



UNIVERSIDAD DE
MURCIA



Tell-OP Final Application Report

Mattia Zago

mattia.zago@um.es

Gregorio Martínez Pérez

gregorio@um.es

ABSTRACT

This document presents the implemented functionalities for the Output 10 of the European project “TELL-OP - Transforming European Learner Language into Learning Opportunities”. Along with each functional requirement it presents the chosen implementation technology either server-side (if any) and client-side. Each functional requirement also features sample screenshots from a real case scenario.



Supported by
the Erasmus+ programme
of the European Union

TELL-OP - Transforming European Learner Language into Learning Opportunities A
KA200 Higher Education Strategic Partnership 2014-1-ES01-KA203-004782 licensed
under a Creative Commons Attribution-Noncommercial-Share Alike 4.0
International License.



Facultad de Informática
Departamento de Ingeniería de la Información y las Comunicaciones

Campus Universitario de Espinardo. 30100 Murcia
T. 868 88 4664 - F. 868 88 4151 - www.um.es/web/diic



TABLE OF CONTENTS

Section I. License	3
1.01 Application	3
1.02 Server	3
1.03 Operational Ownership	4
Section II. Functionalities	5
2.01 Languages	5
2.02 Dictionaries	6
(a) Collins Dictionary	9
(b) Stands4 Dictionary	10
(c) Other dictionaries	11
2.03 Part of Speech Tagger	12
2.04 Language Level Distribution	15
2.05 Exercise Analysis	15
Section III. Implementation	16
3.01 Enumerators	16
3.02 Technology	18
Section IV. Pages	19
4.01 Login	19
4.02 Dashboard	20
4.03 Main Search	21
4.04 Settings	22
4.05 Exercise View	23
4.06 Exercise Analysis	24
Section V. Indexes	25
Section VI. References	26



Section I. License

1.01 Application

The application is released under the Apache 2.0 License, 2004 version. Here follows the official statement, included in the header of each non-auto-generated source code file:

```
Licensed under the Apache License, Version 2.0 (the "License"); you may not
use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

The full license is available online on the official GitHub repository, you may obtain a copy at <https://github.com/TellOP/APP/blob/master/LICENSE>.

Several libraries referenced by the Tell-OP source code and bundled with the Tell-OP app require reproduction of the license text in the distribution; those are publicly available online, you may obtain a copy at <https://github.com/TellOP/APP/blob/master/NOTICE>.

The copyright holder is the University of Murcia. Here follows the official statement, included in the header of each non-auto-generated source-code file:

```
Copyright © 2016 University of Murcia
```

The list of collaborators is available online on the official GitHub repository, you may obtain a copy at <https://github.com/TellOP>.

1.02 Server

The server is released under the Apache 2.0 License, 2004 version. Here follows the official statement, included in the header of each non-auto-generated source code file:

```
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

The full license is available online on the official GitHub repository, you may obtain a copy at <https://github.com/TellOP/SERVER/blob/master/LICENSE>.



The official GitHub repository includes third-party components licensed under their own terms and conditions. The notice file is publicly available online, you may obtain a copy at <https://github.com/TelloP/SERVER/blob/master/NOTICE>.

The copyright holder is the University of Murcia. Here follows the official statement:

Copyright © 2016 University of Murcia

The list of collaborators is available online on the official GitHub repository, you may obtain a copy at <https://github.com/TelloP>.

1.03 Operational Ownership

As at April, 1st 2017, the application is released with different operational ownerships in order to facilitate the fast deployment cycle:

- Android version: University of Murcia¹ - Open Beta.
- UWP version: Mattia Zago² - Open Beta.
- iOS version: Mattia Zago³ - Closed Beta.

The intent is to transfer all the ownerships to the University of Murcia once each version exits the beta testing phase.

¹ <https://play.google.com/store/apps/details?id=es.um.tellop>

² <https://www.microsoft.com/en-us/store/p/tellop-beta/9nblggh4413j>

³ <https://github.com/TelloP/APP/wiki/Beta-testing#ios-beta-testing>



Section II. Functionalities

This section will introduce the functionalities available in the Tell-OP application from the release 0.5beta (version 6). The schema used to explain the communication protocols and the use-cases are aligned with the UML® formalism⁴, a standard visual modelling language, used in this document to analyse, describe and for providing a user-friendly schema of the implemented functionalities.

2.01 Languages

Thanks to the localization native functionalities of each supported operative system, the application supports the multi/language layout configuration. Currently the application is delivered with the full translation in English (en-GB, en-US), Spanish (es-ES), German, (de-DE), French (fr-FR), Portuguese (pt-PT), Turkish (tr-TR) and Italian (it-IT); according to the guidelines, the default locale for the interface is en-US.

The translation service is managed with the online service Transifex [1]. The access is public and anyone can submit a translation review or a new locale translation for both the application⁵ and the server⁶. On the user device, the locale is automatically detected and the corresponding translation, if any, is applied to the interface.

Regarding the main functionalities, both the application and the server have MULTI-5 capabilities, here follows a detailed list of each language comprehensive of relative support and implementation status:

1. British English (en-GB): fully supported, all the functionalities are implemented and tested.
2. German (de-DE): partially supported, some functionalities are not available due to the lack of online services.
3. Spanish (es-ES): partially supported, some functionalities are not available due to the lack of online services.
4. French (fr-FR): compatible but not supported in this prototype.
5. Italian (it-IT): compatible but not supported in this prototype.

The user can change the language for the functionalities in the Settings page, available either from the Login page and from the secondary menu in some pages (Dashboard, Main Search).

⁴ Unified Modelling Language™ (UML®), <http://www.omg.org/spec/UML/index.htm>

⁵ In order to join the App translation team, please refer to the official Wiki page, available in the GitHub repository, at <https://github.com/TellOP/APP/wiki/Localization>.

⁶ In order to join the App translation team, please refer to the official Wiki page, available in the GitHub repository, at <https://github.com/TellOP/SERVER/wiki/Localization>.

2.02 Dictionaries

The first main functionality of the Tell-OP application is the ability to search any word in any supported language independently from the actual page showed. In order to do so, the server provides supports for five independent online dictionaries, i.e. the Collins API [2], the Oxford Dictionaries [3], the Stands4 Web Services [4], the StringNet Navigator [5] and the Netspeak [6].

The use case for this functionality consists of two sub-use cases, one for the application and the other for the webserver. Instead of representing the server as an actor, the schema shows the inner controllers as a white-box component, Figure 1 highlights the relationships between the controllers, but not the interaction with the online provides (external actor). For such a schema, see the protocol in Figure 2.

