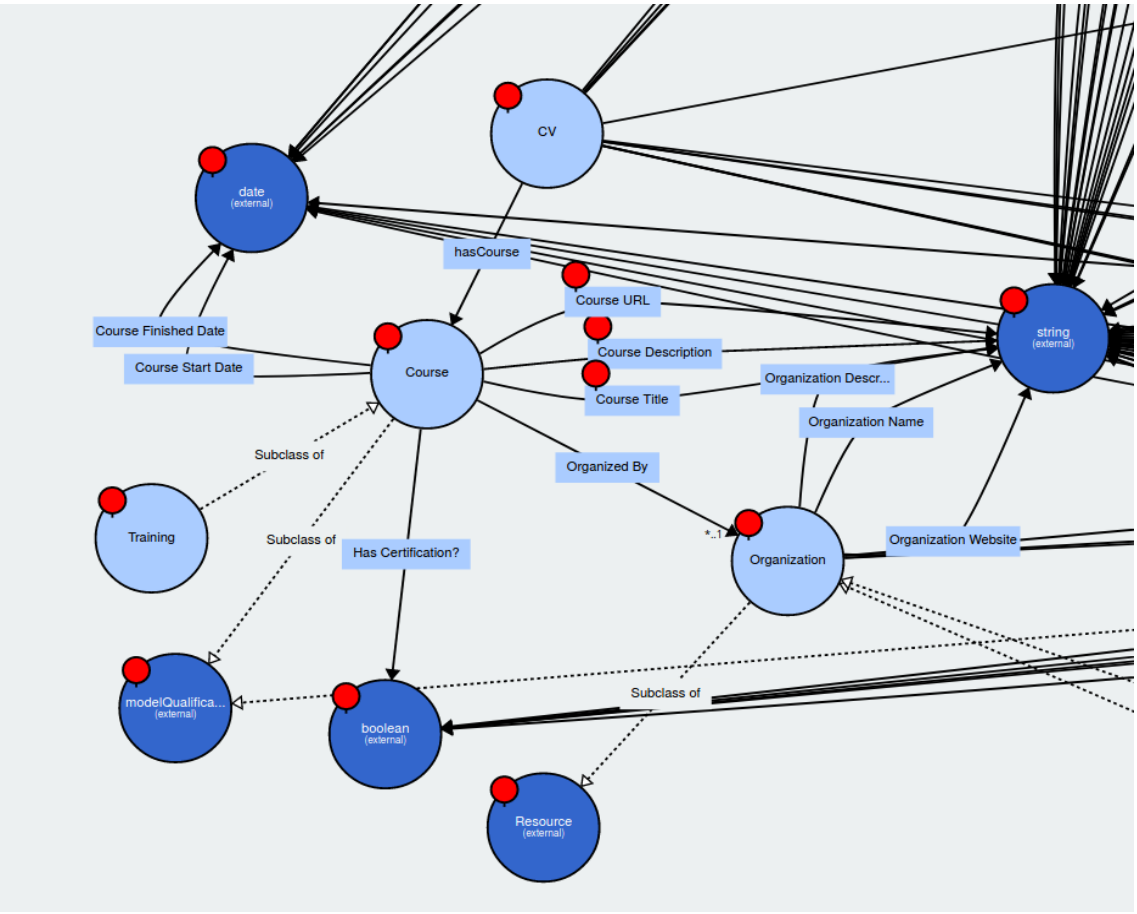

Figure 4.0.7: Visualization of Course Class

The class and its properties are still present in our updated ontology as one person can take a course from, for example, websites online, and such data should be saved and the ontology needs to make that possible.

- **Skill data (class Skill)**

  Class **Skill** describes a particular skill the job seeker acquired. This class is part of the ResumeRDF ontology.

  The way ResumeRDF, Europass, Linkedin and Xing handle skills is different in every domain, and as such, we will compare one by one.

  In the ResumeRDF ontology, the class Skill has the following properties: skillName, skillLevel(0..5), skillLastUsed, skillYearsExperience.

  On the other hand, Europass handles skills in an entirely different way. There are subclasses of skills such as, Communication skills, Organisation/managerial skills, job-related skills, digital skills. Organisational skills, job-related skills and communication ones can be filled in using just a description box. Digital skills can be self-assessed through properties such as information processing, communication, content creation, safety and problem-solving. A person can also add related ICT certificates and a general description for other computer skills.

In Xing, one can add skills by just their skill name and then use drag and drop to create a top skills list.

In Linkedin, a person can also add skills only by their skill name. Typically, when the skills are listed, they are grouped into four categories: Industry knowledge, Tools & Technologies, Interpersonal Skills and Other Skills. According to endorsement given by the person's connections, the skills are ordered by top endorsements.

While designing the Skill class for Résumé2RDF all this information was taken into account. The properties that made it to the final design are: **skillName**, **skillDescription**, **skillCategory**(deriving from the values from Linkedin), **skillLevel**(integer from 0 to 5), **skillHasCertificate** (boolean) and **skillCertificateName**. This ontology can express the Europass skill model by putting the appropriate skill name and its description. This Skill class is also equivalent to the skill class of the ESCO ontology[52].

Visualization of the final Skill class included in Résumé2RDF ontology can be seen in figure 4.0.8.

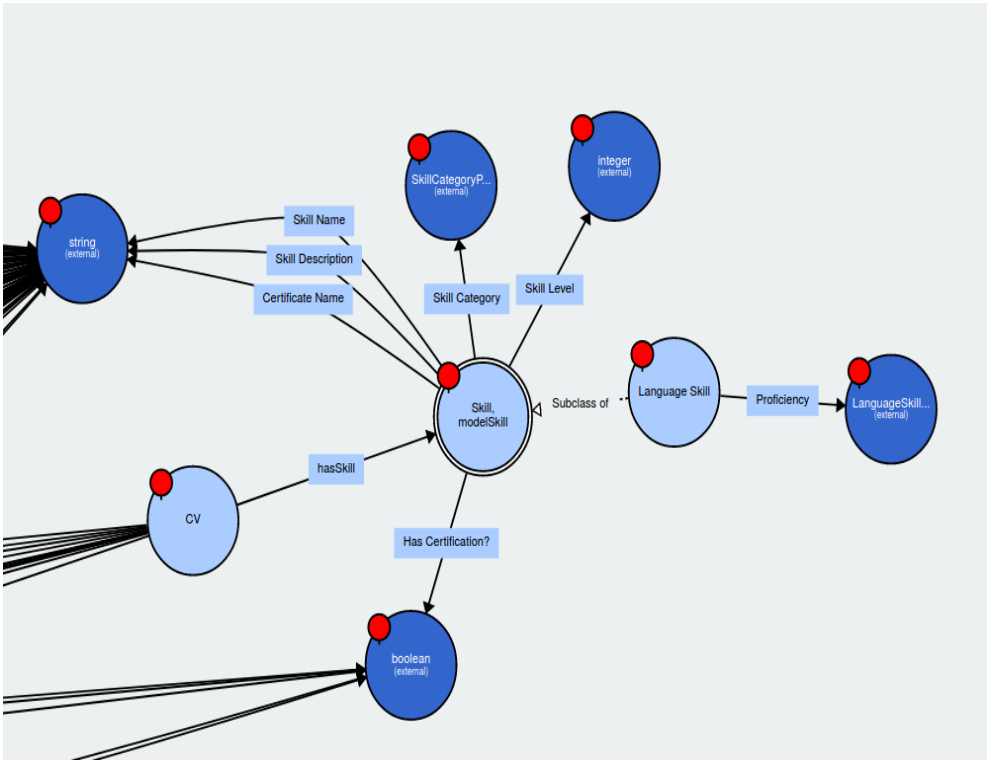

Figure 4.0.8: Visualization of Skill Class

- **Language Skill data (class LanguageSkill)**

  Class **LanguageSkill** describes a certain language skill the person who owns the CV gained. This class is part of the ResumeRDF ontology.

  The way ResumeRDF, Europass, Linkedin and Xing handle language skills is different in every domain and as such, we will compare one by one.

  In the ResumeRDF ontolgy, the class LanguageSkill, subclass of Skill, has the following properties: lngSkillLevelReading(0..5), lngSkillLevelWritten(0..5) and the inherted ones

from Skill.

On the other hand, Europass handles language skills differently. A person can add a new language skill (foreign language) by defining the following data: language name, self-assessment data for listening, reading, speaking and writing and a diploma/certificate name about this particular language. A person can also add their mother tongue name(s).

In Xing, one can add a language skill by adding the language name and the skill level (Basic knowledge, Good knowledge, Fluent, First Language).

In Linkedin, a person can also add a language skill by adding the language name and the proficiency level (Elementary proficiency, Limited working proficiency, Professional working proficiency, Full professional proficiency, native or bilingual proficiency).

While designing the Language Skill class for Résumé2RDF, all this information was taken into account. Language Skill is a subclass of the Skill class. The properties that made it to the final design are **languageSkillProficiency**(values influenced by Linkedin) and the ones inherited from Skill.

- **Project data (class Project)**

  Class **Project** describes a past/current project that the person took/is taking place in. This class did not exist in the ResumeRDF Ontology.

  During the comparative study between the data in ResumeRDF versus Linkedin, Xing and Europass, the following things were noted.

  In Europass, one can add a project not as a specific feature, but as additional information of type project and a description of such information.

  Xing does not have a specific project adding feature.

  In Linkedin, one can add a project as a specific feature. The properties required while filling the forms are: **projectName**, **projectStartDate**, **projectEndDate**, **projectCreator**, **projectURL**, **projectDescription**, **projectIsCurrent**.

  Résumé2RDF describes the Project class with the same properties found in Linkedin and adds the new property: **projectRole**(String) to describe the role the person had in this project. Visualization of the Project class can be viewed in figure 4.0.9.

- **Publication data (class Publication)**

  Class **Publication** describes a published work of the job seeker. This class did not exist in the ResumeRDF Ontology.

  During the comparative study between the data in ResumeRDF versus Linkedin, Xing and Europass, the following things were observed.

  In Europass, one can add a publication only as additional other information of type publication and a description of such information.

  Xing does not have a specific publication adding feature.

  In Linkedin, a person adds a publication via a specific feature. The properties required while filling the forms are: **publicationTitle**, **publicationPublisher**, **publicationDate**, **publicationAuthor**, **publicationURL**, **publicationDescription**.

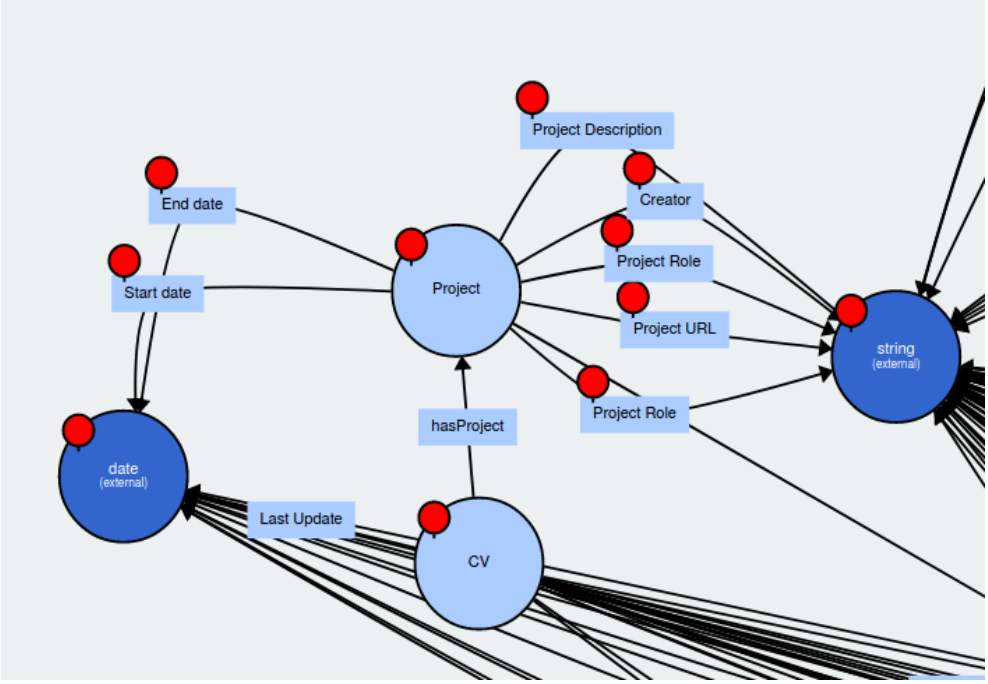

Figure 4.0.9: Visualization of Project Class

Résumé2RDF defines the Publication class with the same properties found in Linkedin. Visualization of the Publication class can be viewed in figure 4.0.10.

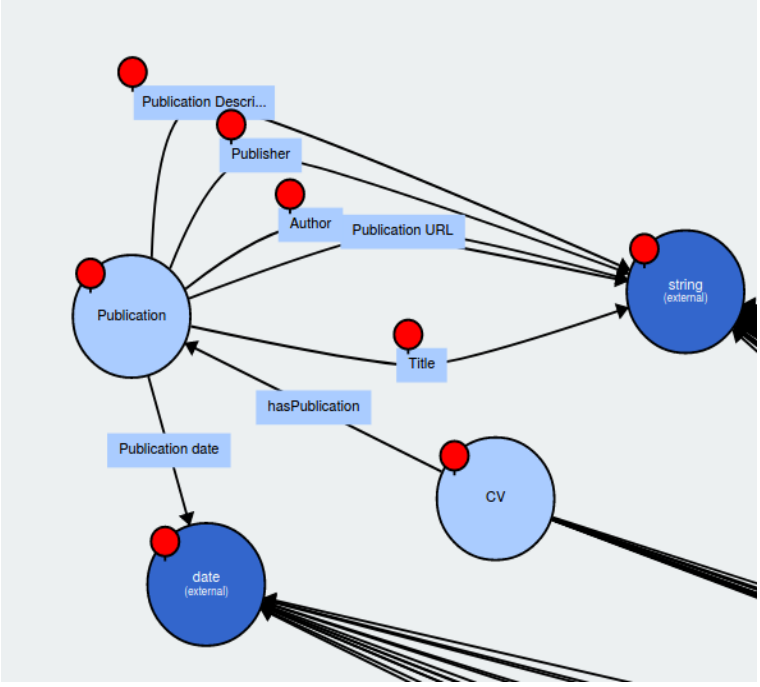

Figure 4.0.10: Visualization of Publication Class

- **Patent data (class Patent)**

  Class **Patent** describes a particular patent that the person who owns the CV applied

for/registered. This class did not exist in the ResumeRDF Ontology.

During the comparative study between the data in ResumeRDF versus Linkedin, Xing and Europass, the following things were noted.

Europass and Xing do not have a specific patent adding feature.

In Linkedin, one can add a patent and the properties required while filling the forms are: **patentTitle**, **patentOffice**, **patentNumber**, **patentInventor**, **patentStatus**, **patentIssueDate**, **patentURL**, **patentDescription**.

Résumé2RDF describes the Patent class with the same properties found in Linkedin. Visualization of the Patent class can be viewed in figure 4.0.11.

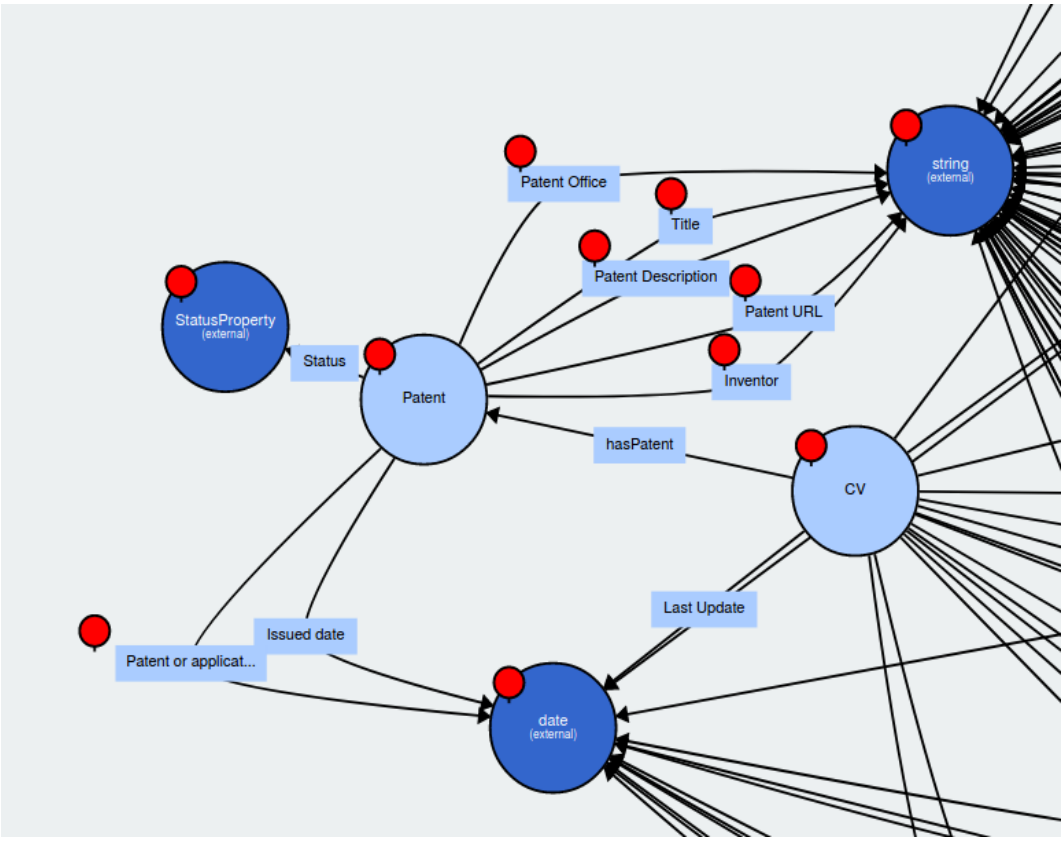

Figure 4.0.11: Visualization of Patent Class

- **Honor & Awards data (class HonorAward)**

  Class **HonorAward** describes an honor or award given to the job seeker. This class did not exist in the ResumeRDF Ontology.

  During the comparative study between the data in ResumeRDF versus Linkedin, Xing and Europass, the following things were noted.

  In Europass, a person can add an honor or award only as additional other information of type Honor/Award and a description of such information.

  Xing has a specific feature to add an award and the required properties are awardYear, awardName, awardLink.

Linkedin allows a person to add a honor/award. The properties needed while filling the forms are: **awardTitle**, **awardIssuer**, **awardIssueDate**, **awardDescription**.

Résumé2RDF describes the HonorAward class with the same properties found in Linkedin as it is a generalized version of the ones from Xing. Visualization of the HonorAward class can be viewed in figure 4.0.12.

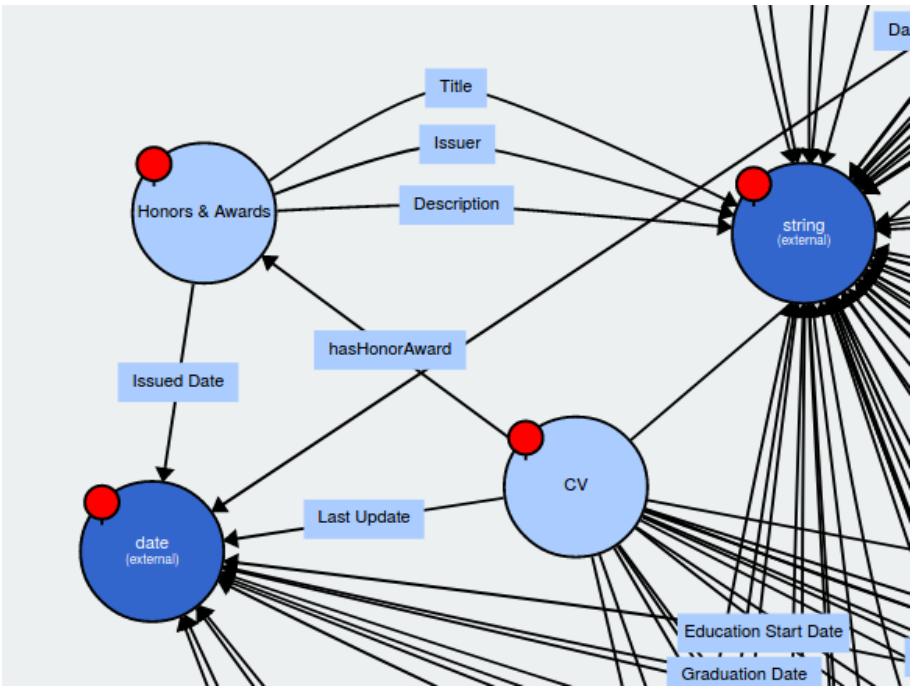

Figure 4.0.12: Visualization of Honor Class

- **Reference data (class Reference)**

  Class **Reference** points to a person who provides the reference with the **referenceBy** property. This class is part of the ResumeRDF. During the design, we also added the property **refRelationDescription** to specifically describe the relation between the person who owns the CV and the person who will provide the reference.