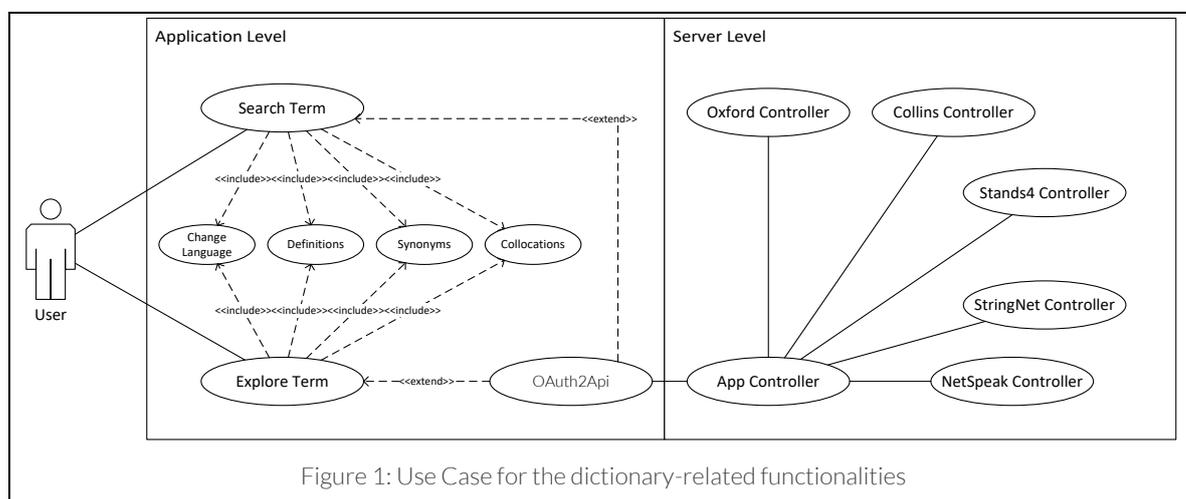
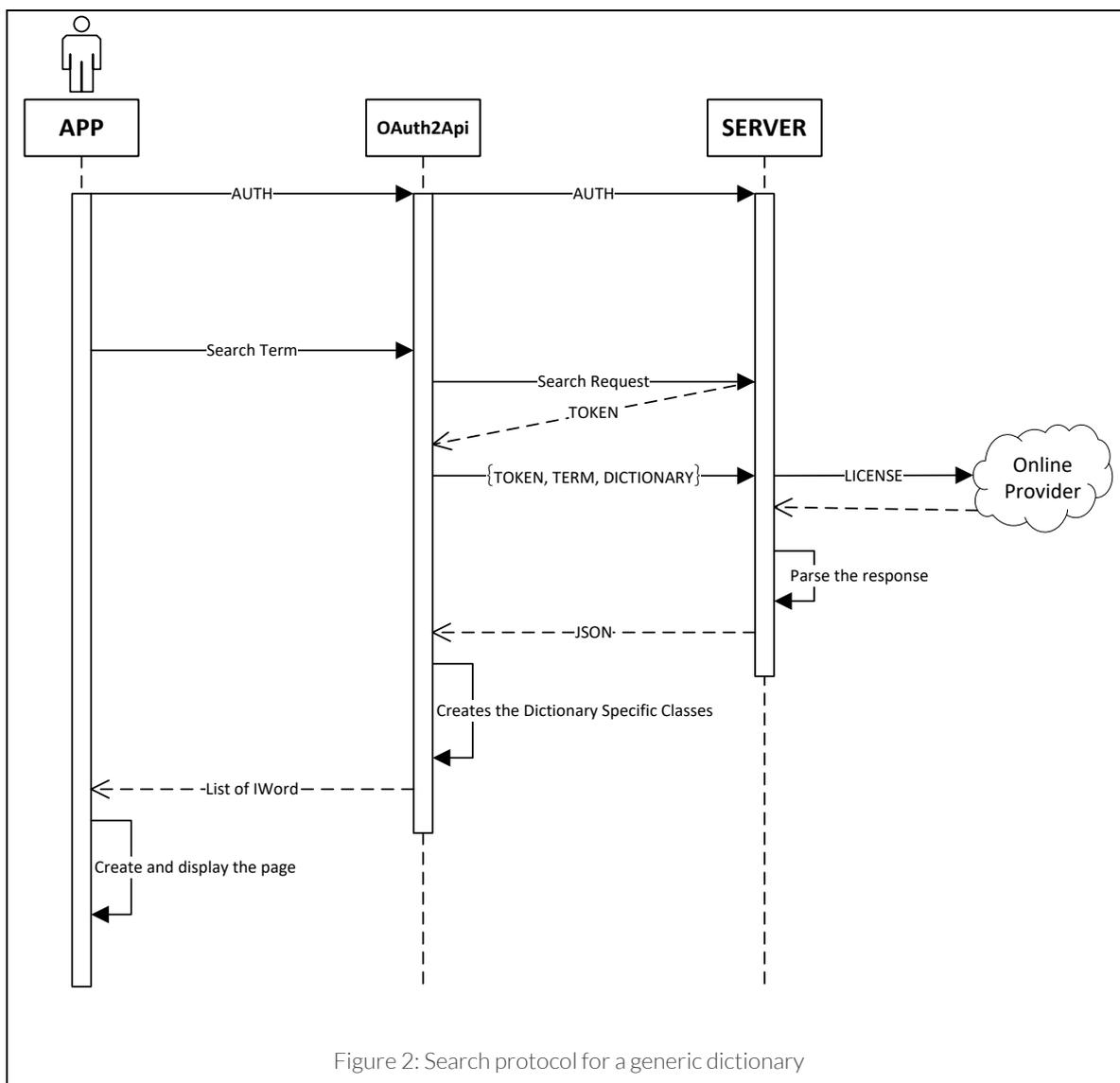


Figure 1: Use Case for the dictionary-related functionalities

As shown in the Figure 1 there are two access points for the dictionaries, the first one is through the button “Search (Q)”, the second one is by tapping on an enabled word (“Exploration”).

Whenever a user decides to search for, or explore, a term, the application send the requested word, along with the language LCID, to the web server according to the protocol showed in Figure 2.



Once the response is obtained from the online API, see the following sub-sections for the technical details, the server parses the data into a machine-friendly JSON response and sends it back to the application.

The app maps the fields in the JSON to the corresponding dictionary-related classes. All the atomic results are implemented with the same interface “IWord”. A non-exhaustive schema is presented through a UML class schema in Figure 3.



(a) Collins Dictionary

The Collins Dictionary package is used as main dictionary in the app. It supports both English and German while providing definitions, synonyms, antonyms, examples and related terms. The Collins API license was requested by Mattia Zago on 29/03/2016 and will expire on 28/03/2017. This license is limited with 5000 requests per month, Collins provides commercial licenses for upscaling this limit. As at April, 30th 2017 the Collins license expired, the proposed upgrade to a commercial license was rejected, thus this functionality has been disabled.

(i) Package Schema

In Figure 4 is presented the Collins Dictionary class schema for the application.