  Europass, Linkedin and Xing do not have a specific feature to add a reference. Typically, references are done upon request of the recruiter, and not part of the CV.

  Résumé2RDF includes the Reference class for future further development. Visualization of the Reference class can be seen in figure 4.0.13.

- **Other Information data (class OtherInformation)**

  Class **OtherInformation** describes a type of information that the person could not insert before. This class is part of the ResumeRDF and makes possible adding information such as different interests, hobbies, organisations involved with and so on.

  Europass lets you add additional information of a specific type and a description of such information.

  Xing has a specific feature to add only organisations or interests and not other type of information.

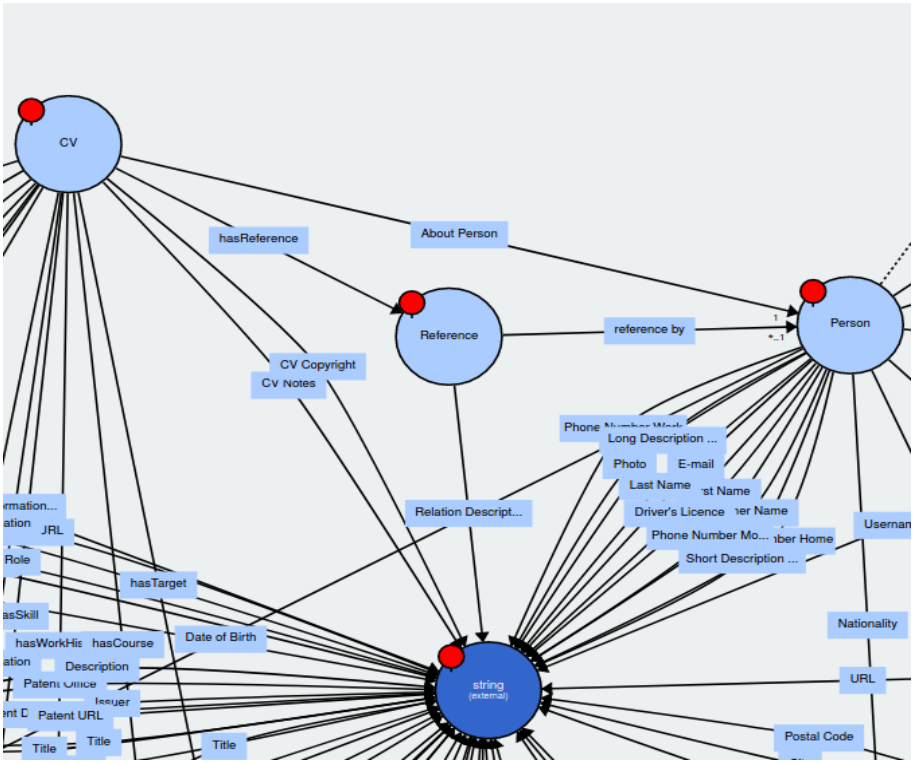

Figure 4.0.13: Visualization of Reference Class

Linkedin allows a person to add interests and voluntary experiences.

Résumé2RDF describes the **OtherInformation** class with the same properties found in ResumeRDF as its a good generalization model of this information. Visualization of the Reference class can be seen in figure 4.0.14.

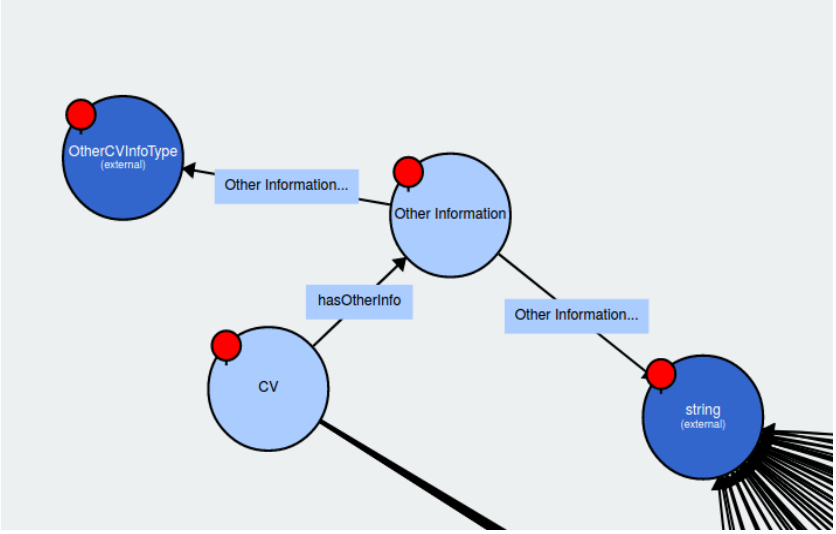

Figure 4.0.14: Visualization of Other Information Class

As a result of this research, the required classes and properties are defined as part of the Résumé2RDF Ontology.

Chapter 5

RDF2Résumé Platform

RDF2Résumé [43] is a platform for writing and maintaining a CV guided by a semantic-aided user interface. Built like a single-page web application, RDF2Résumé intakes user data in manual text form or file type, and outputs a semantically annotated HTML and enriched Curriculum Vitae PDFs. As a result, recruiters and the recruitment process as a whole will be improved and performed on a semantic-based process.

The platform is open-source, and the source code can be found here[1]. A little preview of the platform's features can be seen in this video[2]. The application's characteristics, architecture and documentation of its implementation and main features can be found in the sections below.

5.1 Platform's Characteristics

Before diving in deeper into the platform inner details, let us first understand the type of users the platform is targeted to and the primary use cases considered before its design and logic were implemented.

5.1.1 Target Audience

RDF2Résumé is designed to be used by every person that wants to describe their background information, skills, or achievements, regardless of their knowledge in the semantic web. With simple inputs that match everyday web application uses, the user can then download files that are either semantically annotated or created through a semantic-enabled process seamlessly. There are no specific users in the app at the moment, every user is regarded as a casual user.

[1] https://github.com/e-elezi/rdf2resume

[2] https://www.youtube.com/watch?v=b-7C98PqaaI&feature=youtu.be

5.1.2 Use Cases

The use cases regarding our platform are defined below, and also in the figure 5.1.1.

- Saving CV data in an RDF form
  A casual user enters CV related data in a text input based form. The user also writes input in different languages. In the end, the user saves the data entered in an RDF standard.

- Uploading an RDF file that matches ontology
  A casual user uploads an RDF file, the format of which matches the ontology. The data inside the file will then fill the respective forms automatically.

- Editing previously saved data
  A casual user uploads a file. The file contains CV related data in an RDF standard. The data in the file fills the corresponding text inputs automatically. The user updates already saved data, adds new data or deletes data that is no longer true.

- Generating a résumé pdf

  - Alternative Flow 1
    A casual user enters CV related data in a text input based form. After finishing the input process, the user chooses the design and the language the pdf will be generated in. In the end, the user generates a simple résumé pdf via the data entered up until that point.
  - Alternative Flow 2
    A casual user uploads a file. The file contains CV related data in an RDF standard. The user can either further update their information or choose the design and the language the pdf will be generated in. The user generates a simple résumé pdf via the uploaded data.

- Generating an enriched résumé pdf via a semantic-enabled process

  - Alternative Flow 1
    A casual user enters CV related data in a text input based form. After finishing the input process, the user chooses the design and the language the pdf will be generated in. In the end, the user generates an enriched résumé pdf via the data entered up until that point and a a semantic-enabled process.
  - Alternative Flow 2
    A casual user uploads a file. The file contains CV related data in an RDF standard. The user can either further update their information or choose the design and the language the pdf will be generated in. The user generates an enriched résumé pdf via the uploaded data and a semantic-enabled process.

- Generating a semantically annotated personal website

- Alternative Flow 1
  A casual user enters CV related data in a text input based form. After finishing the input process, the user chooses the language the website will be generated in. In the end, the user generates a modern-looking personal website semantically annotated via the data entered up until that point.
- Alternative Flow 2
  A casual user uploads a file. The file contains CV related data in an RDF standard. The user can either further update their information or choose the language the website will be generated in. In the end, the user generates a modern-looking personal website semantically annotated via the data entered up until that point.

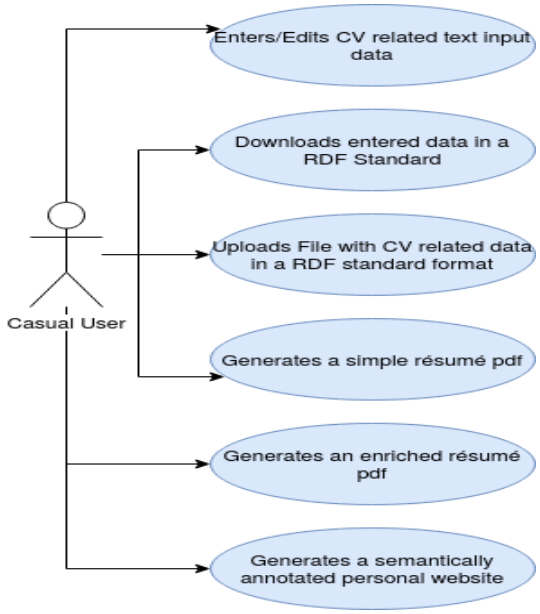

Figure 5.1.1: RDF2Résumé's use cases diagram

5.2 Systems design

Systems design is the process of defining the architecture, modules, components and data for our platform to satisfy the specified characteristics above.

5.2.1 Architecture

As most users are on the web these days, it is only logical that the most used apps are web applications. That would mean that implementing a web application would attract many more users compared to, for example, a desktop application. In addition, most generating résumé applications are also web ones.

The characteristics defined in the use cases in 5.1.2 naturally are not appropriate for an only mobile application. The processing required is best suited for a backend web server. On

the other hand, many web applications nowadays can be used using a mobile thanks to the responsiveness of their design.

RDF2Résumé platform was designed as a single-page web application(SPA) [51] that has a responsive design regarding other devices such as tablets and mobile phones. The web application communicates with a backend web server that does most of the processing, as shown in figure 5.2.1.

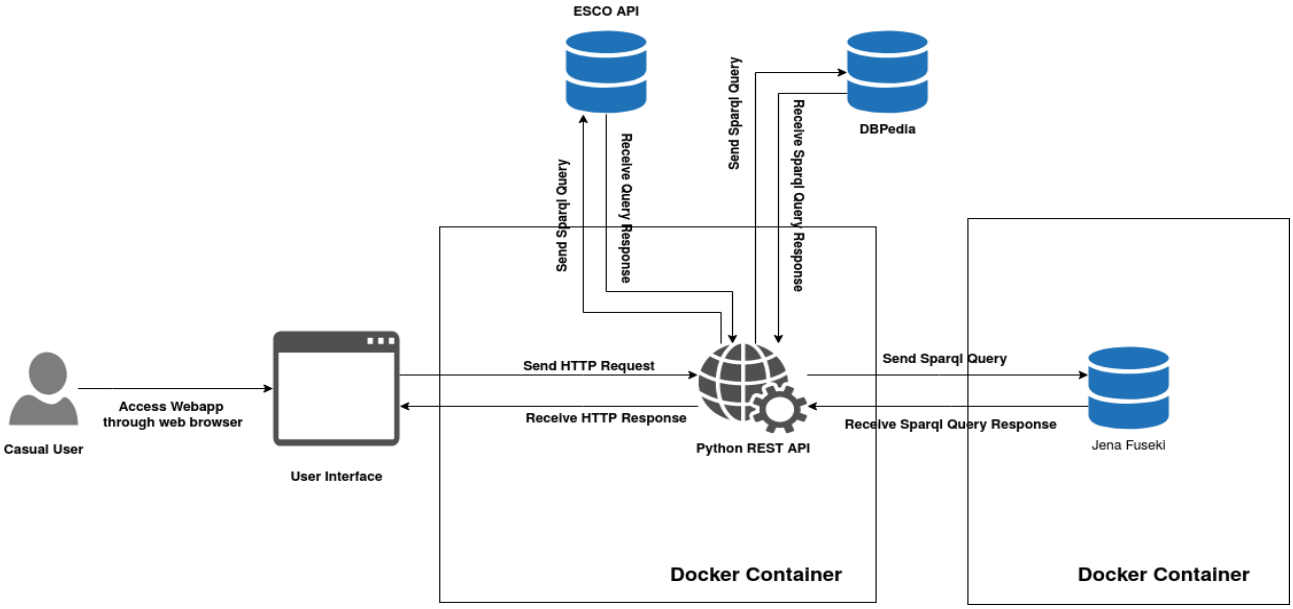

Figure 5.2.1: RDF2Résumé's architecture diagram

The architecture is based on the following main events:

- The user is able to access the application from a web browser
- The application sends HTTP requests and receives HTTP responses to and from an API, corresponding to the related events fired from the user.
- The API is able to send SPARQL [54] queries to a Jena Fuseki Triple store [3] or DBPedia [14][32] and get query responses back in order to process.
- The API is able to send HTTP requests to other APIs (e.g. ESCO API) and get HTTP responses back in order to process.
- The API is able to create Latex [25] files and convert them into pdf via the python library pdflatex.
- The API is able to create HTML files that semantically-annotated using RDFa [44].

The platform is implemented in a way to be easily deployable and run by using Docker [15] containers.

5.2.2 Logical Design

The logical design of our system pertains to an abstract representation of the data flows, inputs and outputs of the system. We use Activity and Sequence Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case.
An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.
A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together.

Below we will analyze how our platform will handle the main use cases described in section 5.1.2.

- Uploading a RDF file

  To understand the logic behind uploading the RDF file, please observe figure 5.2.2.

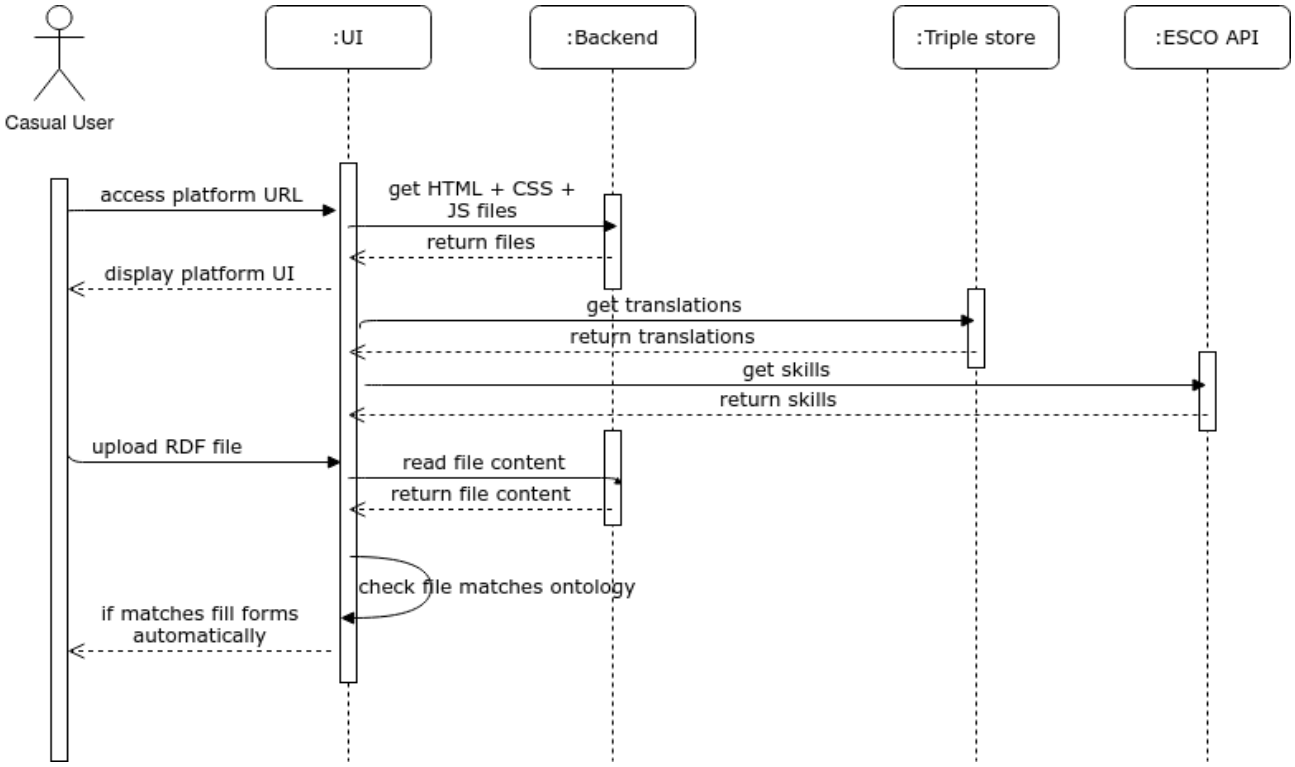

Figure 5.2.2: Sequence diagram of uploading an RDF file

A casual user accesses the platform via a web browser, and the backend REST API returns the appropriate HTML, CSS and Javascript files to be displayed. When the platform is run, the respective translations for titles and labels of form inputs are fetched from the Triple Store via SPARQL. The platform also fetches an array of skill names from the ESCO API, as clearly equally appointed to by the ontology.

After everything is loaded in the platform, the user can click the **upload RDF file** button and choose a file to upload. The file is then sent to the backend REST API for its content to be read. The content is returned to the UI of the platform that does

necessary checks to see if the file matches the ontology. If the latter is true, the form inputs are then filled automatically with the data available in the file. Now the user can update their information further or proceed with other features of the platform.

- Generating a simple résumé pdf

  When generating a simple résumé pdf, the user can follow two alternative flows as seen in 5.2.3. The user can either enter the text manually in the appropriate forms or upload an RDF file. In the second approach, additional checks will be made to detect whether the RDF format matches the ontology. If so, the relevant input forms are automatically filled. In either of the cases, after the user finishes adding/updating his data, he can generate a simple résumé pdf. More of the behind of the scenes of this process can be seen in figure 5.2.4.