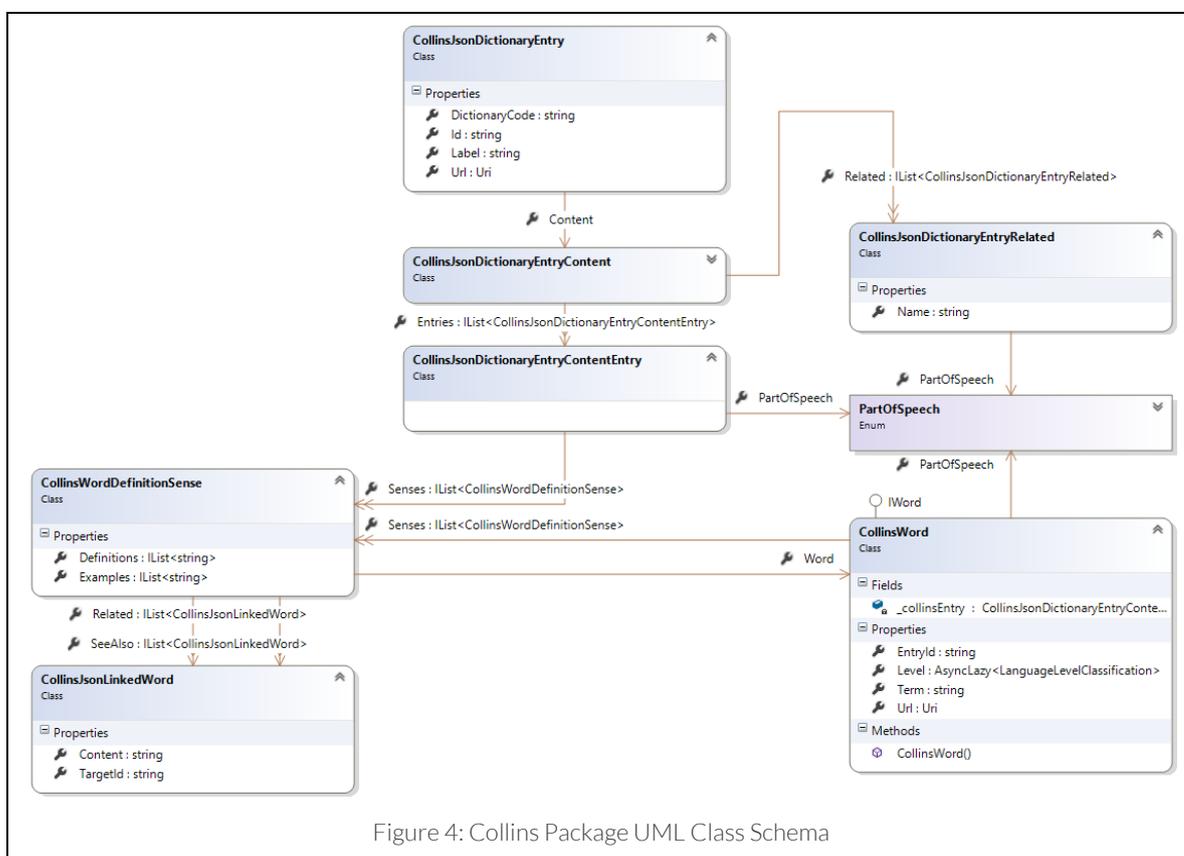


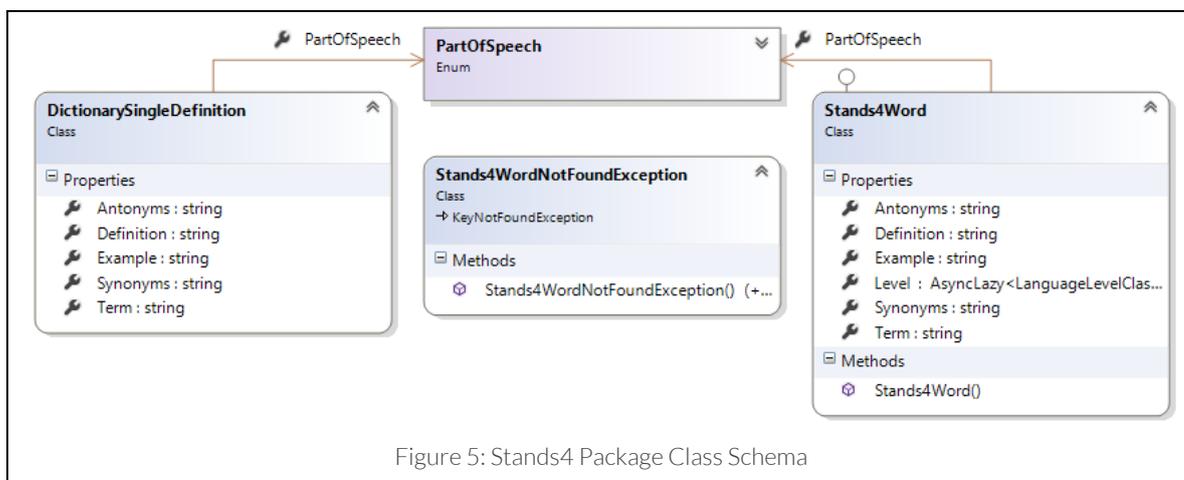
Figure 4: Collins Package UML Class Schema

(b) Stands4 Dictionary

The Stands4 dictionary is the second most used dictionary in the app, it provides definitions, synonyms and antonyms. The license was requested by Mattia Zago on 27/03/2016 and it should not expire. The limit is set to 100 requests per day, and Stands4 provides premium commercial licenses in order to increase this limit.

(i) Package Schema

In Figure 5 is presented the class schema for the Stands4 package in the application.



(c) Other dictionaries

This subsection presents the minor dictionaries used in the application. Due to their low complexity, their class schemas are presented together in Figure 6.

(i) Oxford Dictionary

The Oxford dictionary was chosen to be the main Spanish reference for definitions and examples. The license key was updated on 19/01/2017 and should be available until 18/01/2018. This license is limited to 3000 requests per month, there are also available commercial solutions for increasing the upper limit.

(ii) StringNet Dictionary

The StringNet web service provides a list of collocations for a given English word. The website doesn't provide an API accessible, so the webserver retrieves the HTML page and the parse it in order to send the data back to the application.

(iii) Netspeak Dictionary

The Netspeak web service provides a free-to-use open API for word collocations. It partially supports both Spanish and German.

(iv) Classes Schema

The classes schema for Oxford Dictionary, StringNet Dictionary, and Netspeak Dictionary are presented in Figure 6.

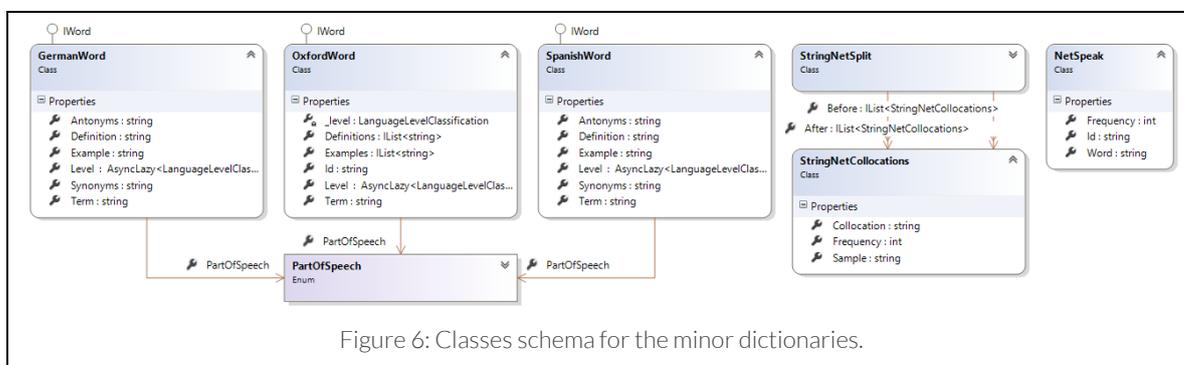
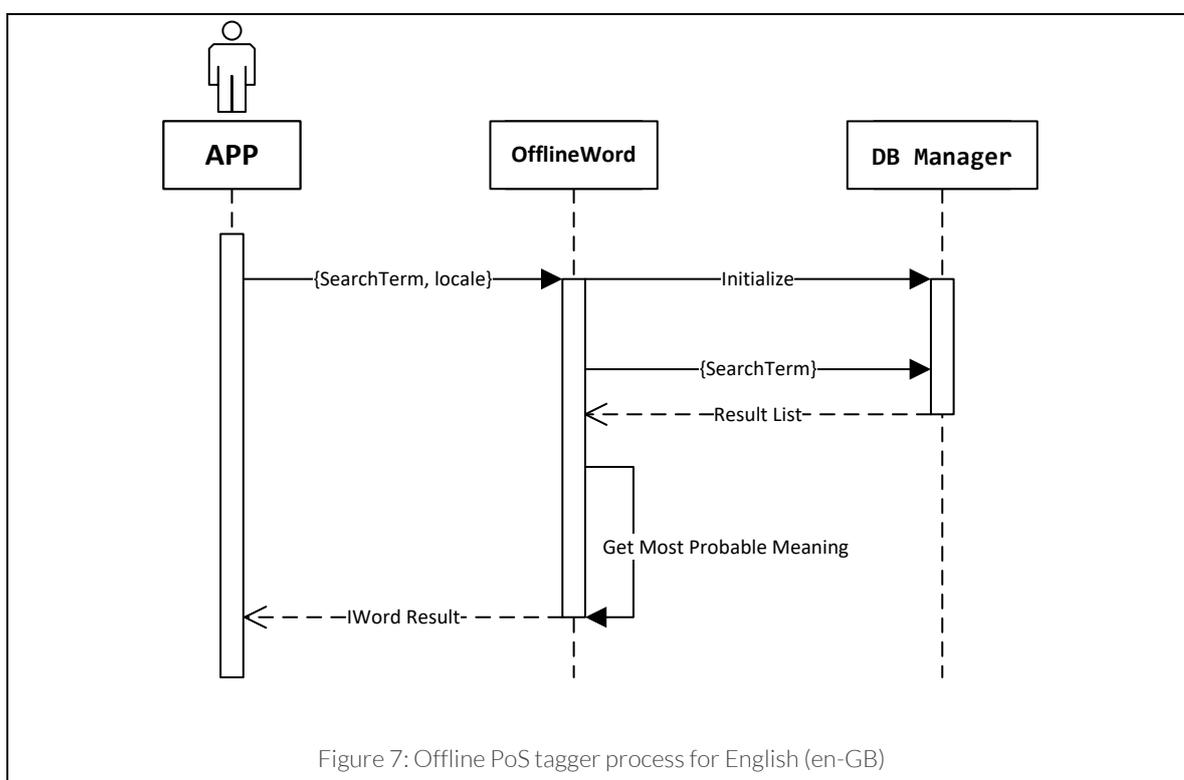


Figure 6: Classes schema for the minor dictionaries.