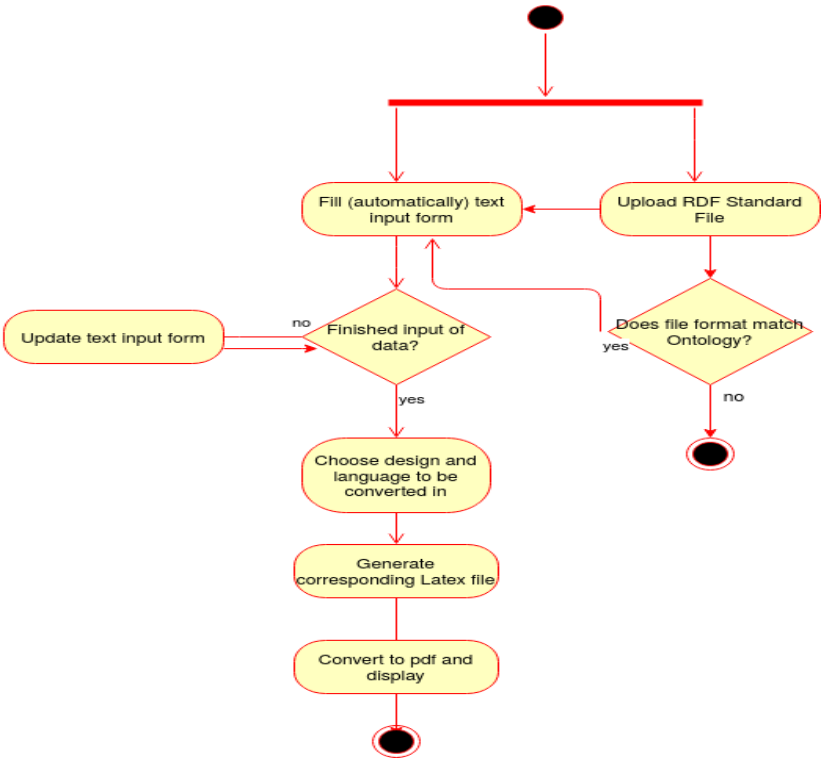

Figure 5.2.3: Activity diagram of generating a simple résumé pdf

After the user decides on the design and the language the pdf will be converted in, the backend is then in charge to do most of the processing. The REST API generates a Latex file that corresponds to the design picked by the user. While developing this file, the backend also fetches the appropriate translations from the triple store via SPARQL. After the file is created, and the content is written, the file is converted to pdf via a library. The generated PDF is returned to the UI to be displayed and presented to the user.

- Generating an enriched résumé pdf via a semantic-enabled process

  The process of generating an enriched résumé pdf is very closely alike the previous description. That can also be observed in the corresponding activity diagram figure

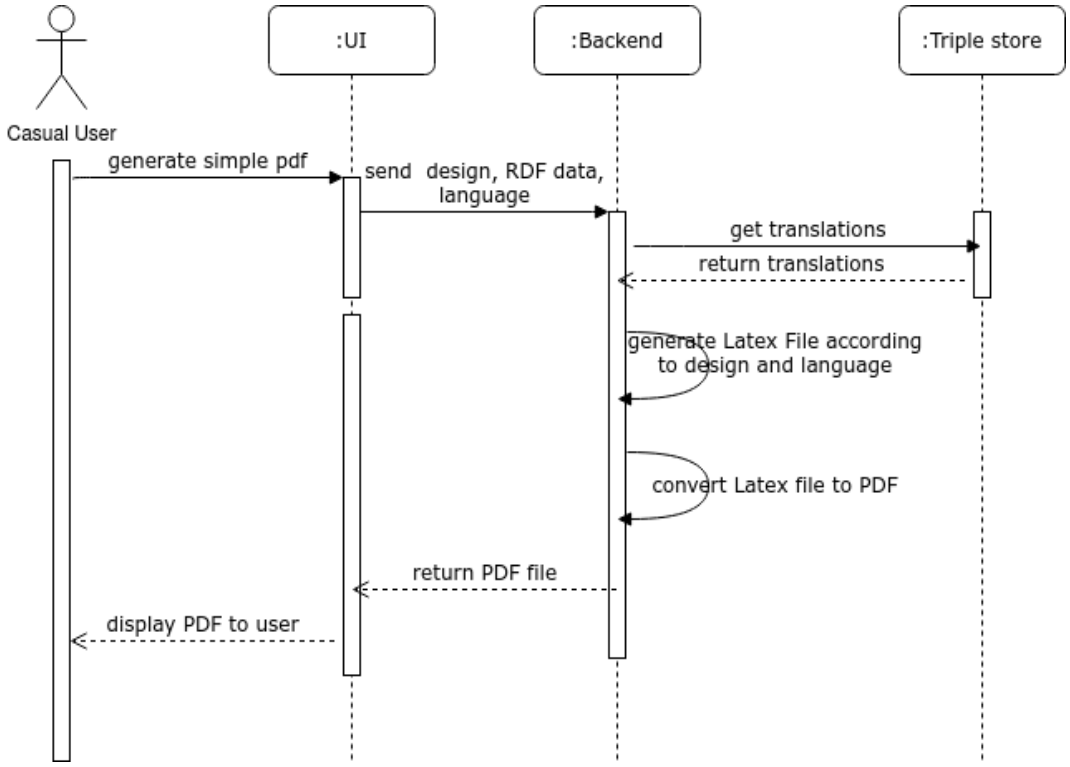

Figure 5.2.4: Sequence diagram of generating a simple résumé pdf

5.2.5 and sequence diagram figure 5.2.6. The most visible and fundamental change is the enrichment add-on process.

The enrichment part of the pdf derives from hyperlinking famous entities such as names of cities, countries or organization names to their respected Wikipedia pages. This process is accomplished by recognizing such entities and then utilizing SPARQL queries and DBPedia to gather such data. After the appropriate URLs are fetched, they are added during the process of generating the Latex file. The final converted pdf file is deemed enriched.

- Generating a semantically annotated personal website

  The use case of generating a semantically annotated personal website can also be split into two alternative flows. A user can enter all their CV related data manually or upload an RDF file as seen in figure 5.2.7. Needless to say, the file is checked to match the ontology. At the end of the data input, the user chooses the language the website will be converted in. It is the backend REST API's duty, as seen in figure 5.2.8, to then create the HTML file using the RDF data and also annotate it with RDFa attributes. At the end of the implementation, the REST API sends the files to the UI to be available for download.

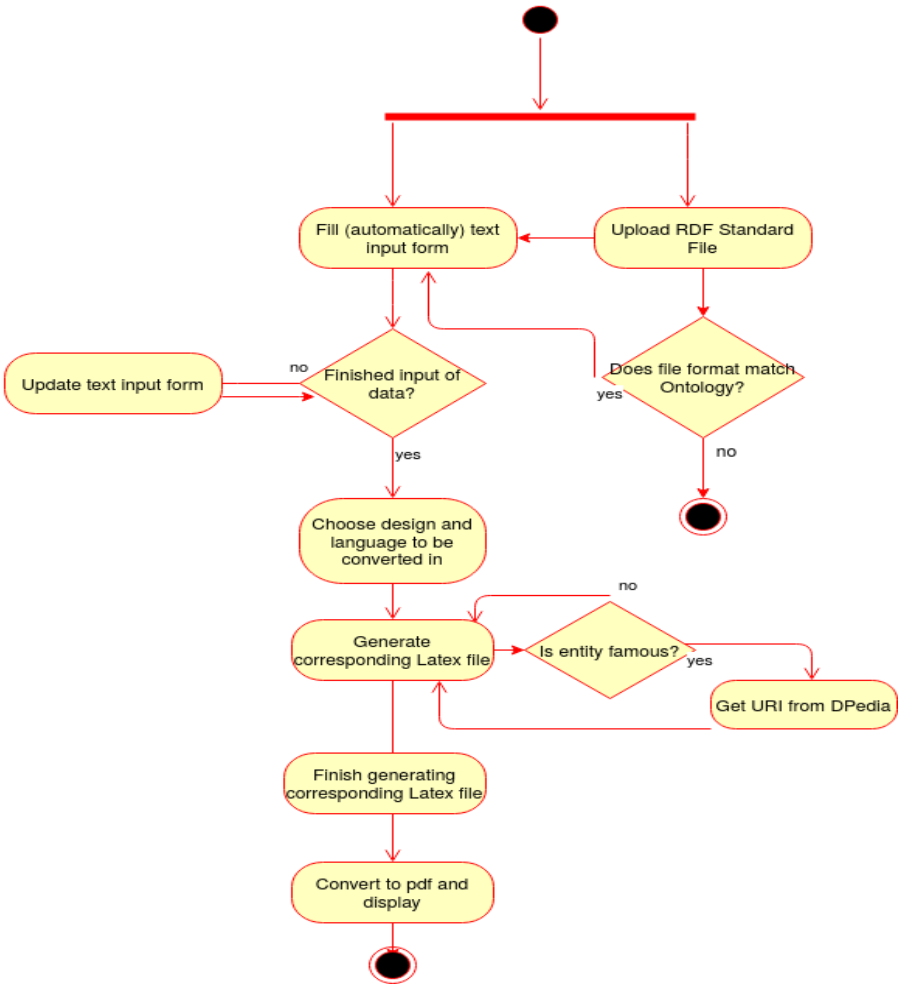

Figure 5.2.5: Activity diagram of generating an enriched résumé pdf

5.2.3 Physical Design/Implementation

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed.

Let us first comprehend how the user interface was designed and put together and then understand more in-depth the data flow in the platform and finally provide more technical details on the implementation of the logic of the system.

5.2.3.1 User Interface Design

User Interface Design is concerned with how users add information to the system and how the system presents the information back to them. Before starting the implementation of the UI, important decisions were made regarding the main components of the View and the reusability of those components for a modern responsive design and a user-friendly interaction. Components are used as they are the main building block when using ReactJS. The

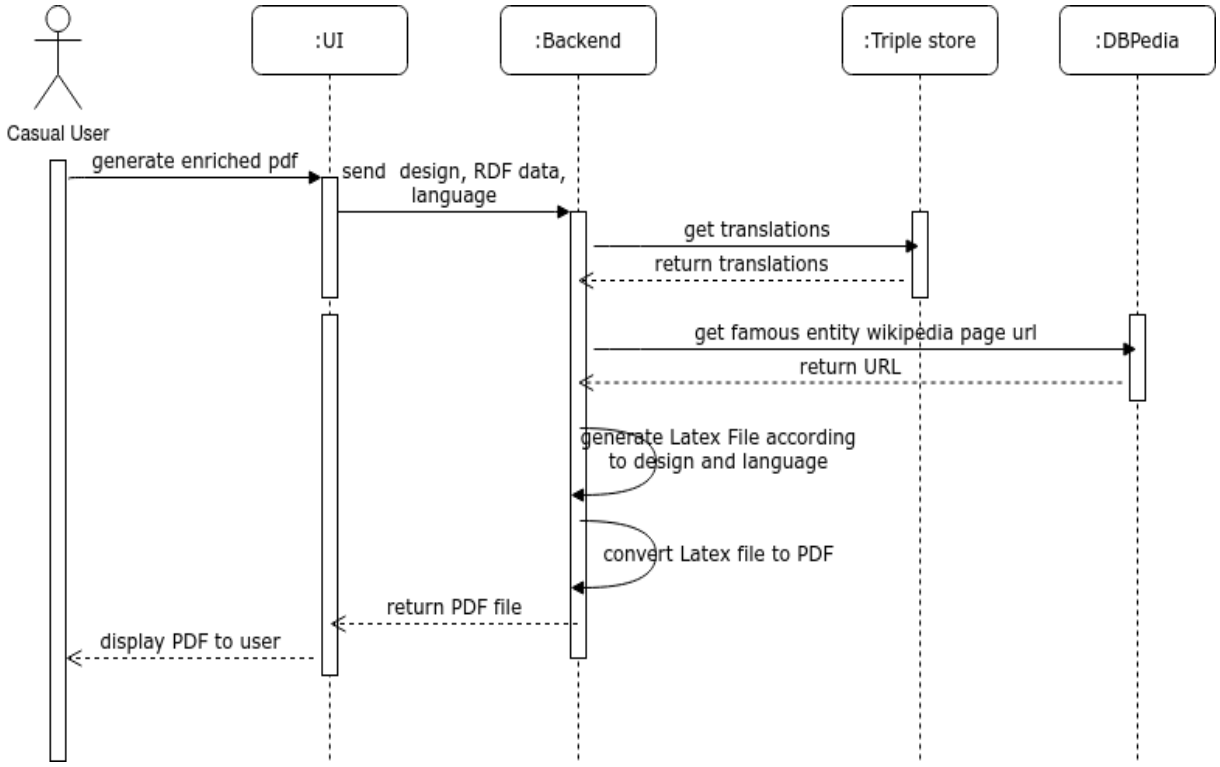

Figure 5.2.6: Sequence diagram of generating an enriched résumé pdf

proposed approach can be viewed in figure 5.2.9.

The primary component is the App Component that holds all other subcomponents. Further down, we can see the other important ones, specifically the Dashboard/Main Component and the Upload One. These are crucial components as they are directly correlated to their specific use cases of filling in forms and uploading an RDF file. More information about them can be found in the subsections below. The About Component is a simple component to hold static data about the purpose of the platform, its author and contacting information.

5.2.3.1.1 Dashboard/Main Component

Dashboard component is a larger one because it encapsulates other smaller components for each section of a CV, based upon the Résumé2RDF Ontology. A small glimpse of it can be seen in figure 5.2.10 and also the previous component diagram figure 5.2.9.

The smaller views/forms are as follows: About CV Form, Personal Information Form, Target Form, Work History Form, Education Form, Courses Form, Skills Form, Reference Form, Publication Form, Patent Form, Projects Form, Honor and Awards Form and Other Information Form.

The form inputs that can be filled, match the properties of the relevant Class in the Résumé2RDF Ontology. Further screenshots for all forms can be found in the Chapter 7. Forms that can contain multiple instances such as WorkHistory, Education, Courses and so on provide CRUD operations as well.

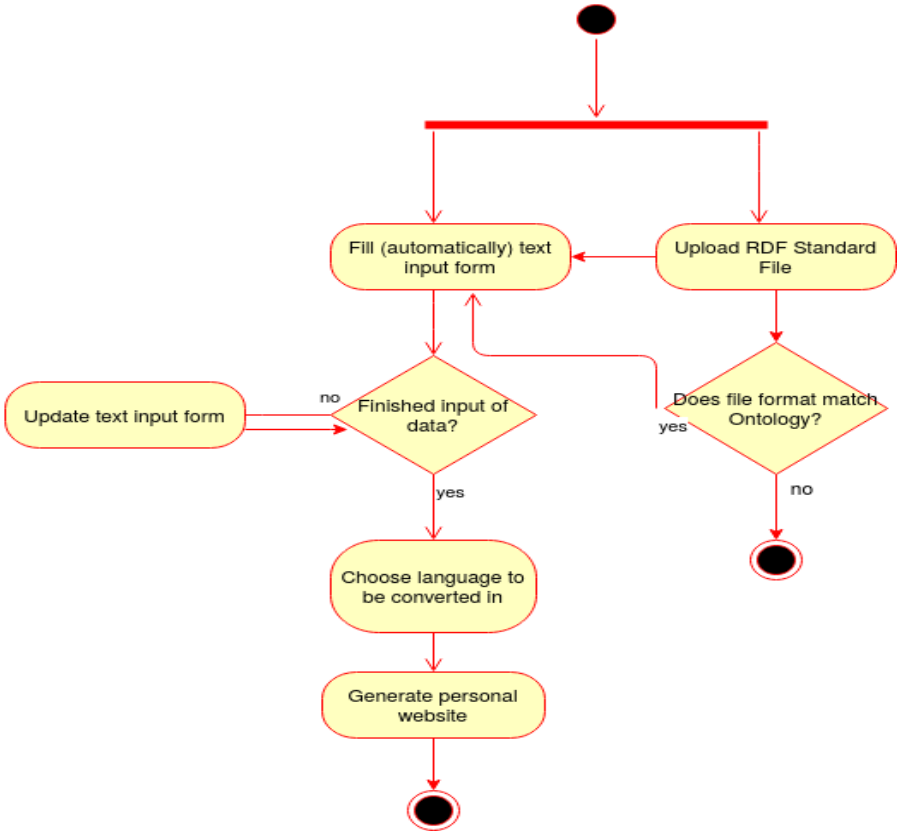

Figure 5.2.7: Activity diagram of generating a semantically-annotated personal website

In figure 5.2.11, we can observe the modal view that is displayed to the user when they want to add a new education to their CV. Just as the user is able to create a new instance, they are also able to edit or delete the specific instance and thus realizing the CRUD operations.

This view also allows changing the language the information is shown. This exciting feature, not only lets the user navigate through the app in the language specified, but also enables saving their CV data in that particular language as well. As a result, the generated outputs can be in different languages as well.

Another interesting subcomponent is the submit one. In this component, the user can finalize their actions by either downloading the JSON-LD file, generating RDFa or PDF or enriched PDF. One example of generating the PDF Résumé can be seen in figure 5.2.12.

5.2.3.1.2 Upload Component

This component lets you upload a JSON-LD type file that fits with the Résumé2RDF Ontology. A small glimpse of that can be seen in figure 5.2.13.

In the event where the file type does not match JSON, the person will not be able to upload the file. If the structure of the JSON-LD does not match the Résumé2RDF Ontology, then the form views previously shown, will not be automatically filled for further updating.

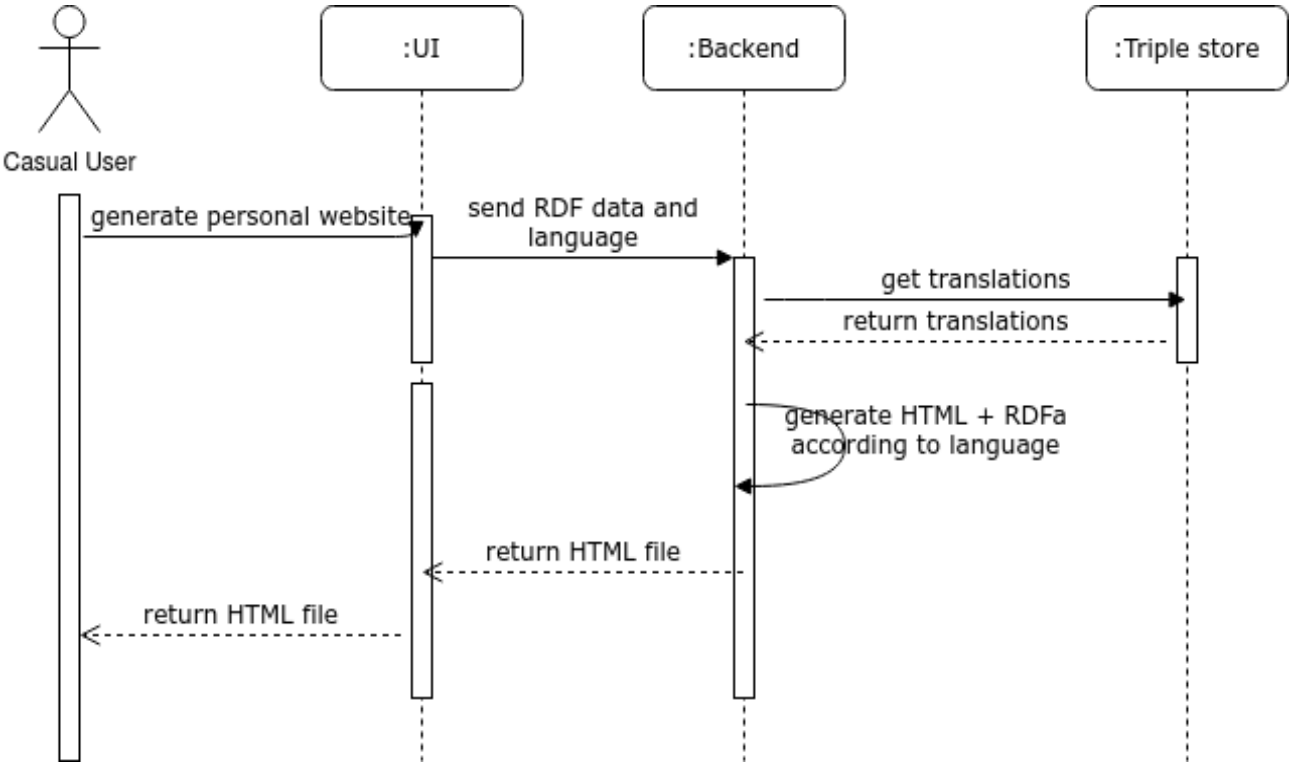

Figure 5.2.8: Sequence diagram of generating a semantically-annotated personal website

5.2.3.2 Data Design

Data Design is concerned with how the data is represented and stored within the system. The application is implemented using the Résumé2RDF ontology as the reference data structure, and below, we will specify the inputs and outputs possible, also seen in this general data flow figure 5.2.14.

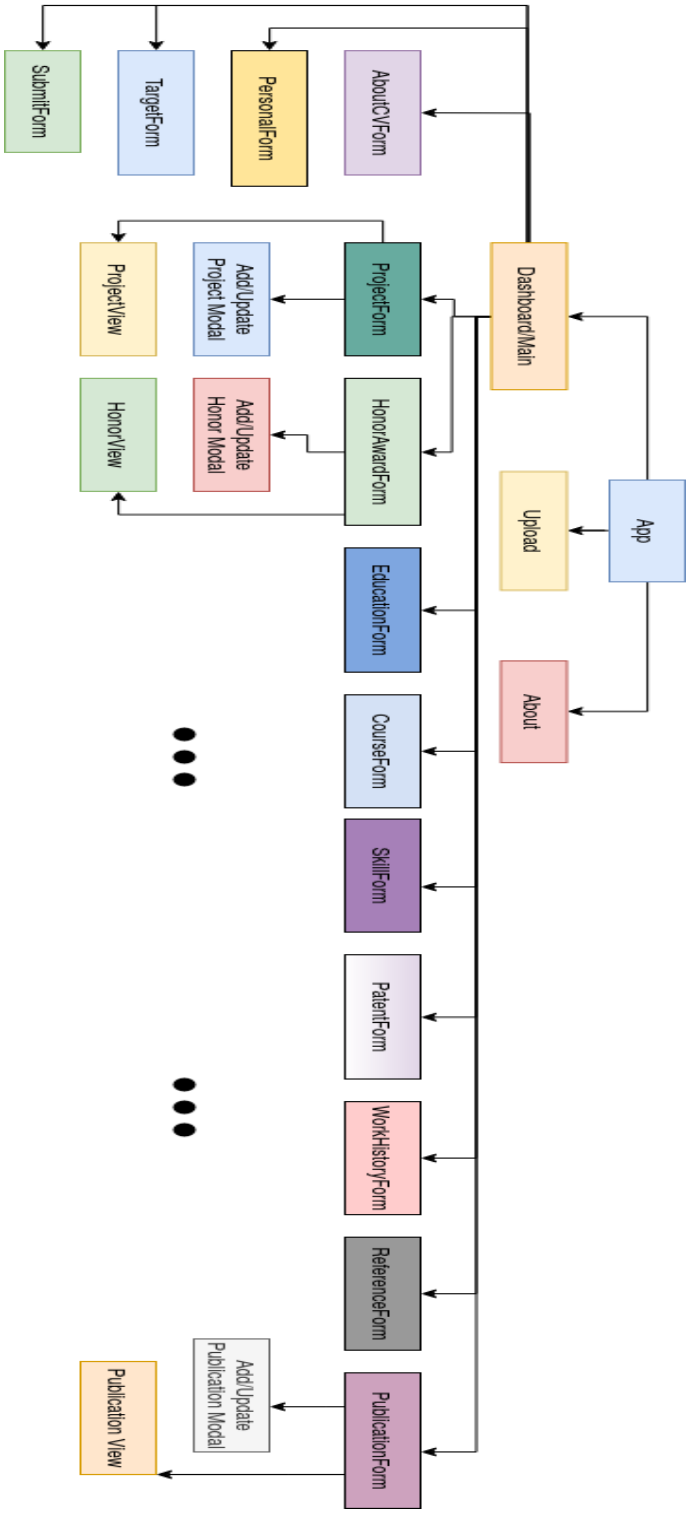

Figure 5.2.9: RDF2Résumé UI Component Diagram

Inputs of the application

- A user can write all their background information necessary manually for a résumé (as specified by the Résumé2RDF ontology) by using HTML forms.

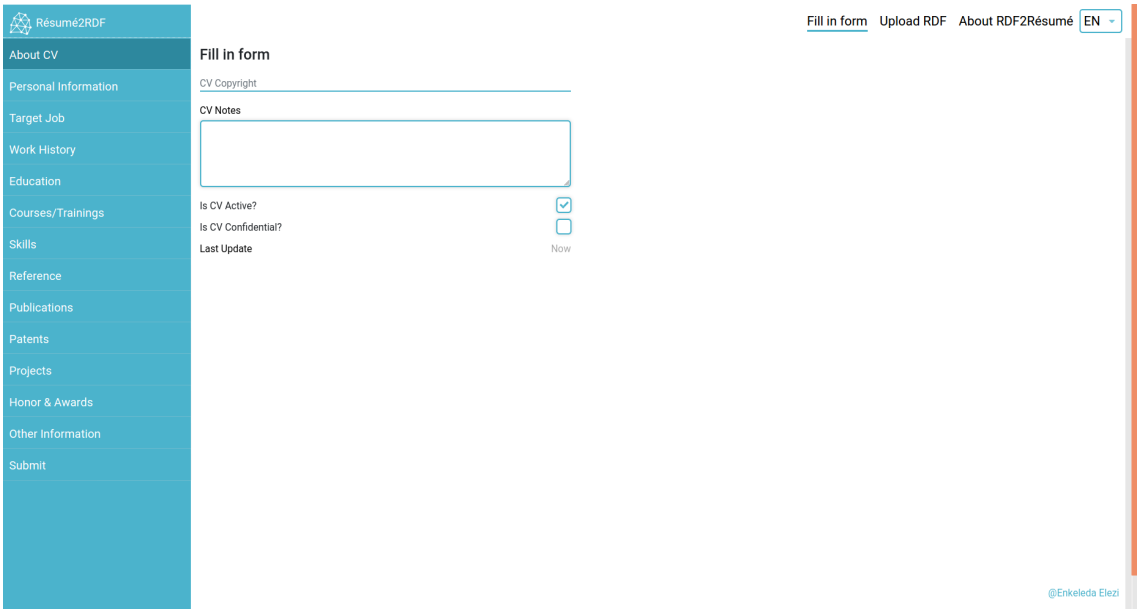

Figure 5.2.10: RDF2Résumé's Dashboard View

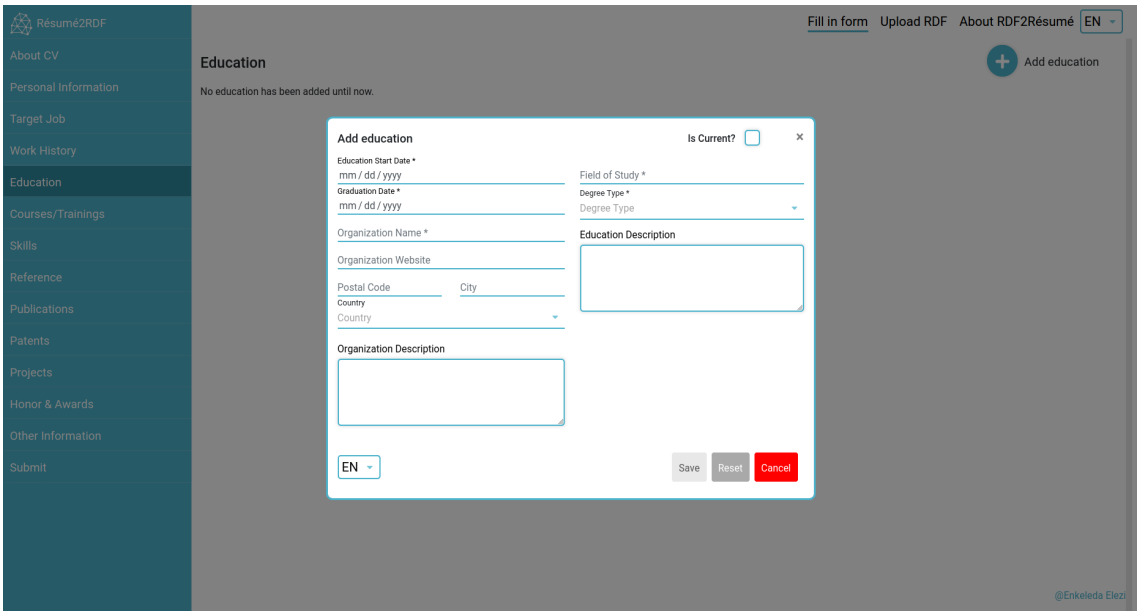

Figure 5.2.11: Modal view for adding a new education

- A user can upload a JSON-LD file. The file needs to match the Résumé2RDF Ontology in order for the values to fill the appropriate forms.
- A user must specify the language in which the PDF/RDFa needs to be generated in. The default language is English.

Outputs of the application*

- A user can download in a JSON-LD format all of the data, that he/she has entered up to that point. The file structure matches the Résumé2RDF Ontology.

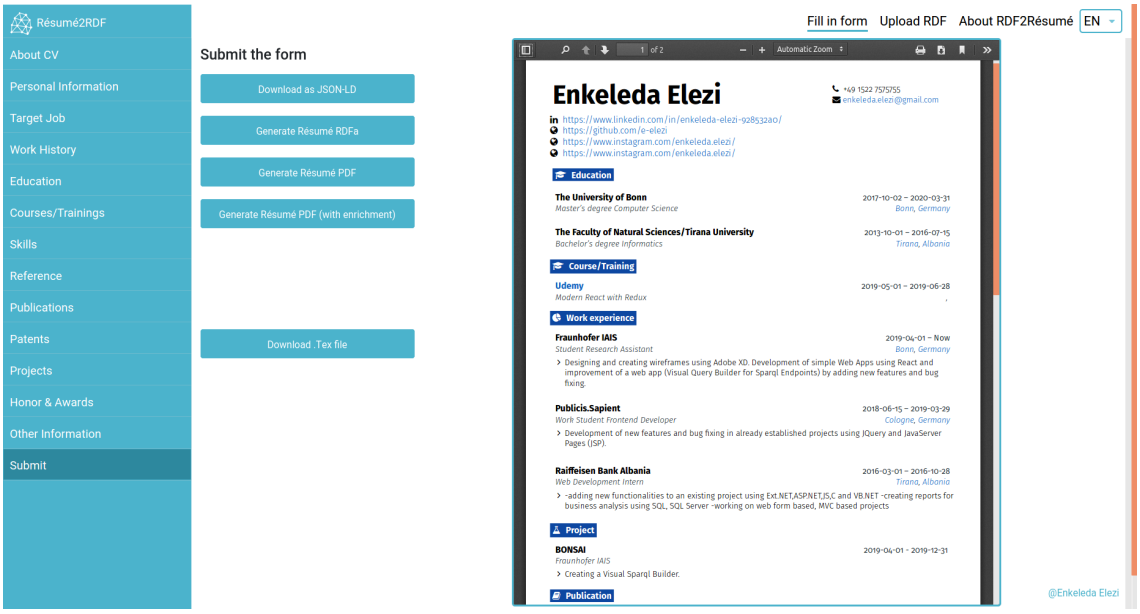

Figure 5.2.12: RDF2Résumé's Submit View

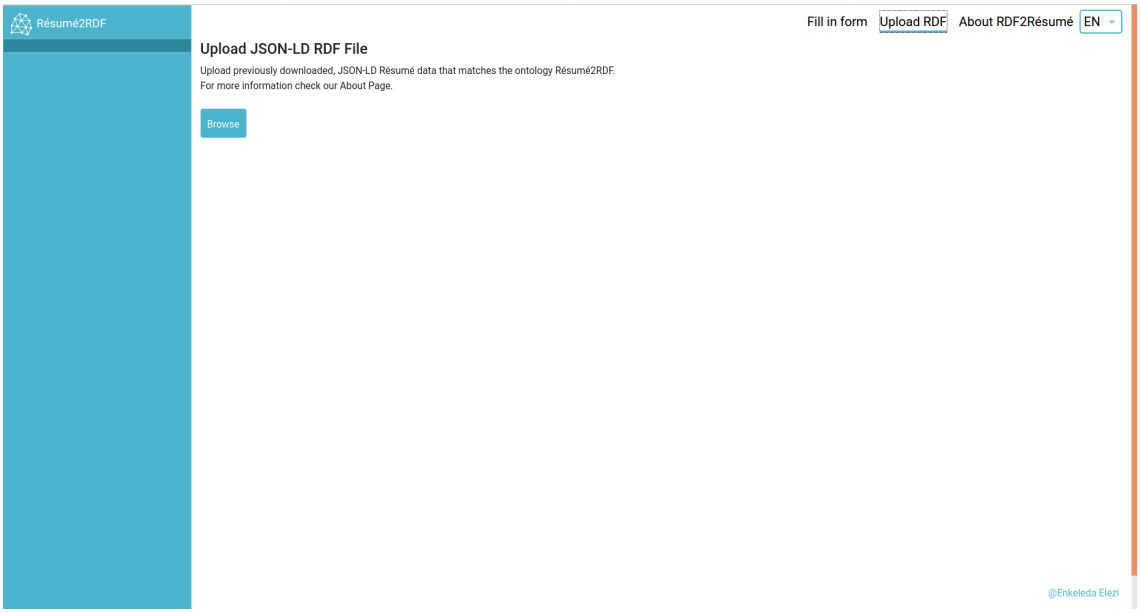

Figure 5.2.13: RDF2Résumé's Upload View

- A user can download a zip folder that contains a modern-looking HTML semantically annotated via RDFa, that they can use as their personal website.
- A user can download a résumé PDF, generated according to the data provided by them and also the accompanying Latex file for further personal edits.
- A user can download an enriched version of the résumé PDF. It is generated by adding URLs of famous named entities. The user can also download the accompanying Latex file for further personal edits.

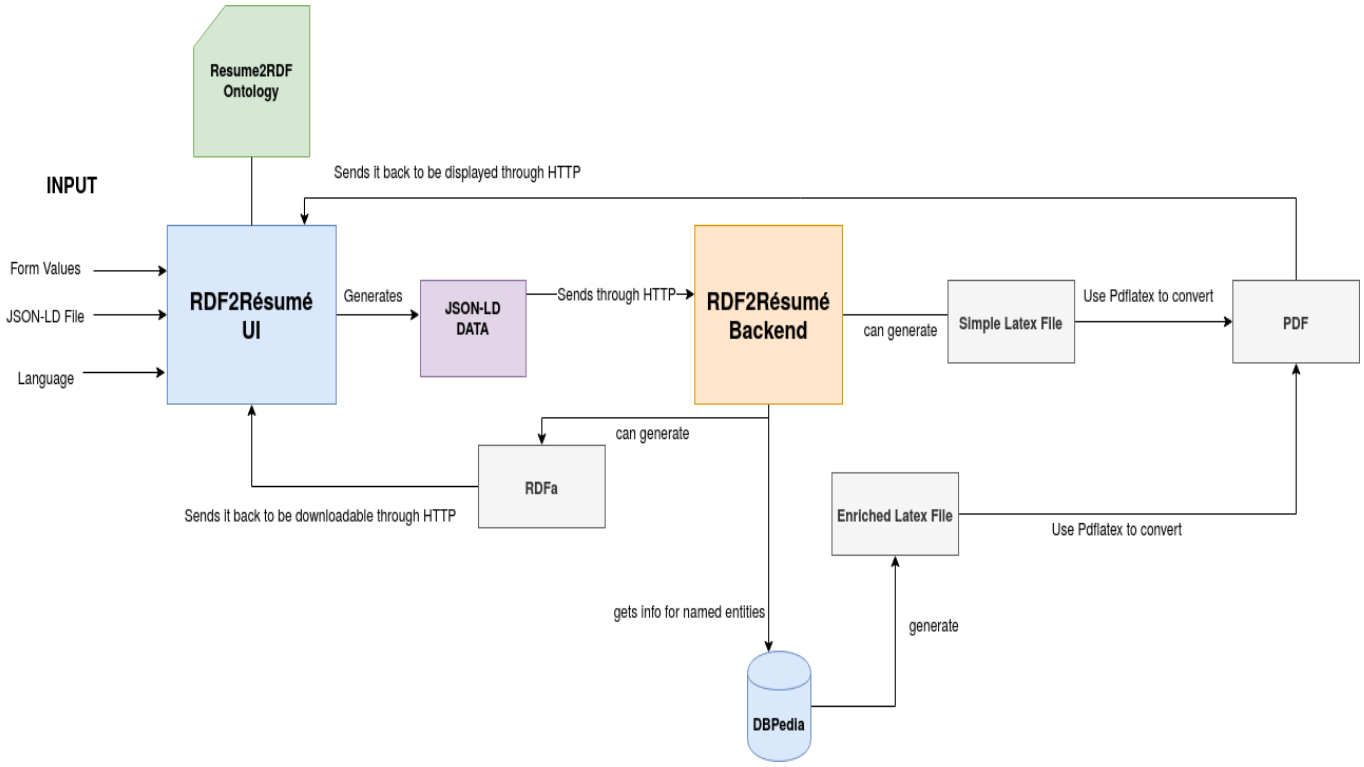

Figure 5.2.14: RDF2Résumé's Possible Data flow

*All downloads of the application can be available in the language specified by the user, taking into consideration that the input data in that language exists.

5.2.3.3 Process Design

Process Design is concerned with how the data moves through the system is transformed. The Python backend REST API is the one in charge of most of the processing done from the platform. The UI sends the form values in a JSON-LD format and then it is the backend's task to process them and output either an RDFa zip folder or a résumé PDF.

5.2.3.3.1 JSON-LD File

When a user saves the data in the platform, the saved format is in JSON-LD, and it matches the Résumé2RDF ontology. JSON is a lightweight data-interchange format, and its a common practice with web applications. JSON-LD is the method of encoding Linked Data using JSON. The user obtains the file in an RDF standard that can be uploaded for further edits later on. To understand the structure of the JSON-LD file, we can see how we define context and prefixes in listing 5.1. An example of how the data would be saved in the file, we can see the excerpt of a sample JSON-LD file in listing 5.2.

```
"@context": {
    "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
```

```
    "country": "http://www.bpiresearch.com/BPMO/2004/03/03/cdl/
   Countries#",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "my0": "http://example.com/rdf2resume_ontology.rdf#",
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
    "mybase0": "http://example.com/rdf2resume_base_ontology.rdf
   #"
  }
```

Listing 5.1: Defining context in JSON-LD

```
...
"my0:hasPublication": [{
    "@type": "my0:Publication",
    "my0:publicationTitle": [{
        "@value": "RDF2Résumé",
        "@language": "en"
      },
      {
        "@value": "RDF2Résumé",
        "@language": "de"
      },
      ...
    ],
    "my0:publicationPublisher": "University of Bonn",
    "my0:publicationDate": "2020-01-14",
    "my0:publicationAuthor": "Enkeleda Elezi",
    "my0:publicationURL": "",
    "my0:publicationDescription": [{
        "@value": "Paper about the Master thesis RDF2Résumé.",
        "@language": "en"
      },
      {
        "@value": "Arbeit über die Masterarbeit RDF2Résumé",
        "@language": "de"
      },
      ...
    ]
  }]
```

Listing 5.2: Excerpt of a Publication's details in a sample JSON-LD file

5.2.3.3.2 RDFa generation

Before the implementation of the RDFa generation, a specific design template was chosen. The key points for the template are a modern and responsive design and local Javascript and

CSS libraries/files. The CSS and Javascript files for this template are static files that will be part of the zip folder. The only file that is dynamically generated is the HTML index file.