2.03 Part of Speech Tagger

Whenever a user writes an exercise, the app analyses in real time the content in order to provide useful metrics such as the percentage of nouns, verbs, adjectives and, in general, for each part of speech, the list of words categorized.

In order to obtain such data, the app provides two mechanisms, the first one, the offline mode, is enabled and usable for the English idiom, while the second one, the online mode, is enabled and usable for the German and Spanish idioms.



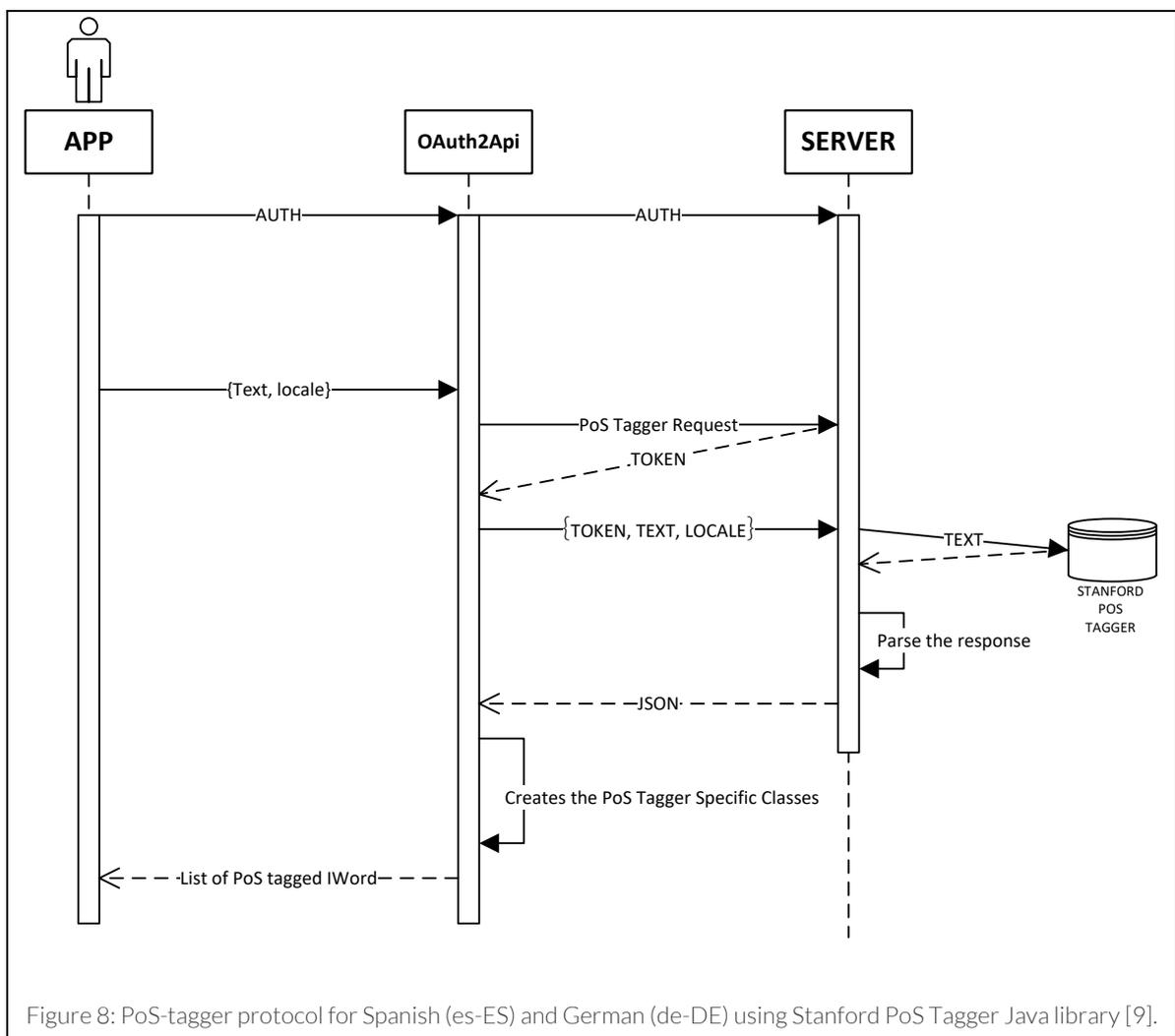
As presented in Figure 7 the offline PoS-tagging process is an interactive search in the local SQLite [7] database. The data stored in this database are a cached copy of English Profile [8] database.

In order to correctly interact with the user text, and due to the innate difficulty of the semantic analysis, the process is divided in sub-process, each one taking cares of a smaller problem. First of all, the text is divided in token according to the following RegEx⁷. Each token is then delivered to a method that removes all the contractions (such as don't, he's, etc.) with the expanse form (do not, he is, etc.). Thus, the output represents a text semantically identical

⁷ Regular Expression, a special string text for describing a search pattern. See www.regular-expressions.info for more information.

to the original text, but with all the words separated and independently analysable. Each word is then converted to the corresponding headword, and only then the base term is searched in the local database for a match. The search function outputs one or more part of speech for a given base term, thus an algorithm chooses one of the results and outputs a single PoS.

Clearly in order to complete this operation successfully the lemma algorithm and the selection algorithm must be accurately tuned. For this prototype, unfortunately, those components are trivial, and the reports always the most frequent (a.k.a. common) solution.



Since online resources like English Profile [8] aren't available for idioms such as Spanish or German, the PoS-tagger is demanded to the webserver, which analyses the text with the Stanford PoS Tagger [9]. Since this library is a Java software, the developer team realizes a



custom PHP wrapper for it based on the work from Charles Hays [10]. The results are then mapped to our part of speech enumerator, according to the Table 1.

Locale	Part of Speech (Stanford)	Tell-OP PoS
Spanish (es-ES)	A00000, AQ0000	ADJECTIVE
Spanish (es-ES)	CC, CS	CONJUNCTION
Spanish (es-ES)	DA0000, DD0000, DE0000, DI0000, DN0000, DP0000, DT0000	DETERMINER
Spanish (es-ES)	FO, FAA, FAT, FC, FD, FE, FG, FH, FIA, FIT, FP, FPA, FPT, FS, FT, FX, FZ, I	INTERJECTION OR DISCOURSE MARKER
Spanish (es-ES)	NC00000, NCON000, NCOP000, NCOS000	COMMON NOUN
Spanish (es-ES)	NP00000	PROPER NOUN
Spanish (es-ES)	RG, RN	ADVERB
Spanish (es-ES)	SP000	PREPOSITION
Spanish (es-ES)	VMG0000, VMIC000, VMIF000, VMII000, VMIP000, VMIS000, VMM0000, VMN0000, VMP0000, VMSI000, VMSP000	VERB
Spanish (es-ES)	VAG0000, VAIC000, VAIF000, VAII000, VAIP000, VAIS000, VAM0000, VAN0000, VAP0000, VASI000, VASP000	AUXILIARY VERB
Spanish (es-ES)	VSG0000, VSIC000, VSIF000, VSII000, VSIP000, VSIS000, VSM0000, VSN0000, VSP0000, VSSF000, VSSI000, VSSP000	MODAL VERB
Spanish (es-ES)	<i>DEFAULT FALLBACK</i>	UNCLASSIFIED
German (de-DE)	ADJA, ADJD	ADJECTIVE
German (de-DE)	KOUI, KOUS, KON, KOKOM	CONJUNCTION
German (de-DE)	ITJ	INTERJECTION OR DISCOURSE MARKER
German (de-DE)	NN	COMMON NOUN
German (de-DE)	NE	PROPER NOUN
German (de-DE)	FM	FOREIGN WORD
German (de-DE)	PDS, PDAT, PIS, PIAT, PIDAT, PPER, PPOSS, PPOSAT, PRELS, PRELAT, PRF, PWS, PWAT, PRF	PRONOUN
German (de-DE)	ADV, PWAV, PAV	ADVERB
German (de-DE)	APPR, APPRART, APPO, APZR, APPR, APPR, APPR	PREPOSITION
German (de-DE)	PTKNEG, PTKVZ, PTKANT, PTKA	EXISTENTIAL PARTICLE
German (de-DE)	VVFIN, VVIMP, VVINFIN, VVIZU, VVPP, VAFIN, VAIMP, VAINFIN, VAPP, VMFIN, VMINFIN, VMPP	VERB
German (de-DE)	\$, \$. \$(*IGNORE*
German (de-DE)	<i>DEFAULT FALLBACK</i>	UNCLASSIFIED

Table 1: Part of Speech codes association from Stanford tags to Tell-OP enumerator.

2.04 Language Level Distribution

Similarly, to the Part of Speech Tagger, this module maps all the words written to the corresponding CEFR levels, the protocol is illustrated in Figure 9. Unfortunately, since the resources for this type of tagging are available only for British English, the German and Spanish idioms are not supported.

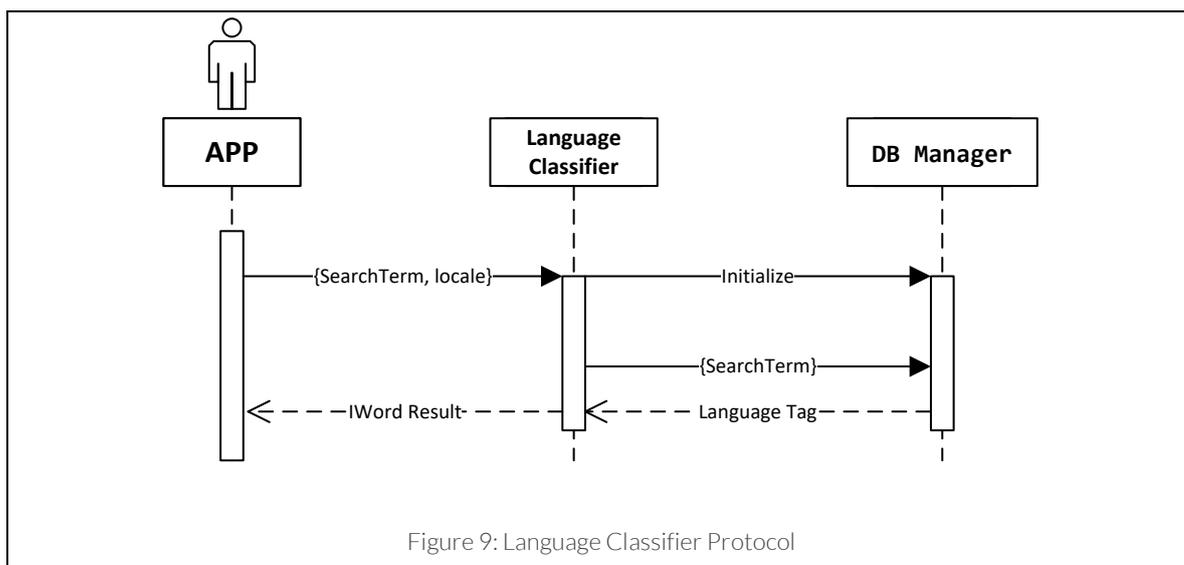


Figure 9: Language Classifier Protocol

2.05 Exercise Analysis

Once the user submits the text, the server logs the entry and then sends the text to the online analyser. This phase is enabled only for the British English idiom, due to the lack of compatible resources for either German and Spanish languages.

The English text is submitted to the Lex Tutor [11] analyser, the result, when compatible, is converted to a JSON format and then send back to the application.

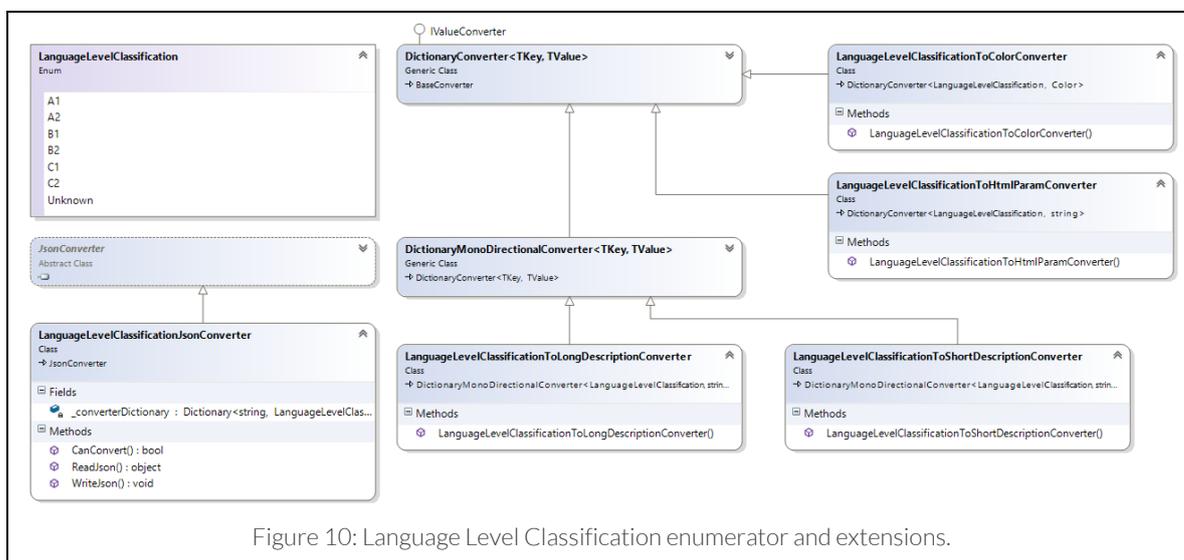
Section III. Implementation

3.01 Enumerators

In order to improve the control over the data, three enumerators were defined:

- the Part of Speech enumerator, which provides an exhaustive list of part of speech tags;
- the Language Level enumerator, which provides all the CEFR levels;
- the Supported Languages enumerator, which provides the list of the available languages.

Each enumerator is equipped with one or more extension classes that provide conversion methods. The class schema is presented in Figure 10, Figure 11 and Figure 12 respectively.