The HTML design was also altered to include RDFa attributes. An excerpt HTML code that is semantically annotated using RDFa can be viewed in figure 5.2.15 .

```
<div typeof="my0:WorkHistory" class="resume-item col-md-6 col-sm-12 col-lg-8">
    <div class="card mx-0 p-4 mb-5"
        style="border-color: #2598f3; box-shadow: 2px 2px 2px rgba(0, 0, 0, 0.21);">
        <div class="resume-content mr-auto">
            <h4 class="mb-3"><i class="fa fa-globe mr-3 text-primary"></i><span property="my0:jobTitle">
                    Web Development Intern</span><span property="my0:jobType"
                    content="http://example.com/resume2rdf value ontology.rdf#Intern"> (Intern)</span>
            </h4>
            <h4 class="mb-3"><i class="fa fa-map-marker mr-3" style="color: #074679;"></i><span
                    rel="my0:employedIn" typeOf="my0:Company">
                    <span property="my0:orgName" content="Raiffeisen Bank Albania">Raiffeisen Bank
                        Albania</span>
                    <span rel="my0:orgAddress" typeOf="my0:Address">
                        <span property="my0:city" content="Tirana"> - Tirana</span><span
                            property="my0:country"
                            content="http://www.bpiresearch.com/BPMO/2004/03/03/cdl/Countries#Albania">,
                            Albania</span></span>
            </h4>
            <p property="my0:jobDescription">-adding new functionalities to an existing project using
                Ext.NET,ASP.NET,JS,C# and VB.NET
                -creating reports for business analysis using SQL, SQL Server
                -working on web form based, MVC based projects</p>
        </div>

        <div class="resume-date text-md-right">
            <span class="text-primary">
                <span property="my0:startDate" content="2016-03-01">03/2016</span> - <span
                    property="my0:endDate" content="2016-10-28">10/2016</span>
                <span property="my0:isCurrent" hidden content="False"></span></span>
        </div>

    </div>
</div>
```

Figure 5.2.15: HTML+RDFa Work History Excerpt

For classes we use the attribute **typeof**, for relations we use the attribute **rel** and for properties we use **property**. Sometimes, for correctly specifying the content of the property, we define the attribute **content**; otherwise, by default, the value of the property equals the text inside the tag. To specify the vocabularies used, we include the **prefix** attribute with its relevant values in the body element of the HTML. Example of such website generation can be seen in figure 5.2.16.

The HTML file is generated after sessions of concatenating strings that hold data about specific sections of the résumé. The prior decision for a specific template helps in updating only certain parts of the HTML that need to be dynamic such as the name of the person, their work history and so on (input data).

After the HTML file is created, the file plus the CSS and Javascript files are zipped and then the zipped file is sent for download to the user. The files provided are easy to publish in the world wide web.

To verify the semantic annotations present in the HTML (that should correspond to the Résumé2RDF ontology) we used RDFa playground[3] for testing purposes.

[3]http://rdfa.info/play/

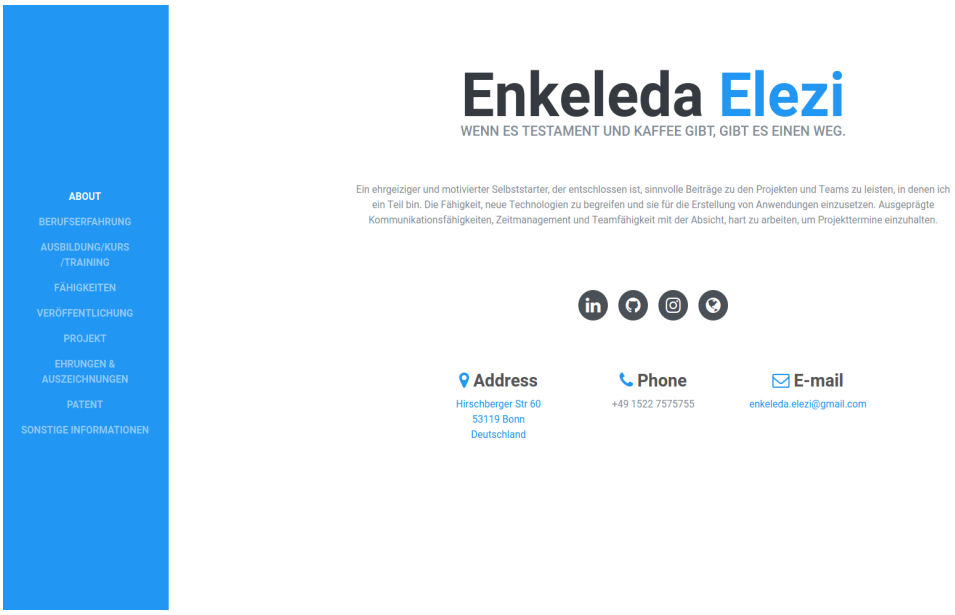

Figure 5.2.16: RDF2Résumé's RDFa Generation Example

5.2.3.3.3 PDF generation & enrichment

Before the implementation of the feature of generating PDF, two specific Latex templates were chosen. Their designs vary from simple to a bit more modern. Nonetheless, adding more latex templates will help to improve the platform further.

After the input data is received, the platform generates a Latex file that uses the template as a base source, and the dynamic values correspond to the input received. The process is similar to the one above of concatenating strings. After the creation of the Latex file, the python server runs the library **pdflatex** to convert it to pdf and then displays the pdf to the user. This main function can be viewed in listing 5.3.

```
def writeJSONtoTEX(data, filename, designNumber, language):
  main = r''''''

  if(designNumber==0):
    main = generateMainDesign1(data, language)

  if(designNumber==1):
    main = generateMainDesign2(data, language)

  content = main + footer

  completeName = os.path.join('build/static/media/pdf', filename)
  with open(completeName + '.tex','w') as f:
    f.write(content)

  cmd = ['pdflatex', '-interaction', 'nonstopmode', completeName + '.tex']
  proc = subprocess.Popen(cmd)
  proc.communicate()

  os.unlink( filename + '.aux')
  os.unlink( filename + '.log')
  os.system("mv " + filename + ".pdf build/static/media/pdf")
  return f
```

Listing 5.3: Main Latex and PDF generating function

For the generation of an enriched PDF, the same process follows. However, before the writing of the dynamic data, a query is sent to DBPedia via http://dbpedia.org/sparql to get the URI of a famous named entity via the property foaf:isPrimaryTopicOf, and then the corresponding element is set as an anchor link in the PDF. The URI is the URL of that entity's Wikipedia's page. Currently, the entities queried are the city of an address, the country of an address and an organization's name. The query is shown in listing 5.4. ObjectName and the language are two arguments provided before running the query. In Python, one can run SPARQL queries by using the SPARQLWrapper library.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?d WHERE {
    ?entity rdfs:label "ObjectName"@lang.
    ?entity foaf:isPrimaryTopicOf ?d.
}
```

Listing 5.4: SPARQL Query DBpedia linking

5.2.3.3.4 Output in different languages

As already mentioned before, the user is provided with the option of choosing between five languages to work in, English, French, German, Italian and Albanian. This is possible because the Résumé2RDF Ontology itself includes labels of each entity in each of these languages.

After the user picks the language they want their CV to be generated in; a SPARQL query is sent to the Fuseki triple store, to retrieve the correct translation of that entity's label in that specific language. The query is shown in listing 5.5. ObjectName and language are the two arguments needed before running the query.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT  ?o WHERE {
    <objectName> rdfs:label ?o.
    FILTER(lang(?o) = "lang")
}
```

Listing 5.5: SPARQL Query label in specific language

5.2.4 Technologies

As far as the UI is concerned, the platform is built upon ReactJS 16.8.6 [45], Font Awesome for icons [21], Redux 4.0.1 [46] for state management, SweetAlert2 [55] javascript library for more user friendly alerts and Axios [5] for AJAX HTTP Requests. For the backend, we used Python 3 [35] and installed the following python libraries: flask, flask_cors, SPARQLWrapper and pdflatex. To facilitate the deployable aspect, we used Docker [15] containers.

Chapter 6

Conclusion & Future Work

This thesis discussed the idea of standardization in the domain of recruitment and, more specifically, résumés. Résumé parsers used in software such as Applicant Tracking Systems, do not capture a high percentage of information in résumés because of the lack of a proper structure. Résumés written in a standard format would be more likely to be correctly interpreted by parsers in ATS. The standardization of résumés and the improvement of the recruitment process can be achieved by using semantic web technologies because they add domain based semantic information in order to be easily discoverable and reusable by machines, as well as humans.

The motivation of using semantic web technologies for résumés originated in our research questions. In the related work chapter, we explored further to answer the questions. During the research, we learned that there are existing CV models such as Europass and even an ontology that describes the information inside a Curriculum Vitae named ResumeRDF. Unfortunately, the only-known resume ontology has not been kept to date and dates back to 2007. Thus we updated and improved it after comparative research between ResumeRDF, Europass, Linkedin, and Xing.

As there are no available platforms or systems at the moment that generate résumés or personal website through a semantic-enabled process or semantic annotations, our other thesis contributions are precisely such a platform. After reaching the final version of the ontology, we implemented the RDF2Résumé platform based on it, for writing and updating a CV. The platform provides features for generating personal websites that semantically annotated via RDFa and enriched résumé PDFs with DBpedia.

Currently, the approach relies on self-declarations of competences and experiences or the uploading of a previously self-declared CV according to the domain ontology. Of course, this is tedious work for the users. It would be interesting for the system to be able to upload CVs in other formats such as .word or .pdf and extract as much data as possible matching with the domain ontology. Then the user could continue to update relevant information and proceed with the other features the platform provides. Kopparapu of TCS Innovations lab [24], developed a system for automated resume information extraction to support rapid resume search and management which is capable of extracting several important informative fields from a free format resume using a set of natural language processing (NLP) techniques.

Furthermore, RDF2Résumé gives a developer the potential to quickly write code and make interlinking possible with other platforms other than DBPedia, such as DBLP for publications or EPO registry for patents. It is evident that making this possible would also improve the platform further.

Something else interesting that can be taken into account for future development is linking this job-seeking platform automatically with other available job offer platforms. Using ontologies can be directly used to make applications more aware of the domain semantics, such as matching an individual's competency profile with a requirements profile. Makwana et. al [48] proposed a system to help recruiters find the appropriate candidate for the job post using semantic analysis.

Finally, as the platform is mostly in its beta form, further testing and usability studies are required to consolidate it as the platform for semantic-enabled résumé generator. The System Usability Scale (SUS) and the Post-Study System Usability Questionnaire (PSSUQ) are popular and respected psychometric questionnaires for measuring the user experience. Unfortunately, running a usability study was not included in this thesis. Defining the ontology and implementing the platform from scratch required much work and time. Our lack of expertise in the very domain of usability study also contributed to the decision of not running the study.

Repository: https://github.com/e-elezi/rdf2resume

Chapter 7

Appendix - A step by step user experience

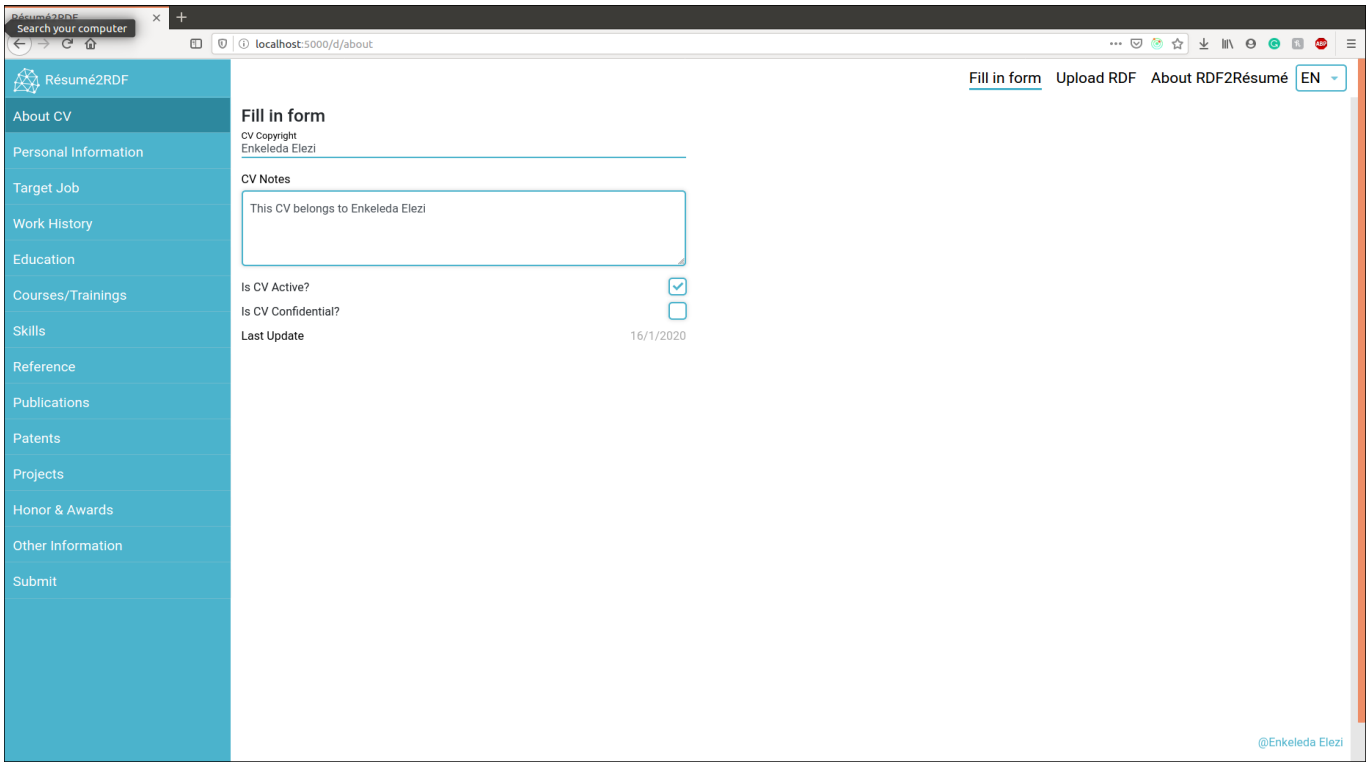

Figure 7.0.1: This is the first form the user is confronted with. The form corresponds to the properties of the CV Class, such as copyright, notes, last updated, and if the CV is still active and confidential.

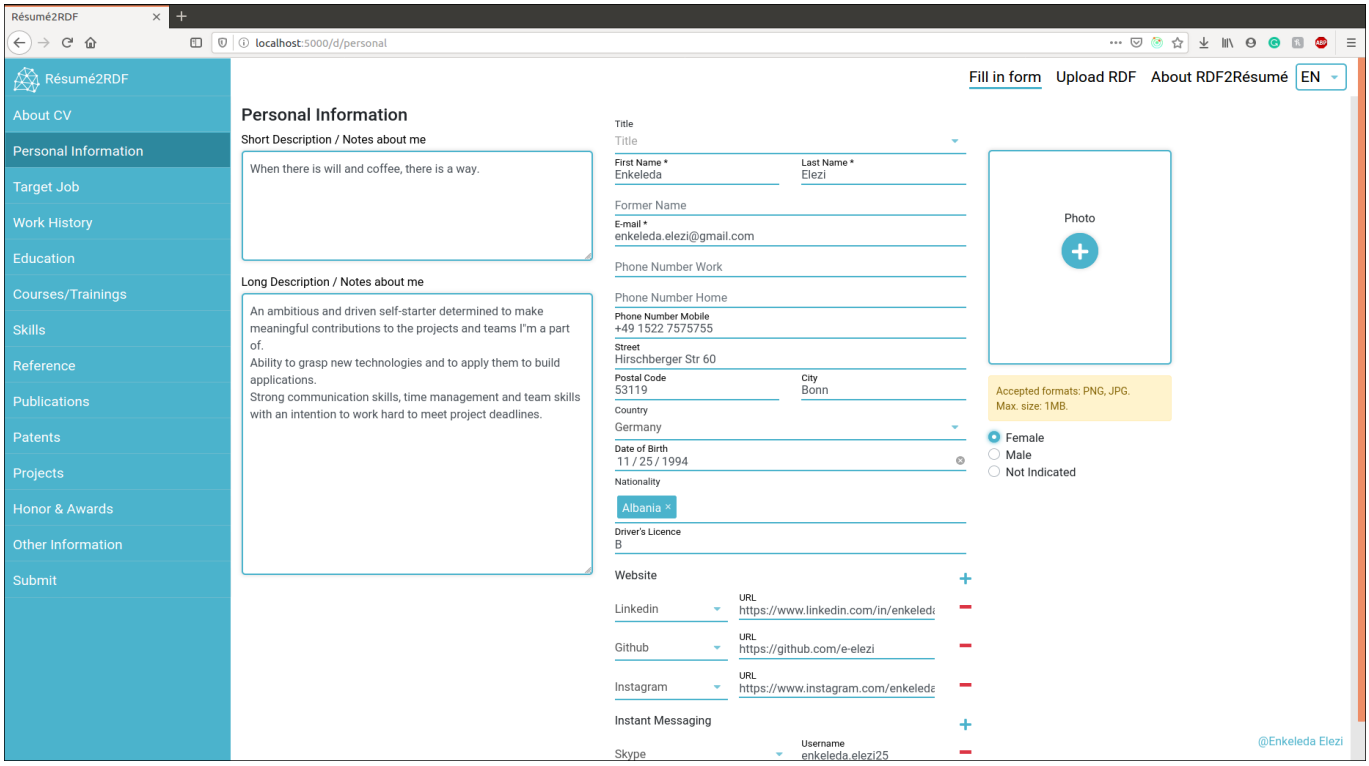

Figure 7.0.2: The second form allows us to fill in personal information regarding the person who owns the CV. The fields correspond to the properties of the class Person.

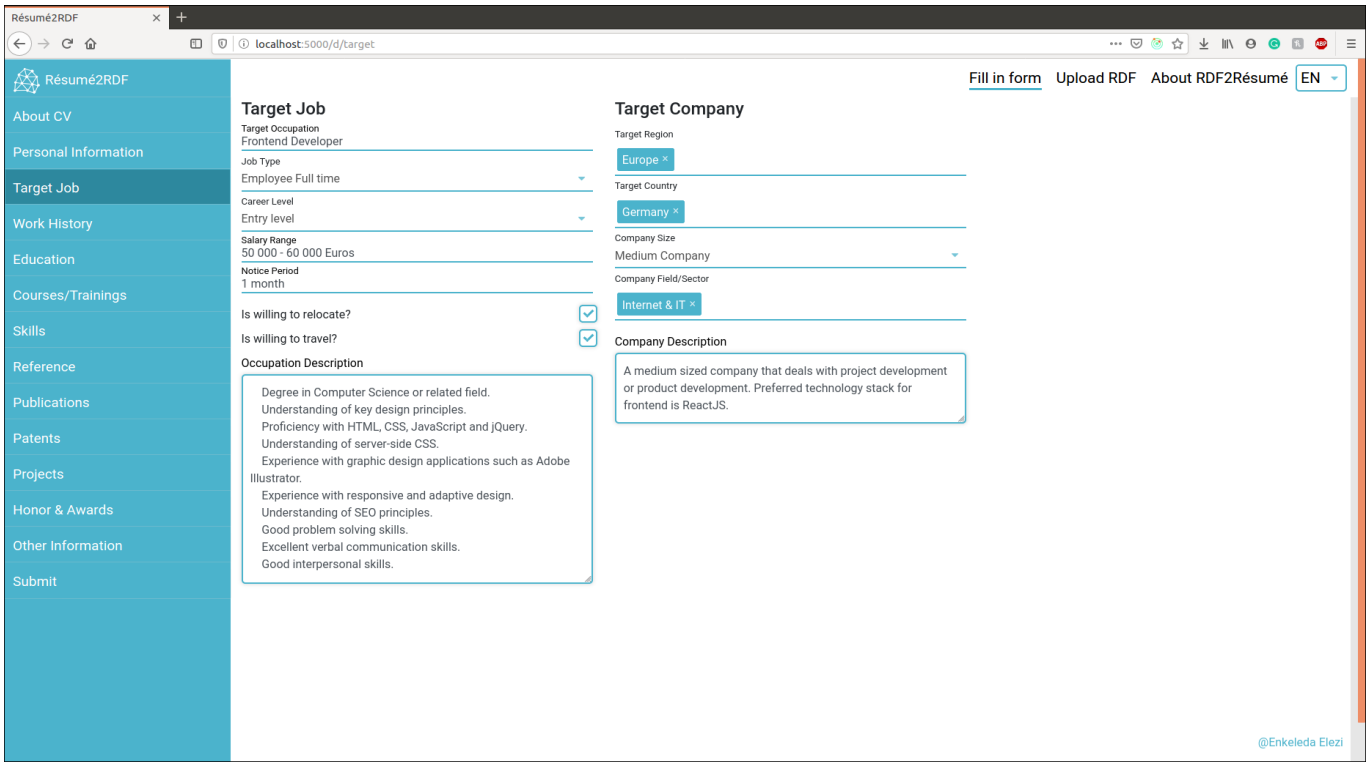

Figure 7.0.3: This form describes the target desired occupation. The form allows defining information for the desired company as well.

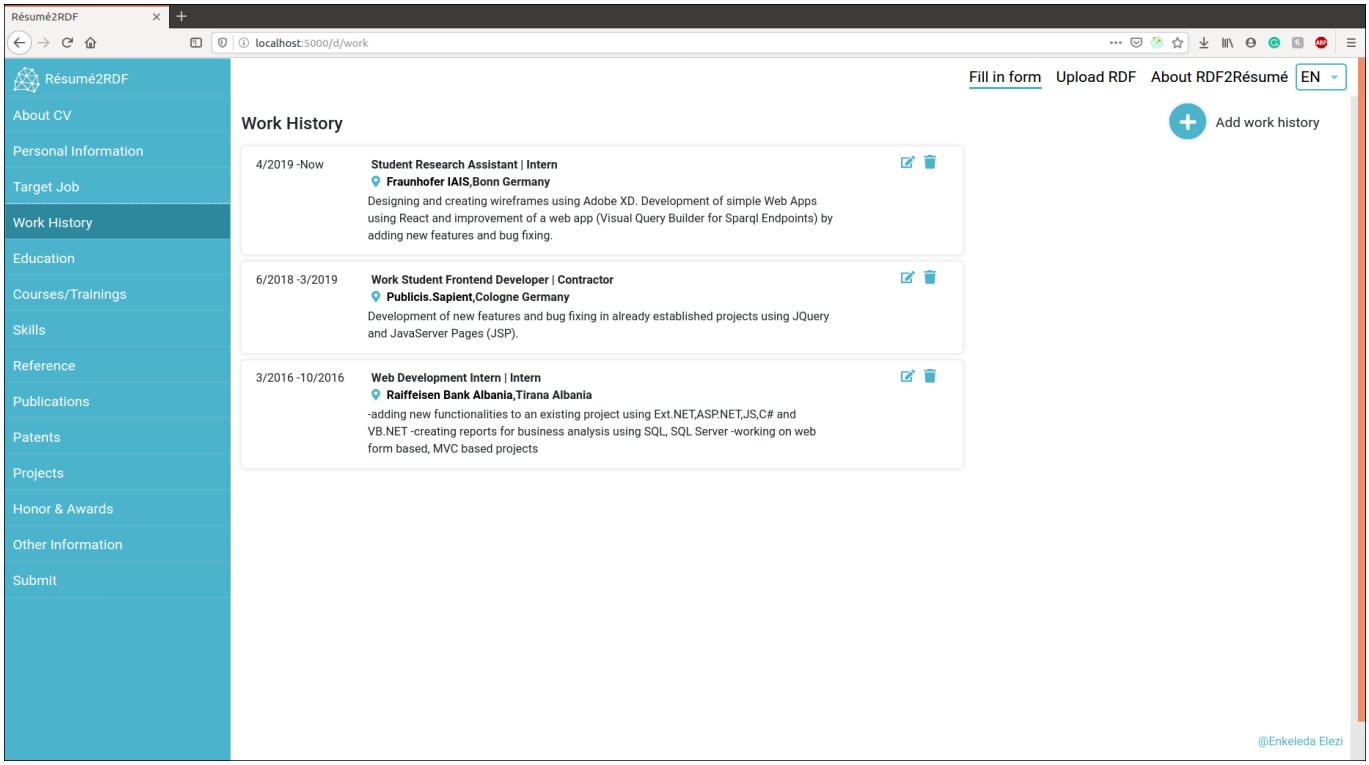

Figure 7.0.4: The Work History form is the first form that provides CRUD functionalities. A user can add a work experience, list them, and have the option to edit or remove specific ones.

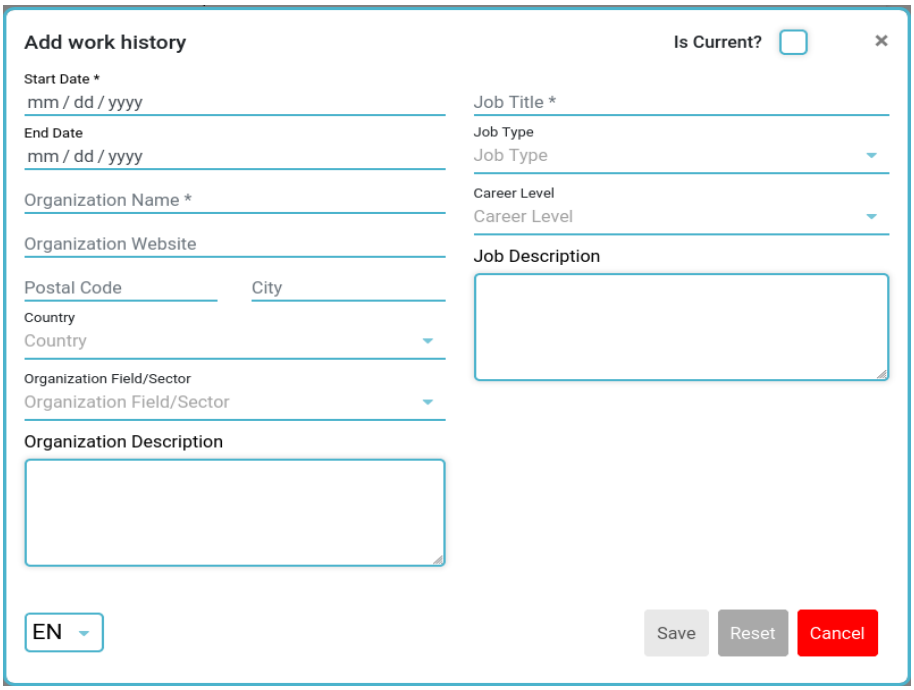

Figure 7.0.5: This modal form provides the user with the ability to fill in information about a work experience. The fields match the Work History Class properties.

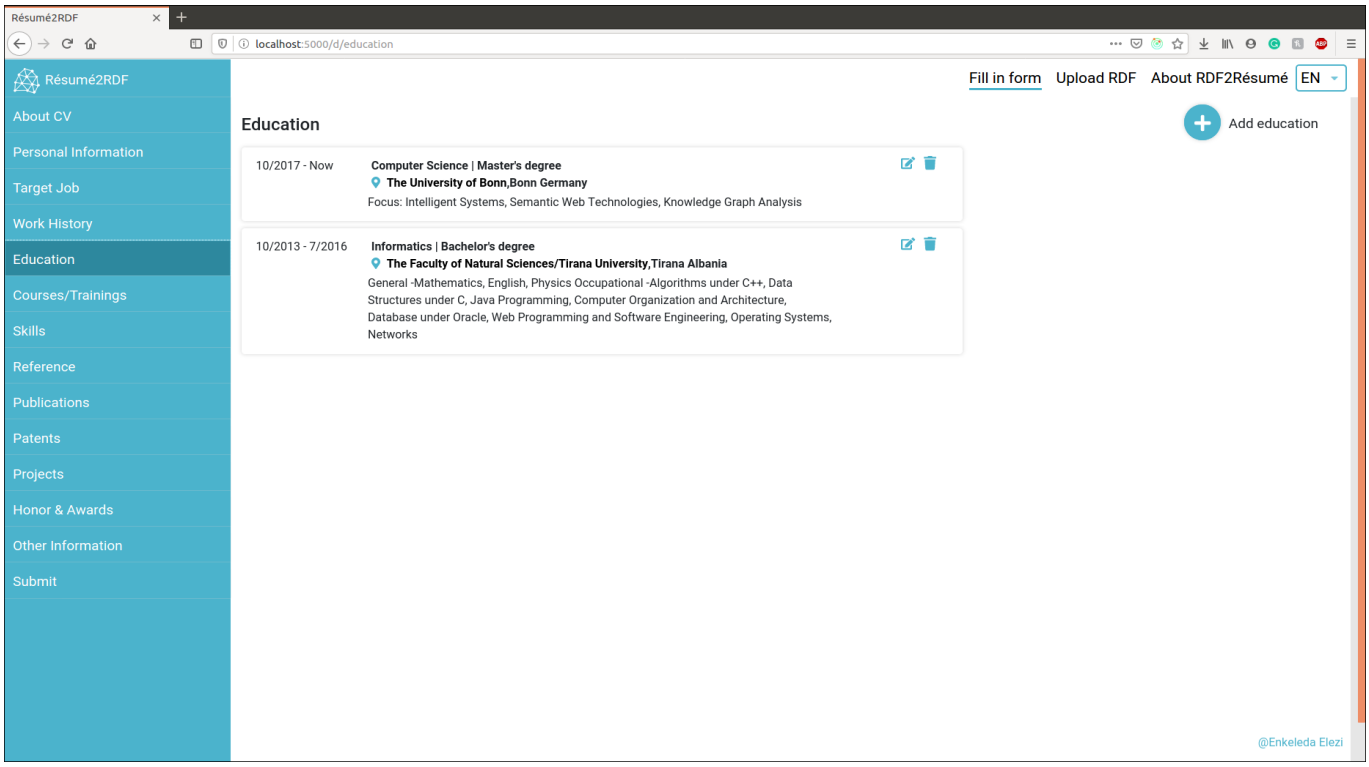

Figure 7.0.6: The Education form defines background information regarding the education of the person. The form also provides CRUD functionalities to add, edit, or remove an experience.

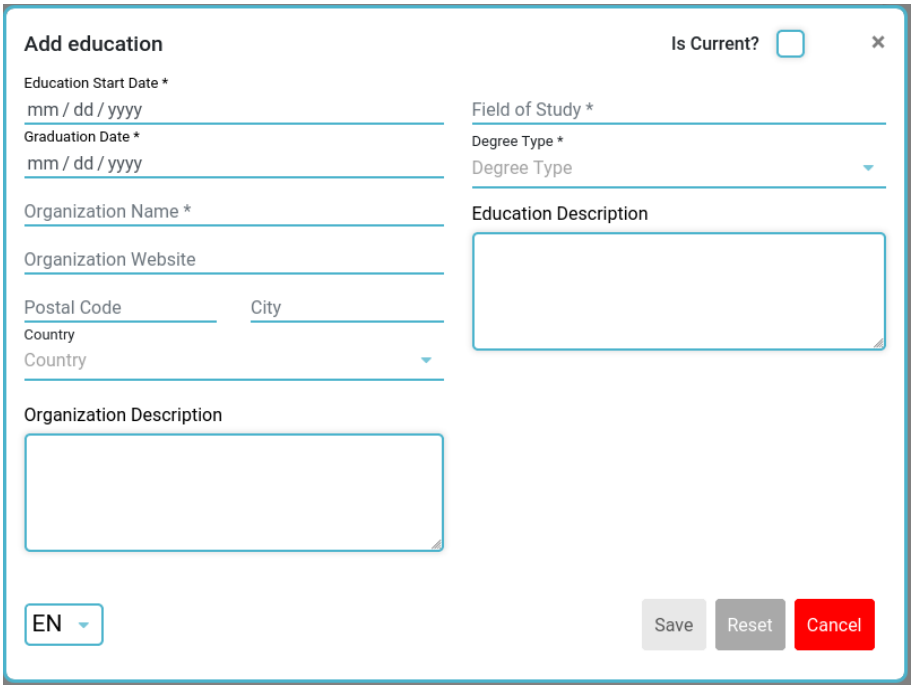

Figure 7.0.7: The modal form of Education provides the fields that match the Education Class's properties.

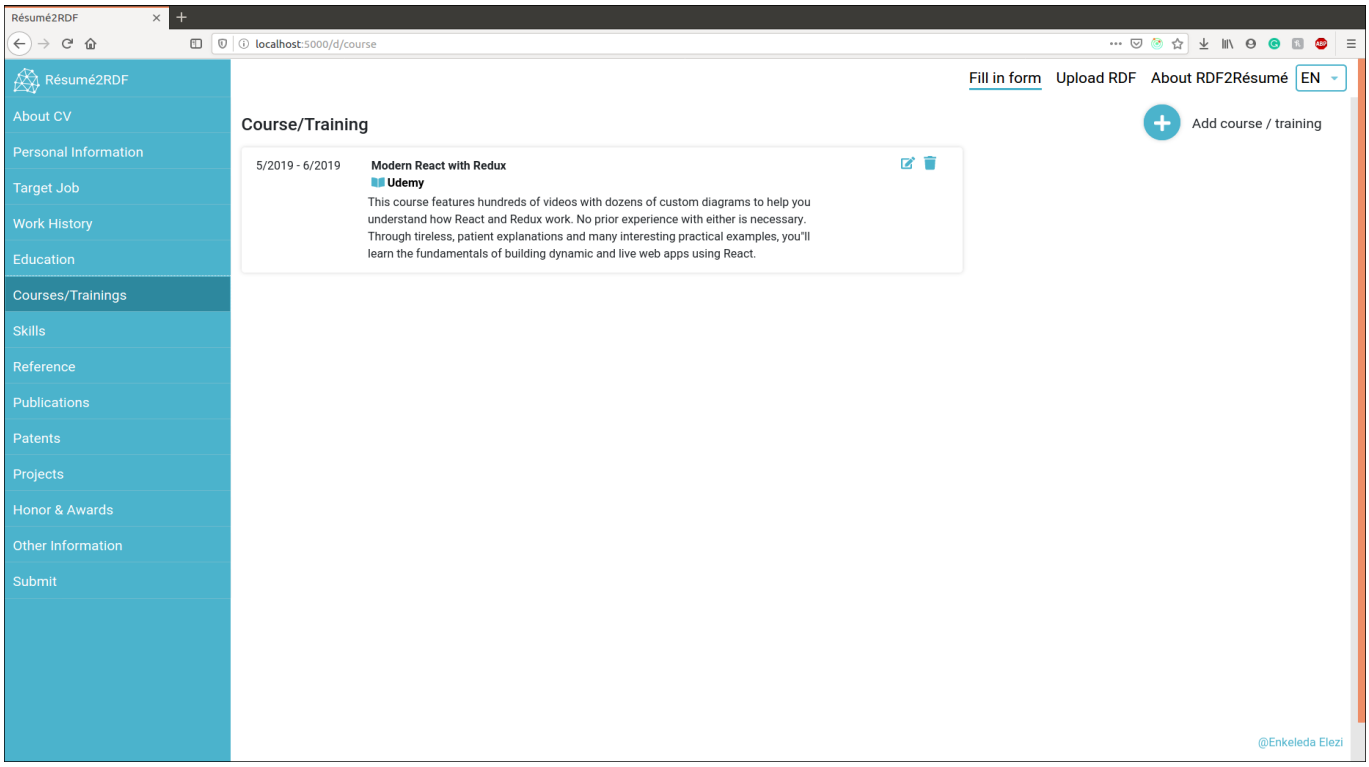

Figure 7.0.8: The Course form, similar to the Education one, also provides information that is related to education but, for a specific course. The user can add, edit, or delete a course and its data.

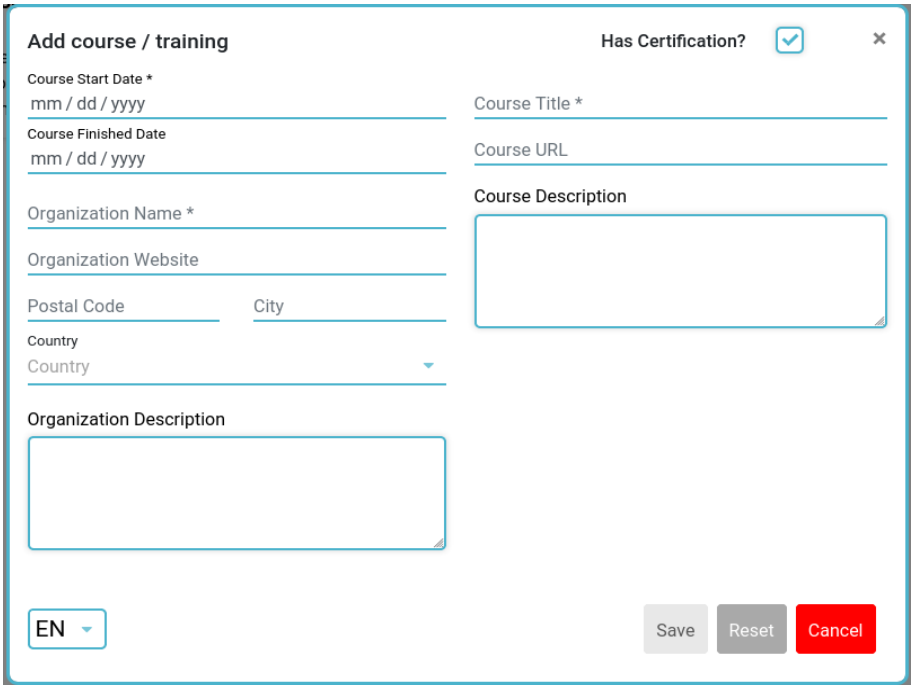

Figure 7.0.9: The fields of the modal form match the Course Class's properties.

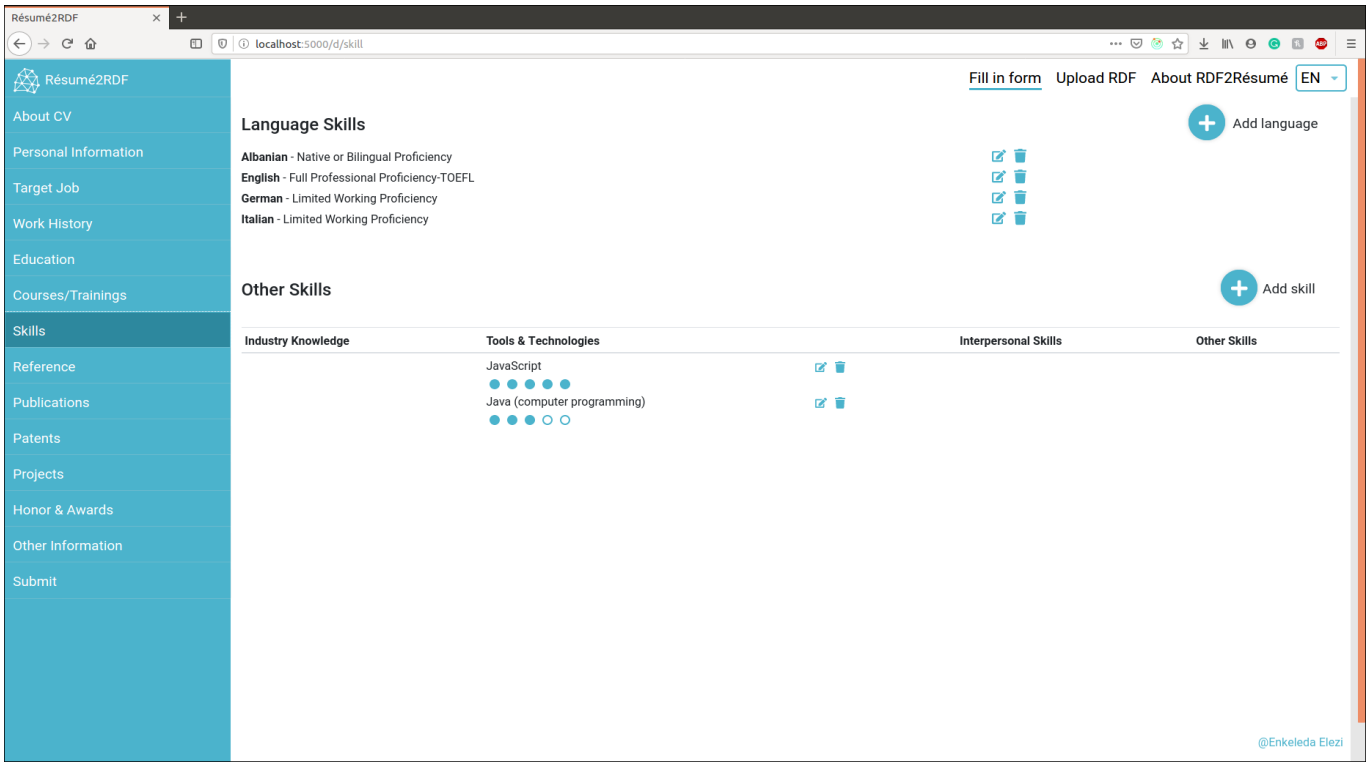

Figure 7.0.10: One of the most important features of the platform is being able to declare the skills acquired throughout the years. The skills are divided into language skills and other ones.