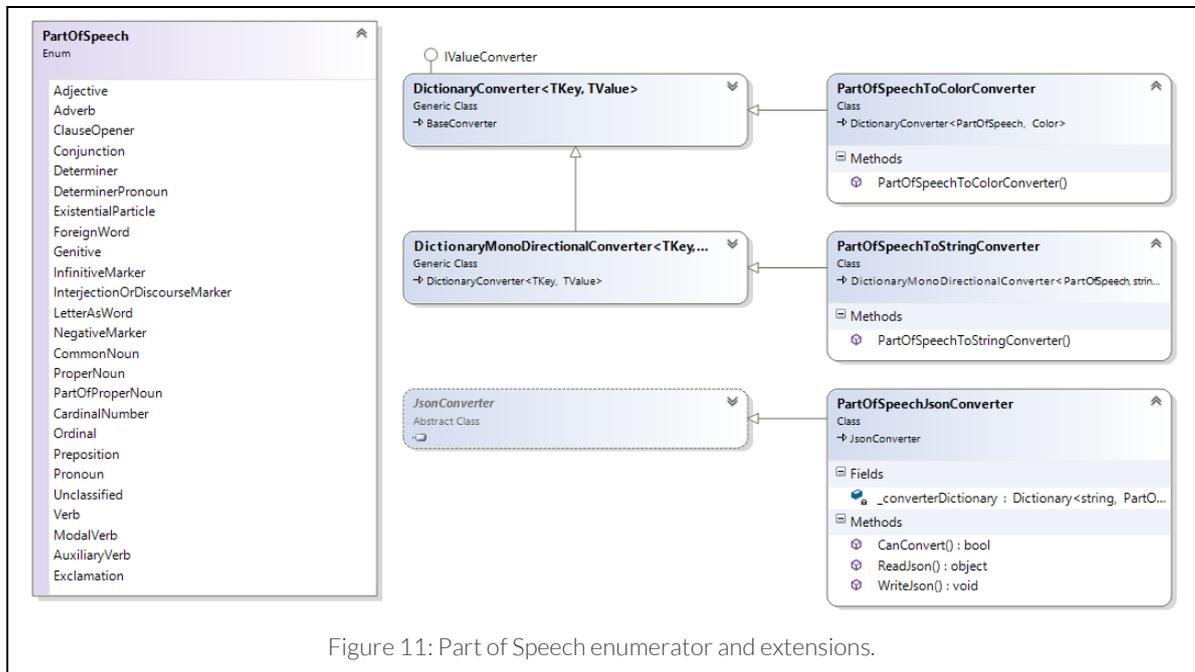


Figure 11: Part of Speech enumerator and extensions.

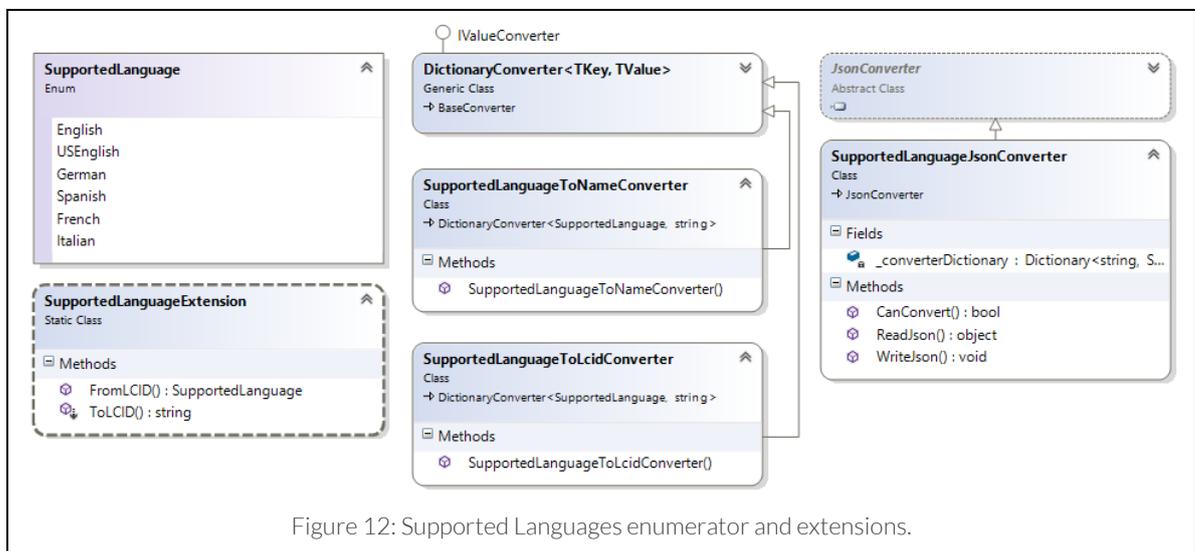


Figure 12: Supported Languages enumerator and extensions.



3.02 Technology

The application was developed using the Xamarin Forms⁸ framework which should allow building native UIs for iOS, Android and Windows from a single, shared C# codebase.

Nevertheless, the real situation is different: while the codebase is mostly shared (more than 97% LOC⁹ are shared among the platforms), the interface code is coarse. Their approach through XAML¹⁰ is innovative and promising, but the differences in each platform are rooted enough to make extremely difficult the development of aligned look-and-feel layouts.

This approach, although valid on the paper, precludes the usage of most of the basic constructs that simplify the development of a given platform. The framework itself forces to design custom controllers instead of using well-known and established techniques typical of iOS, Android and UWP. Once a custom controller is defined, the framework encodes the layout using native primitives, with the consequence of different design outcomes.

Those differences are deep-seated in the framework, forcing the introduction of complicated switches in order to predict every possible variation of the data. This phase of unexpected analysis and code reform caused unexpected delays in the development and the impossibility of delivering on time all the versions.

The layout codebase was then aligned with the Android version as it is the operative system with the highest market share. UWP and iOS were forced into separate branches and developed separately.

Regarding the server codebase, the customization of the Flight¹¹ micro-framework has been proved a valid choice, although forced the dev team to customize most of the functionalities.

The server access is through a RESTful¹² interface, implemented with a combination of PHP, XML, JS, HTML5 and CSS3. All the libraries are shipped with the Composer technology; you may find a complete list by decoding the “composer.json” file¹³. The installation guide is available at <https://github.com/TelloP/SERVER/wiki/Installation>.

The server codebase contains 19,114 Style LOC, 22,519 HTML LOC, 52,014 dynamic JS LOC and 12105 XML LOC.

The compiled project results in over than 540 thousand lines of code.

⁸ <https://www.xamarin.com/forms>

⁹ Lines of code

¹⁰ eXtensible Application Markup Language, see <https://developer.xamarin.com/guides/xamarin-forms/xaml/>

¹¹ <http://flightphp.com/>

¹² Representational state transfer, see https://en.wikipedia.org/wiki/Representational_state_transfer

¹³ <https://github.com/TelloP/SERVER/blob/master/composer.json>

Section IV. Pages

4.01 Login

The first page presented to the user is the login page. In this page there are six different navigation options, although only the “Log in” button allows the user to enter in the application. From this page, and following the “Sign up” button the user is sent through the default web browser to the server page, from where he can register for using the app.

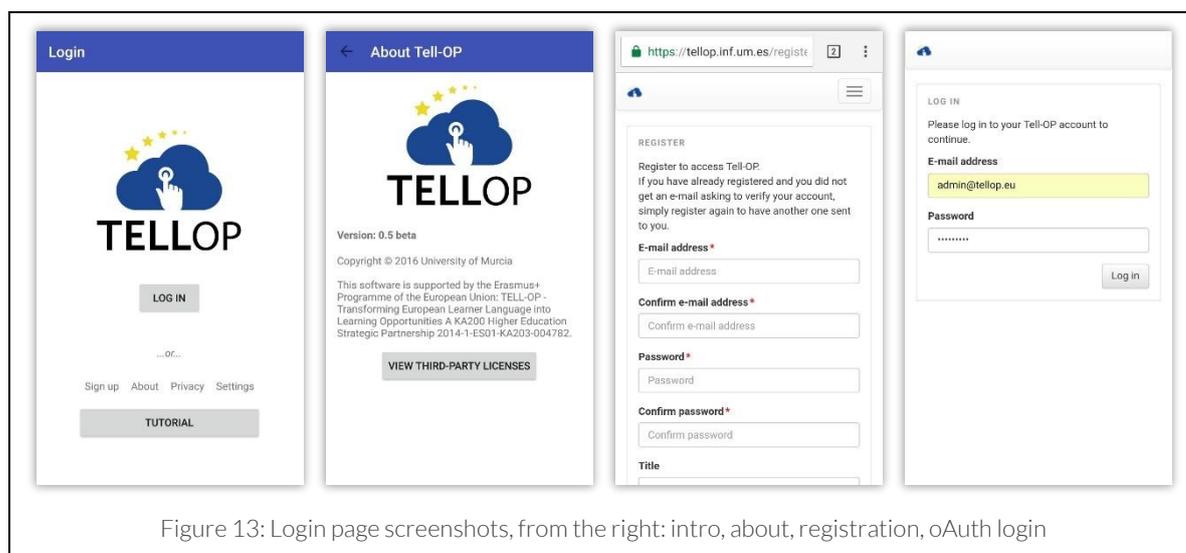


Figure 13: Login page screenshots, from the right: intro, about, registration, oAuth login

4.02 Dashboard

The dashboard page is divided into two tabs:

- Features: which presents, for each level and for each language enabled, all the possible activities that the user can fulfil;
- History: which stores the list of already completed exercises, ordered by date.