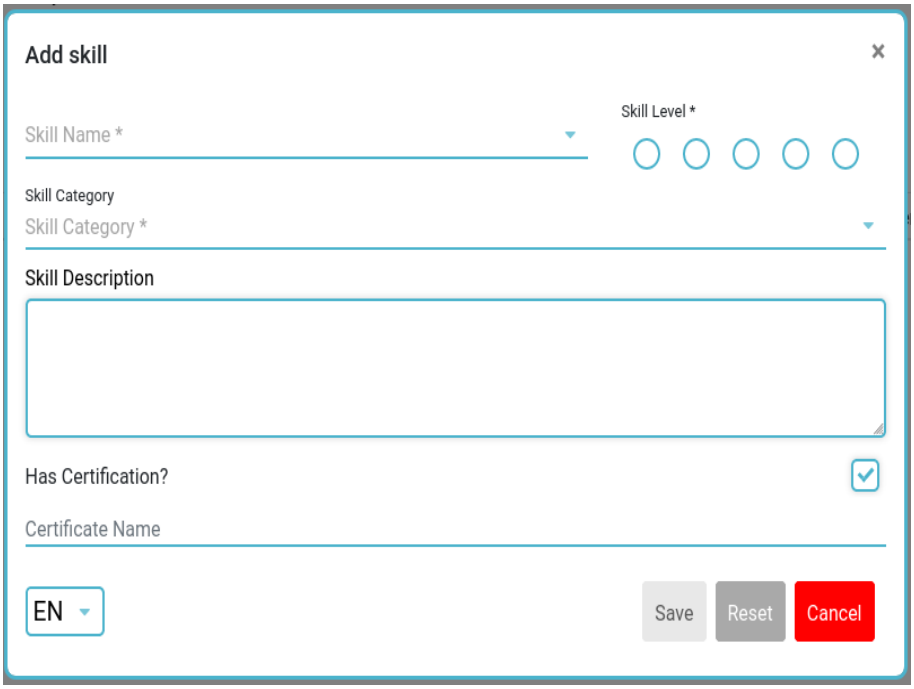

Figure 7.0.11: The user can fill in fields related to a skill's information, based upon Skill Class properties.

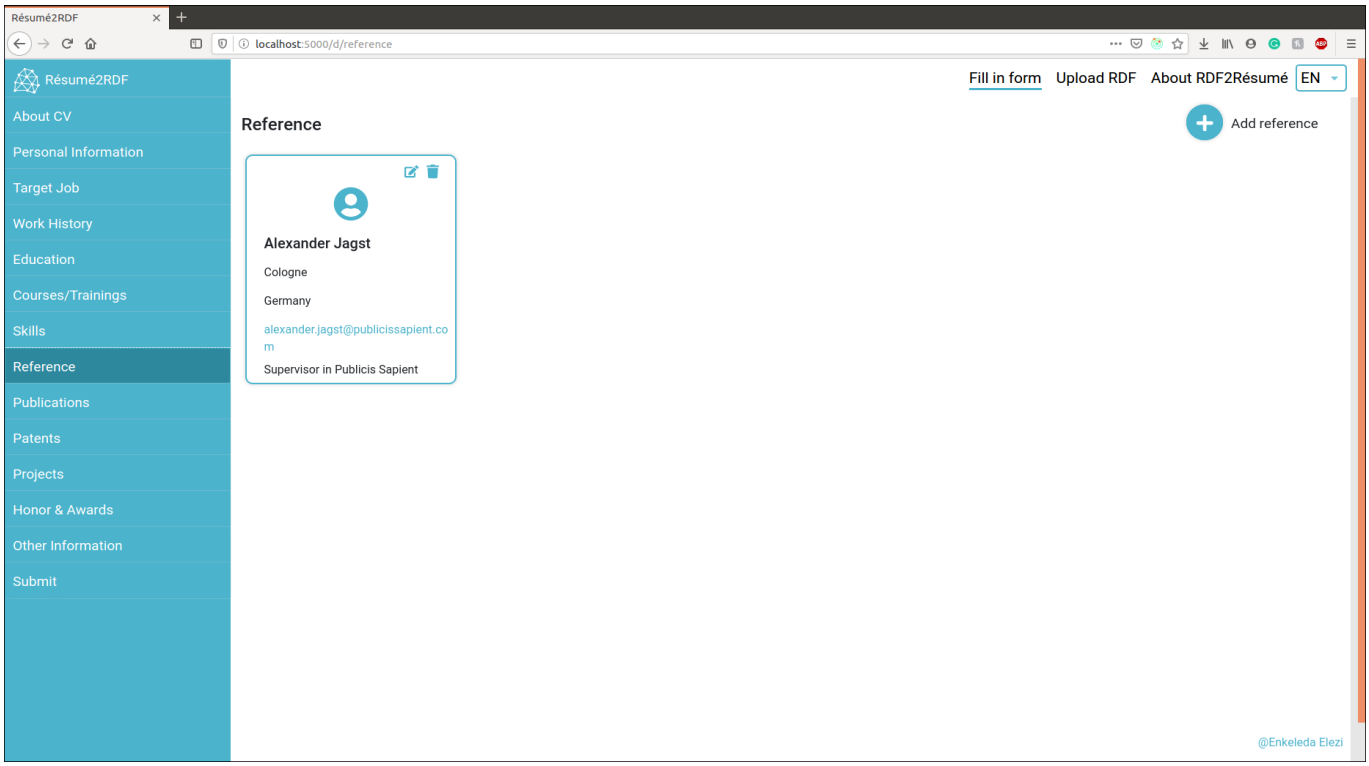

Figure 7.0.12: The reference form gives an overview of the referee people. The form also makes it possible to add, edit or remove a referee.

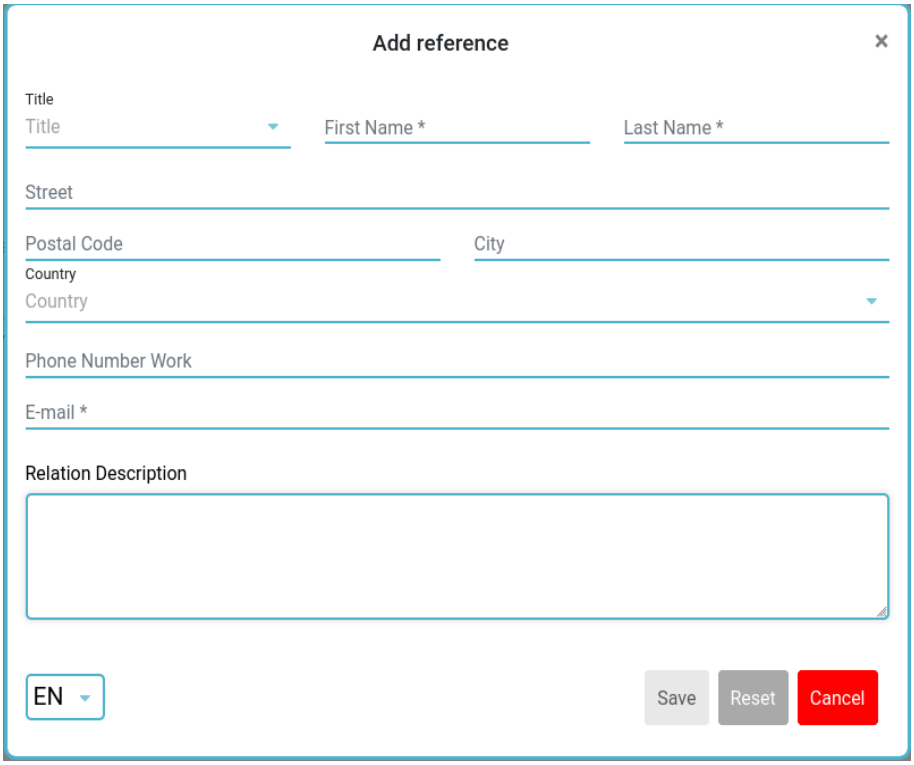

Figure 7.0.13: This modal form contains fields related to the Person Class's properties.

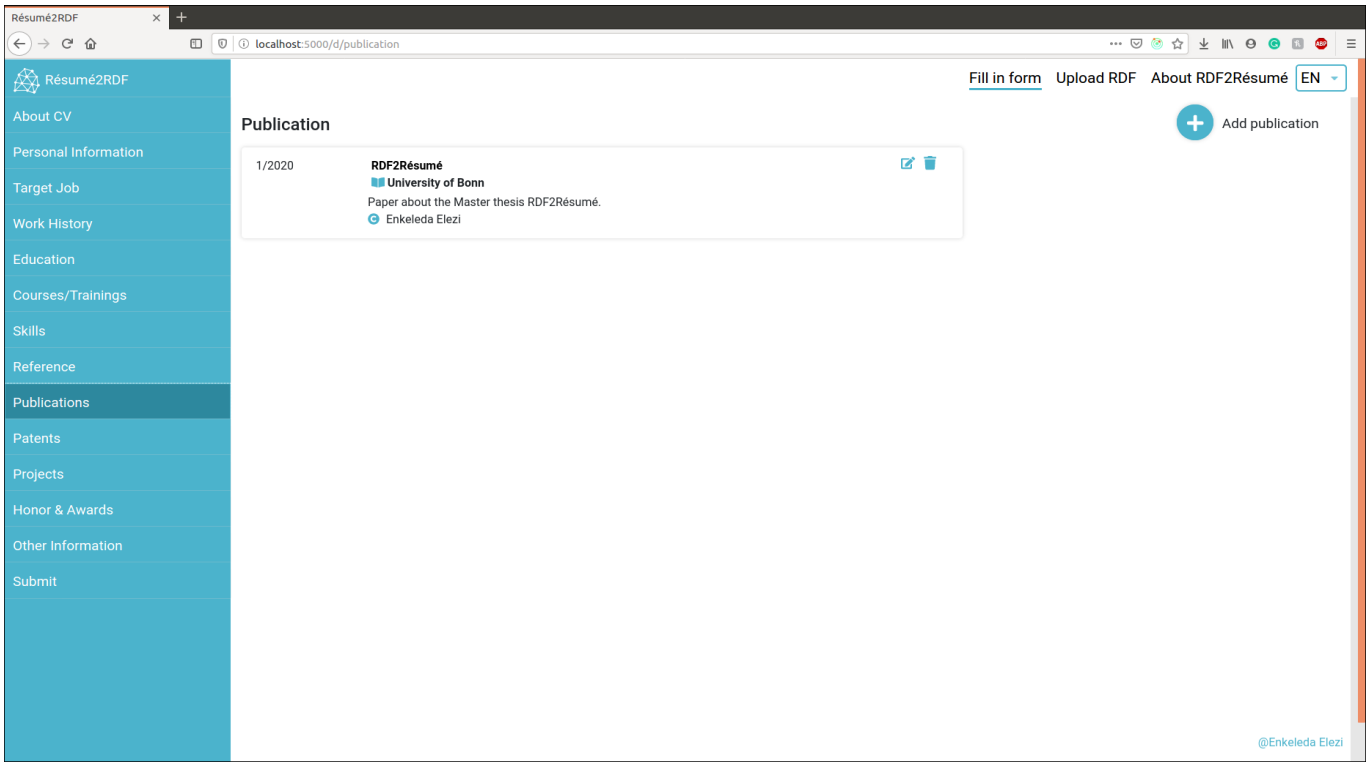

Figure 7.0.14: The Publication form lists all the publications pertaining to the person. The user can also add, edit, or remove a particular publication.

Figure 7.0.15: The fields in this modal form correspond to the properties of the Publication class.

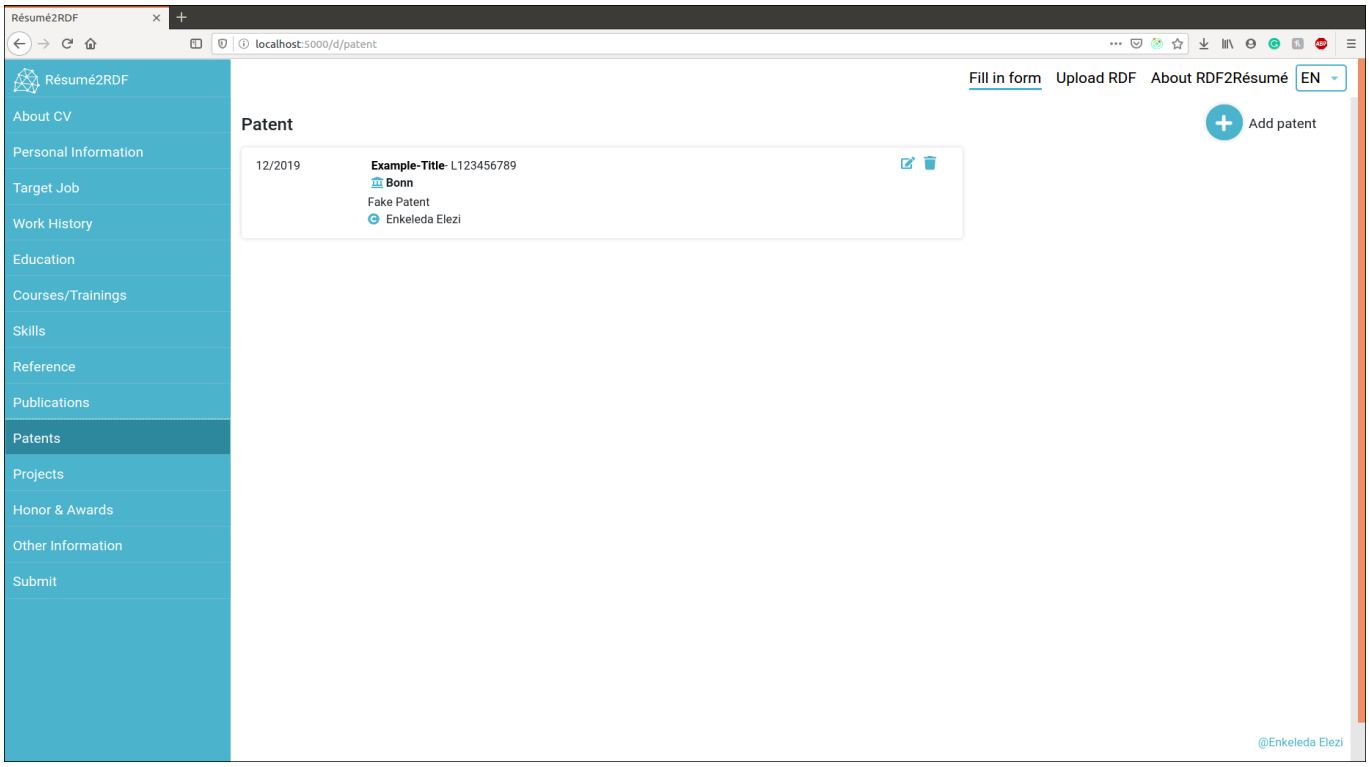

Figure 7.0.16: The Patent form, an extra feature of the platform lets the user add, edit or remove a patent.

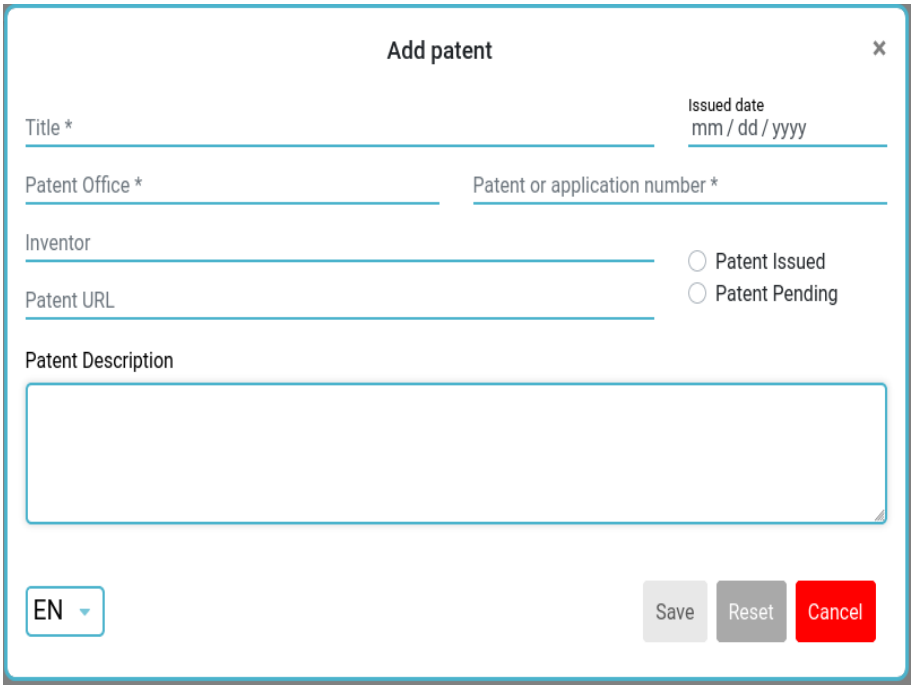

Figure 7.0.17: This modal form allows us to fill in the information regarding a patent. The data correspond to the properties in the Patent Class.

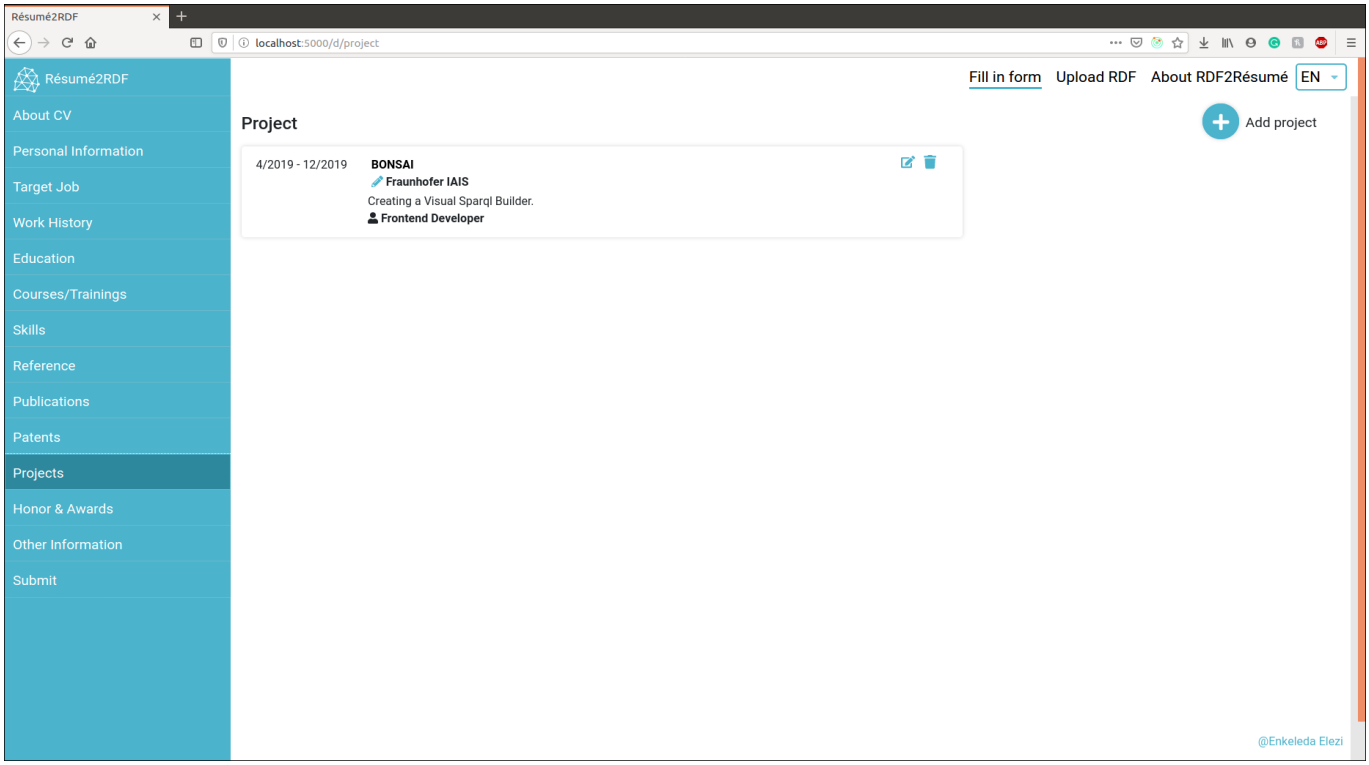

Figure 7.0.18: Defining a project is also really important for employees, especially IT ones. This form allows us to do so by adding, editing, or deleting a particular project.

Figure 7.0.19: The modal form allows us to define information regarding a project and also the role the user had in that project.

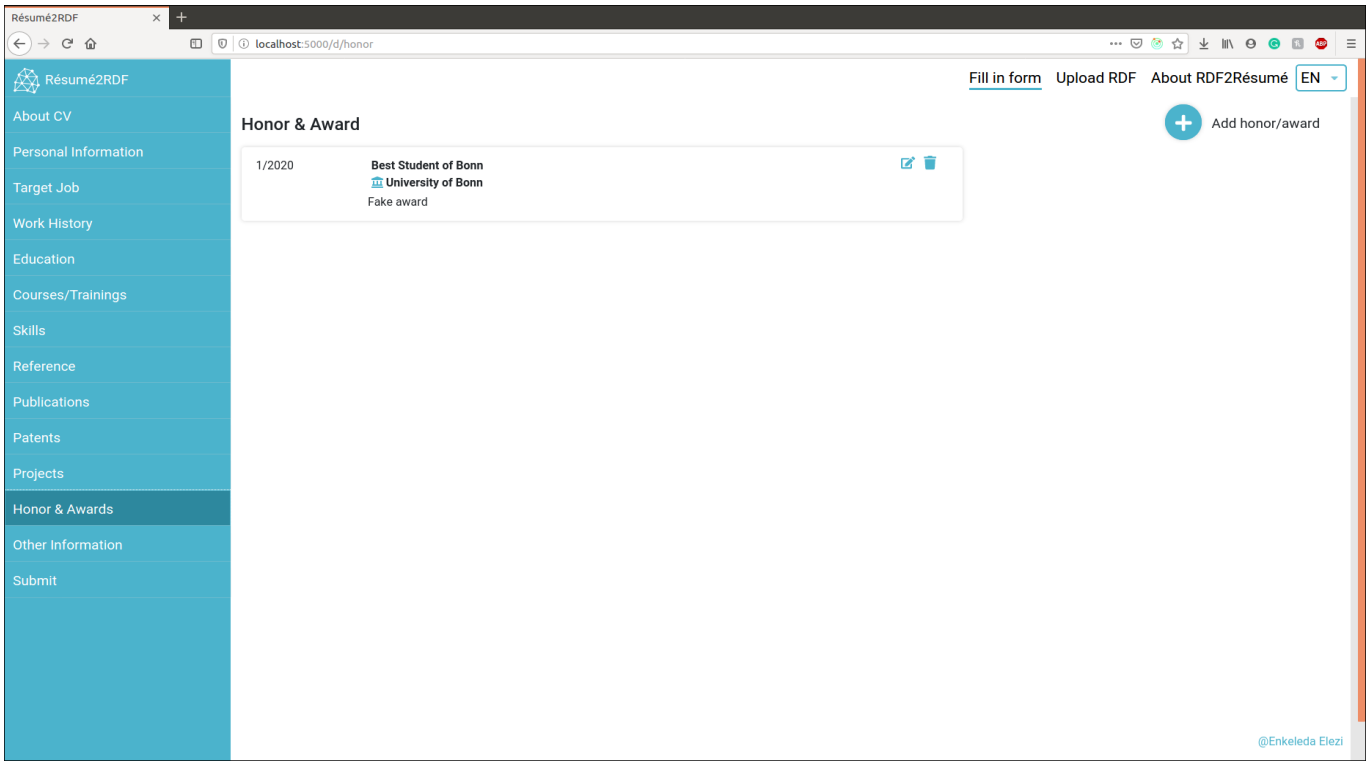

Figure 7.0.20: Another interesting feature of the platform is being able to add information regarding an honor or an award the person acquired. The form includes CRUD functionalities.

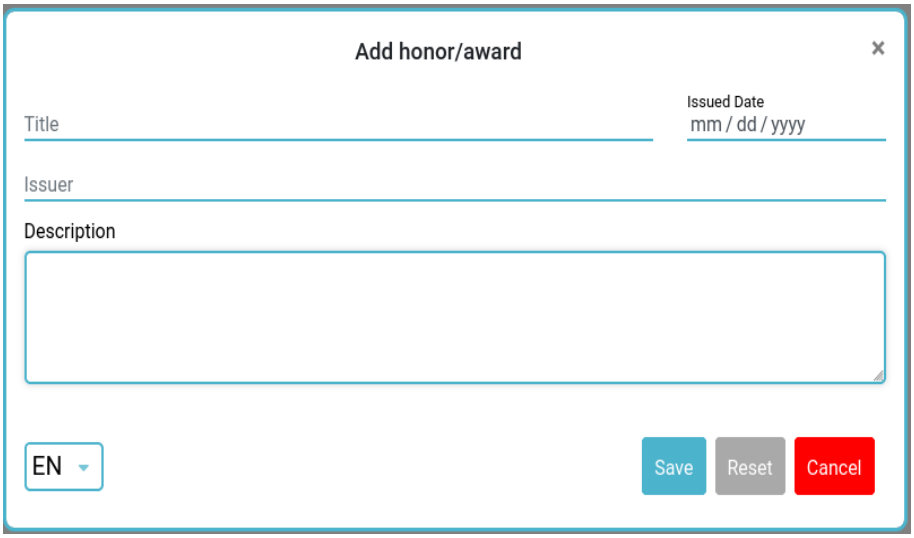

Figure 7.0.21: The modal form defines the relevant fields for a particular honor or award.

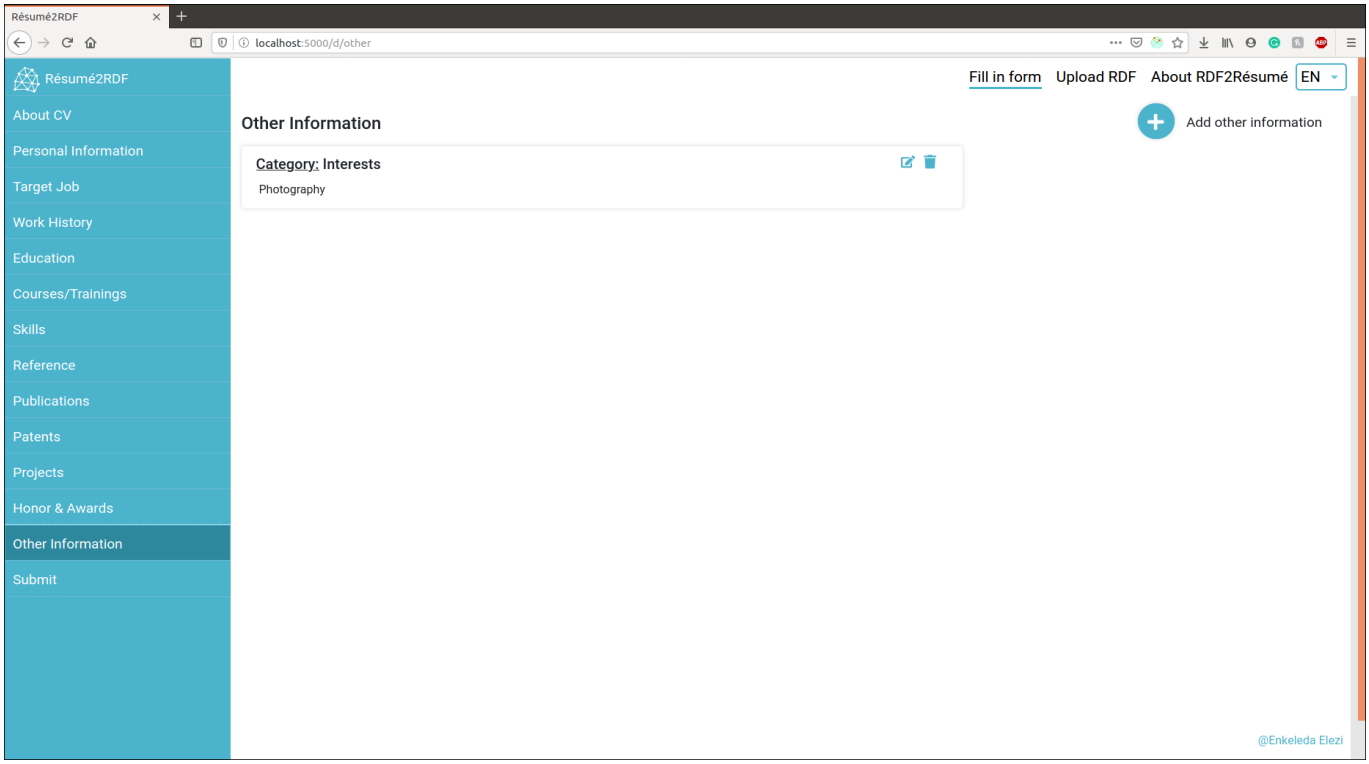

Figure 7.0.22: If the user still wants to add information in the CV, but the previous forms did not define that type, then the user can add it with this generalized other form.

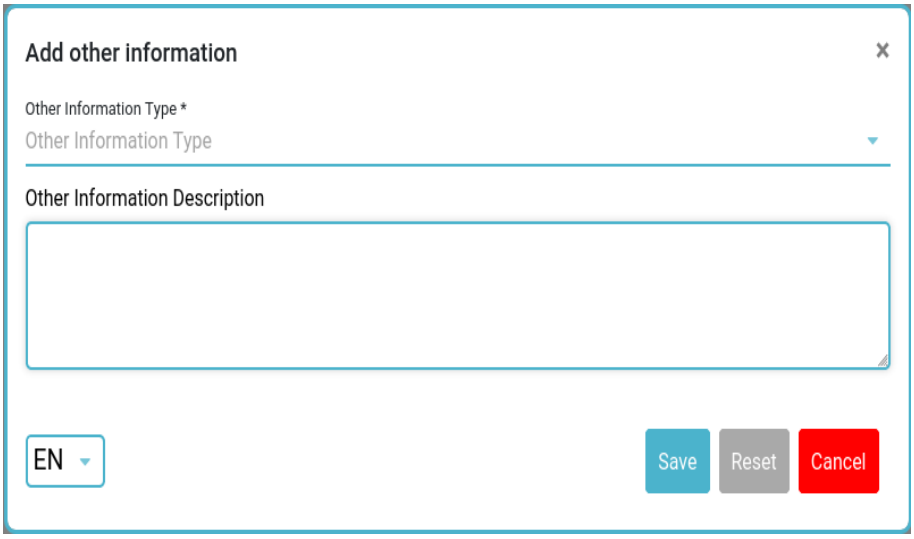

Figure 7.0.23: The modal form allows defining the category of the information being added and a description of the content of that information that corresponds to the Other Class.

Bibliography

[1] Ariyur Mahadevan Abirami and Ch. Padmavathi. Ontology based ranking of documents using graph databases : a big data approach. 2014.

[2] John Allsopp. Microformats: Empowering your markup for web 2.0. 01 2007.

[3] Apache Jena Fuseki. https://jena.apache.org/documentation/fuseki2/. Accessed: 2019-12-12.

[4] Applicant tracking system. https://en.wikipedia.org/wiki/Applicant_tracking_system. Accessed: 2019-11-07.

[5] Axios. https://github.com/axios/axios. Accessed: 2019-12-13.

[6] Costin Badica, Elvira Popescu, Grzegorz Frackowiak, Maria Ganzha, Marcin Paprzycki, Michal Szymczak, and Myon-Woong Park. On human resource adaptability in an agent-based virtual organization. volume 134, pages 111–120, 01 2008.

[7] Wahiba Ben Abdessalem. Using ontology for resume annotation. 11 2012.

[8] Nikos Bikakis, Chrisa Tsinaraki, Nektarios Gioldasis, Ioannis Stavrakantonakis, and Stavros Christodoulakis. The xml and semantic web worlds: Technologies, interoperability and integration. a survey of the state of the art. *Studies in Computational Intelligence*, 418, 01 2013.

[9] Christian Bizer, Ralf Heese, Malgorzata Mochol, Radoslaw Oldakowski, Robert Tolksdorf, and Rainer Eckstein. The impact of semantic web technologies on job recruitment processes. *Wirtschaftsinformatik*, 48:1367–1382, 01 2005.

[10] Uldis Bojars and John Breslin. Resumerdf: Expressing skill information on the semantic web. 01 2007.

[11] Duygu Celik Ertuğrul, Askin Karakas, Gulsen Bal, Cem Gultunca, Atilla Elçi, Basak Buluz, and Murat Alevli. Towards an information extraction system based on ontology to match resumes and jobs. pages 333–338, 07 2013.

[12] Simona Colucci, Tommaso Di Noia, Eugenio Di Sciascio, Francesco Donini, Marina Mongiello, and Marco Mottola. A formal approach to ontology-based semantic match of skills descriptions. *J. UCS*, 9:1437–1454, 01 2003.

[13] George Dafoulas, A.N. Nikolaou, and M. Turega. E-services in the internet job market. pages 9 pp.–, 02 2003.

[14] DBPedia. https://wiki.dbpedia.org/. Accessed: 2019-12-09.

[15] Docker. https://www.docker.com/. Accessed: 2019-12-16.

[16] Europass Portal. https://europass.cedefop.europa.eu/. Accessed: 2019-11-25.

[17] Extending FOAF with Resume Information, uldis bojars. https://www.w3.org/2001/sw/Europe/events/foaf-galway/papers/pp/extending_foaf_with_resume/. Accessed: 2019-11-15.

[18] Maryam Fazel-Zarandi and Mark Fox. Semantic matchmaking for job recruitment: An ontology-based hybrid approach. 525, 01 2009.

[19] FOAF Vocabulary Specification. http://xmlns.com/foaf/spec/. Accessed: 2019-11-15.

[20] Follow Your Nose. https://patterns.dataincubator.org/book/follow-your-nose.html. Accessed: 2019-11-25.

[21] Font Awesome. https://fontawesome.com/. Accessed: 2019-12-13.

[22] hResume. http://microformats.org/wiki/hresume. Accessed: 2019-11-25.

[23] JSON-LD 1.1. https://www.w3.org/TR/json-ld11/. Accessed: 2019-12-01.

[24] Sunil Kumar Kopparapu. Automatic extraction of usable information from unstructured resumes to aid search. 1, 12 2010.

[25] LaTeX – a document preparation system. https://www.latex-project.org/. Accessed: 2019-12-08.

[26] Linkedin. https://www.linkedin.com/. Accessed: 2019-11-18.

[27] Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. Visualizing ontologies with VOWL. *Semantic Web*, 7(4):399–419, 2016.

[28] Ujjal Marjit. Discovering resume information using linked data. *International journal of Web & Semantic Technology*, 3:51–62, 04 2012.

[29] V. Milicic. Semantic web use cases and case studies – case study: Semantic tag. http://www.w3.org/2001/sw/sweo/public/UseCases/Faviki/. Accessed: 2019-11-11.

[30] Roberto Mirizzi, Tommaso Di Noia, Eugenio Di Sciascio, and Michelantonio Trizio. A semantic web enabled system for résumé composition and publication. *IEEE Internet Computing - INTERNET*, pages 583–588, 09 2009.

[31] Malgorzata Mochol, Holger Wache, and Lyndon Nixon. Improving the accuracy of job search with semantic techniques. pages 301–313, 04 2007.

[32] Mohamed Morsey, Jens Lehmann, Sören Auer, Claus Stadler, and Sebastian Hellmann. Dbpedia and the live extraction of structured data from wikipedia. *Program: electronic library and information systems*, 46:157–181, 04 2012.

[33] Eyal Oren, Knud Möller, Simon Scerri, Siegfried Handschuh, and Michael Sintek. What are semantic annotations. 01 2006.

[34] OWL 2 Web Ontology Language Primer (Second Edition). https://www.w3.org/TR/owl2-primer/. Accessed: 2019-12-02.

[35] Python. https://www.python.org/. Accessed: 2019-12-06.

[36] RDF 1.1 N-Quads. https://www.w3.org/TR/n-quads/. Accessed: 2019-12-01.

[37] RDF 1.1 N-Triples. https://www.w3.org/TR/n-triples/. Accessed: 2019-12-01.

[38] RDF 1.1 TriG. https://www.w3.org/TR/trig/. Accessed: 2019-12-01.

[39] RDF 1.1 Turtle. https://www.w3.org/TR/turtle/. Accessed: 2019-12-01.

[40] RDF 1.1 XML Syntax. https://www.w3.org/TR/rdf-syntax-grammar/. Accessed: 2019-12-02.

[41] RDF 1.1 XML Syntax. https://www.w3.org/TR/rdf-syntax-grammar/. Accessed: 2019-12-02.

[42] RDF Schema 1.1. https://www.w3.org/TR/rdf-schema/. Accessed: 2019-12-02.

[43] RDF2Résumé. https://github.com/e-elezi/rdf2resume.

[44] RDFa 1.1 Primer - Third Edition. https://www.w3.org/TR/rdfa-primer/. Accessed: 2019-12-02.

[45] React a javascript library for building user interfaces. https://reactjs.org/. Accessed: 2019-12-13.

[46] Redux a predictable state container for js apps. https://redux.js.org/. Accessed: 2019-12-13.

[47] ResumeRDF Ontology Specification. http://rdfs.org/resume-rdf/. Accessed: 2019-11-15.

[48] Bhavika Vishanji Mrug Ruchita Haresh Makwana, Aishwarya Mohan Medhekar. Resume refinement system using semantic analysis. *International Journal of Innovative Science and Research Technology*, 3, 03 2018.

[49] Résumé. https://en.wikipedia.org/wiki/R%C3%A9sum%C3%A9. Accessed: 2019-11-01.

[50] Semantic Web. https://www.w3.org/standards/semanticweb/. Accessed: 2019-11-02.

[51] Single-page application. https://en.wikipedia.org/wiki/Single-page_application. Accessed: 2019-12-20.

[52] J. Smedt, M. Vrang, and Agis Papantoniou. Esco: Towards a semantic web for the european labor market. volume 1409, 01 2015.

[53] Inc Sovren Group. Sovren. https://sovren.com/. Accessed: 2019-11-21.

[54] SPARQL 1.1 Query Language. https://www.w3.org/TR/sparql11-query/. Accessed: 2019-12-02.

[55] SweetAlert2. https://sweetalert2.github.io/. Accessed: 2019-12-13.

[56] European Union. Europass. https://europass.cedefop.europa.eu/. Accessed: 2019-11-01.

[57] WebVOWL: Web-based visualization of ontologies. http://vowl.visualdataweb.org/webvowl.html. Accessed: 2019-11-25.

[58] Xing. https://www.xing.com/. Accessed: 2019-11-18.