The page also offers the options of opening the Main Search page or the Settings page.

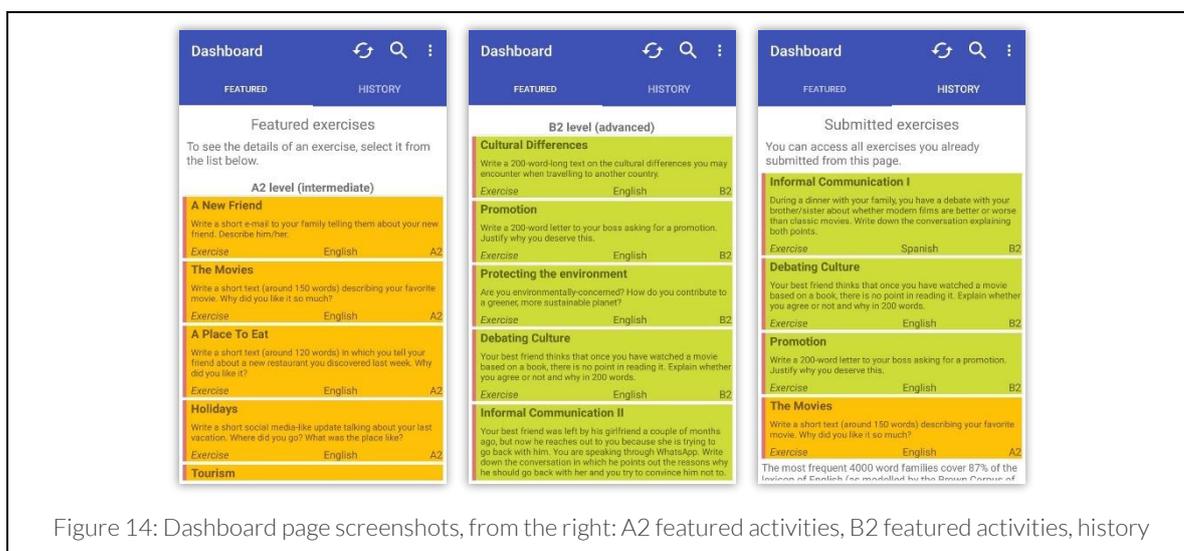


Figure 14: Dashboard page screenshots, from the right: A2 featured activities, B2 featured activities, history



4.03 Main Search

This page presents any search result retrieved from the online resources (see Dictionaries section for more information). It is divided in two panels, the first one presents the result of the dictionary search, where for each sense are presented: the definition, the examples (if any), the related words (if the dictionary supports them and if any), the synonyms and the antonyms. The second panel presents a sample usage for the searched word, specifically, it retrieves from the Netspeak Dictionary the collocation lists ordered by frequency. Those lists are truncated to 100 examples in order to speed up the elaboration. The search language can be changed through the Settings page button, available behind the burger menu, as specified in the Languages section, only British English, German and Spanish have supported dictionaries. The definitions sub-panel may change accordingly to the available dictionaries for each idiom.

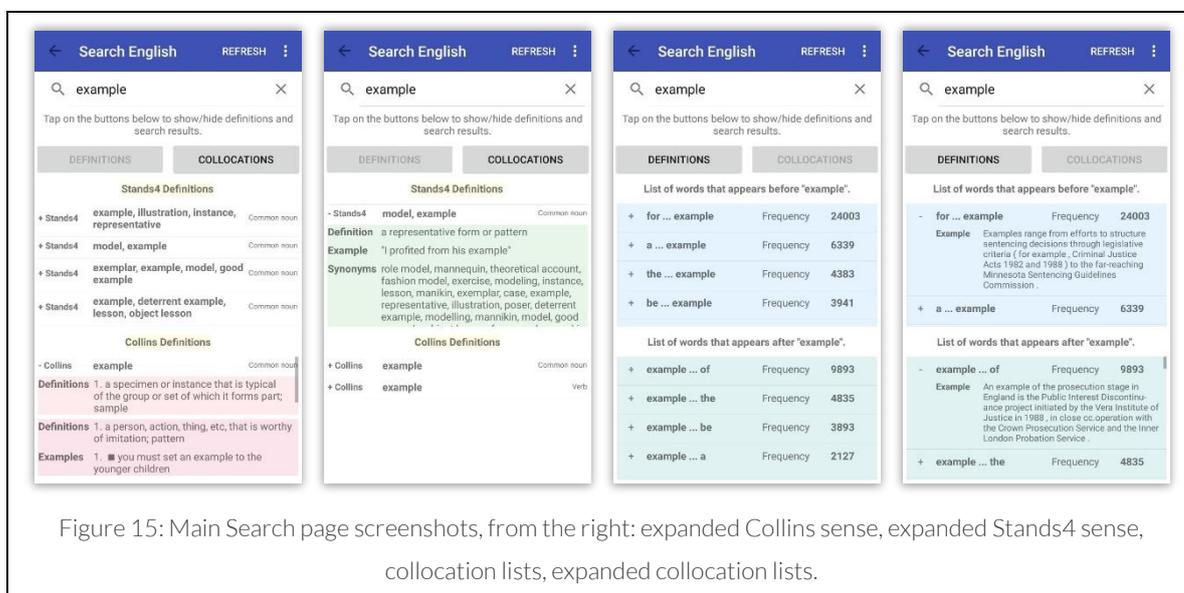


Figure 15: Main Search page screenshots, from the right: expanded Collins sense, expanded Stands4 sense, collocation lists, expanded collocation lists.



4.04 Settings

The settings page stores the variable switch for the supported languages, it enables the user to activate (or deactivate) any supported language. The result of this choice will be active after refreshing the Dashboard page. From this page is also possible to choose the active search languages between the three currently supported.

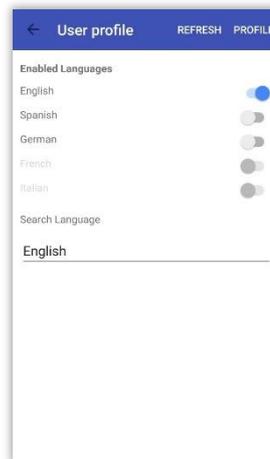
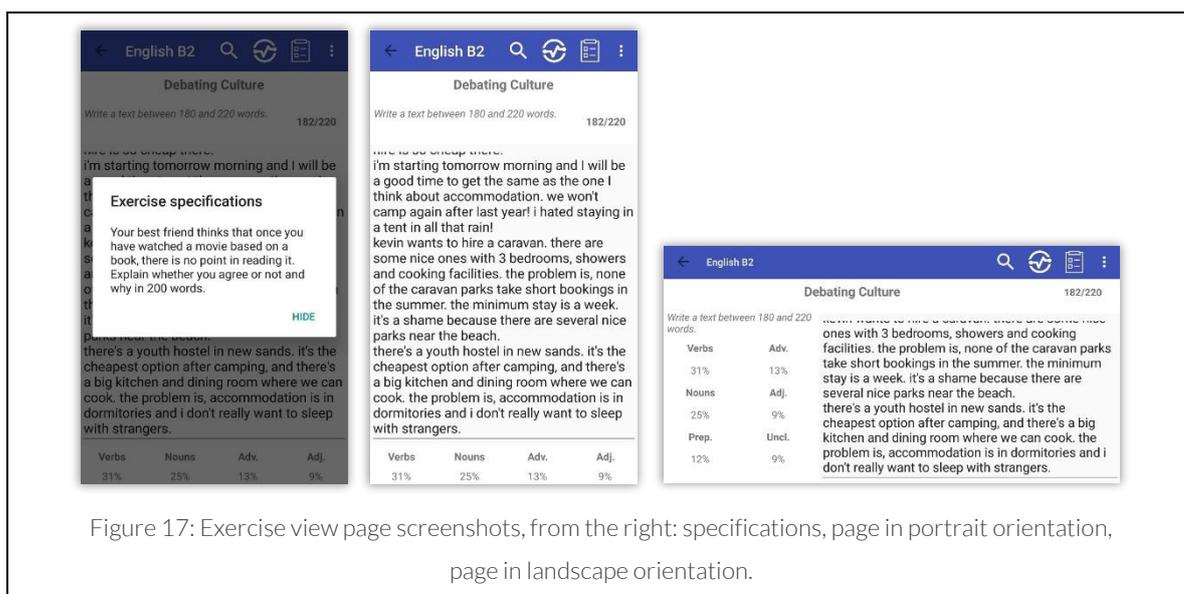


Figure 16: Settings page screenshot

4.05 Exercise View

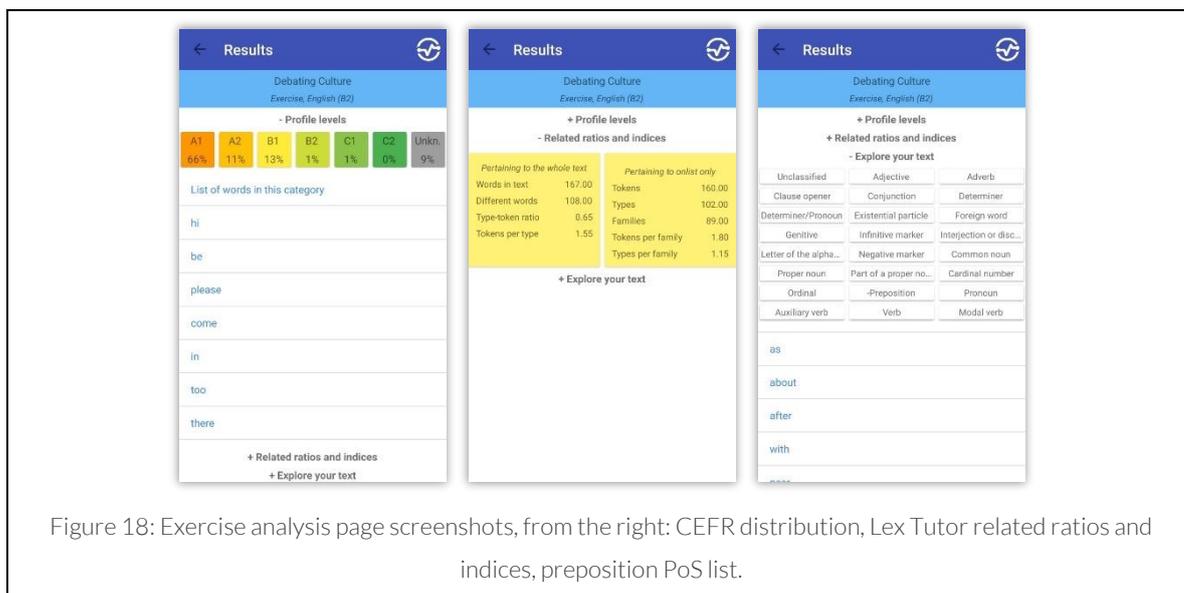
This page permits to complete the selected activity. There are six possible options usable from the upper menu bar:

- Search (🔍): this primary button opens the Main Search page.
- Fast Analysis (🔄): this primary button refreshes the analysis made by the Part of Speech Tagger.
- Show Text (📄): this primary button reopens the consign text for the exercise.
- Submit (behind the burger menu “☰”): this secondary button submits the text to the server, and shows the resulting Exercise Analysis page.
- Reload (behind the burger menu “☰”): this secondary button discards the current uncommitted text and restores the previously saved data.
- Reset (behind the burger menu “☰”): this secondary button discards any user text and resets the exercise.



4.06 Exercise Analysis

This page presents the results of the elaborated text retrieved from the online services as stated in the Exercise Analysis section. It is implemented through three hidden panels, the first one shows the profiling results (see Language Level Distribution section), the second one displays the advanced metrics, and the third one presents the text divided grouped by Part of Speech (see Part of Speech Tagger for more information). Each word is tab-enabled and will open the corresponding search page.





Section V. Indexes

Figure 1: Use Case for the dictionary-related functionalities.....	6
Figure 2: Search protocol for a generic dictionary	7
Figure 3: Non-exhaustive UML class schema for the interface IWord and its implementations.	8
Figure 4: Collins Package UML Class Schema.....	9
Figure 5: Stands4 Package Class Schema.....	10
Figure 6: Classes schema for the minor dictionaries.....	11
Figure 7: Offline PoS tagger process for English (en-GB)	12
Figure 8: PoS-tagger protocol for Spanish (es-ES) and German (de-DE) using Stanford PoS Tagger Java library [9].....	13
Figure 9: Language Classifier Protocol	15
Figure 10: Language Level Classification enumerator and extensions.....	16
Figure 11: Part of Speech enumerator and extensions.....	17
Figure 12: Supported Languages enumerator and extensions.	17
Figure 13: Login page screenshots, from the right: intro, about, registration, oAuth login	19
Figure 14: Dashboard page screenshots, from the right: A2 featured activities, B2 featured activities, history.....	20
Figure 15: Main Search page screenshots, from the right: expanded Collins sense, expanded Stands4 sense, collocation lists, expanded collocation lists.....	21
Figure 16: Settings page screenshot	22
Figure 17: Exercise view page screenshots, from the right: specifications, page in portrait orientation, page in landscape orientation.....	23
Figure 18: Exercise analysis page screenshots, from the right: CEFR distribution, Lex Tutor related ratios and indices, preposition PoS list.	24



Section VI. References

- [1] Transifex US Inc., "Transifex," Transifex US Inc., [Online]. Available: <https://www.transifex.com/>. [Accessed 01 02 2017].
- [2] Collins, "Dictionary API," 01 02 2017. [Online]. Available: <https://www.collinsdictionary.com/api>.
- [3] Oxford, "Oxford Dictionary API," 01 02 2017. [Online]. Available: <https://developer.oxforddictionaries.com/>.
- [4] Stands4 LLC., "Dictionary Definitions API," 01 02 2017. [Online]. Available: http://www.definitions.net/definitions_api.php.
- [5] StringNet, "StringNet Navigator 4.0," 01 02 2017. [Online]. Available: <http://nav4.stringnet.org/>.
- [6] Netspeak, "Developer," 01 02 2017. [Online]. Available: <http://www.netspeak.org/#developer>.
- [7] "SQLite," [Online]. Available: <https://www.sqlite.org/>. [Accessed 01 02 2017].
- [8] English Profile Research Project, "English Profile," [Online]. [Accessed 01 02 2017].
- [9] The Stanford Natural Language Processing Group, "Stanford Log-linear Part-Of-Speech Tagger," [Online]. Available: <http://nlp.stanford.edu/software/tagger.shtml>. [Accessed 01 01 2017].
- [10] C. Hays, "PHP Class Wrapper for Stanford Part of Speech Tagger," [Online]. Available: <http://charleshays.com/php-class-wrapper-for-stanford-part-of-speech-tagger/>. [Accessed 01 02 2017].
- [11] Heatley & Nation, VUW New Zealand & B. Laufer, "Web VP Classic v.4," [Online]. Available: <http://www.lex tutor.ca/vp/eng/>. [Accessed 01 02 2017].