



Integrated intelligent LEARNing environment for Reading and Writing

D6.3 – The Final iLearnRW System



Document identifier	iLearnRW_D6.3_The_Final_ILearnRW_System
Date	2015-09-29
WP	WP6
Partners	DOLPHIN, NTUA, UoM, IoE, LBUS
WP Lead Partner	DOLPHIN
Document status	Final

Deliverable Number	D6.3
Deliverable Title	The Final iLearnRW System
Deliverable version number	V0.6/Final
Work package	WP6
Task	Task 6.3 System Refinement
Nature of the deliverable	Report (R)
Dissemination level	Public (PU)
Date of Version	2015-09-29

Author(s)	C. Litsas
Contributor(s)	H. Martinez, D. Lukeš
Reviewer(s)	C. Miħu, A. Symvonis
Abstract	In this report we describe the final version of the integrated ILearnRW system prototype as it was refined by incorporating feedback from the evaluation phase of the project. A fully operational final prototype is delivered.
Keywords	iLearnRW

Document Status Sheet

Issue	Date	Comment	Author
V0.1	2015-09-03	Initial text, and structure of the document	Chris Litsas
V0.2	2015-09-09	Added text about the iLearnRW serious game (H. Martinez) and other applications	Chris Litsas
V0.3	2015-09-16	Added information about the dictionaries, taken from D. Lukeš	Chris Litsas
V0.4	2015-09-23	Review	Cantemir Mihiu
V0.5	2015-09-25	Review	Antonios Symvonis
V0.6/Final	2015-09-29	Final Version, incorporates comments from C. Mihiu and A. Symvonis	Chris Litsas

Project information

Project acronym:	ILearnRW
Project full title:	Integrated Intelligent Learning Environment for Reading and Writing
Proposal/Contract no.:	318803

Project Officer: Krister Olson

Address:	L-2920 Luxembourg, Luxembourg
Phone:	+35 2430 134 332
E-mail:	krister.olson@ec.europa.eu

Project Co-ordinator: Noel Duffy

Address:	Dolphin Computer Access Ltd. Technology House, Blackpole Estate West, Worcester, UK. WR3 8TJ
Phone:	+01 905 754 577
Fax:	+01 905 754 559
E-mail:	noel.duffy@yourdolphin.com

Table of Contents

1. INTRODUCTION	6
2. ILEARNRW SYSTEM OVERVIEW	8
2.1. TARGET GROUPS	8
2.2. MODE OF OPERATION	9
3. DESCRIPTION OF THE ILEARNRW SYSTEM	11
3.1. YEAR 3 – TECHNICAL DEVELOPMENT OVERVIEW	11
3.2. SERIOUS GAME	12
3.2.1. <i>Game World</i>	12
3.2.2. <i>Mail Sorter</i>	14
3.2.3. <i>Bridge Reinforcer</i>	14
3.2.4. <i>Harvest</i>	15
3.2.5. <i>Serenade Hero</i>	16
3.2.6. <i>Moving Pathways</i>	16
3.2.7. <i>Solomon’s Junkyard</i>	17
3.2.8. <i>Whack a Monkey</i>	18
3.2.9. <i>Mail Delivery</i>	18
3.2.10. <i>Typing Train Dispatcher</i>	19
3.3. READER	20
3.4. ON LINE RESOURCE BANK	24
3.4.1. <i>Profile Viewer</i>	24
3.4.2. <i>Text Analysis</i>	25
3.4.3. <i>Text Annotation</i>	27
3.5. TEXT CLASSIFICATION TOOL	29
3.5.1. <i>Home Screen</i>	29
3.5.2. <i>Profile Viewer</i>	30
3.5.3. <i>Text Analysis</i>	30
3.6. DATA ANALYSIS CONSOLE APPLICATION	33
3.6.1. <i>Parser for the Serious Game data</i>	33
3.6.2. <i>Parser for the Screening Test data</i>	34
3.7. WEB APPLICATION – ADMIN PANEL	36
4. ILEARNRW DICTIONARIES	38
5. TECHNICAL DETAILS	40
5.1. SYSTEM COMPONENTS	40
5.1.1. <i>Server Side</i>	40
5.1.2. <i>Client Applications</i>	41
5.1.3. <i>Scalability of the iLearnRW eco-system</i>	42
5.2. ILEARNRW SERVER - HARDWARE SPECIFICATIONS	43
6. PROJECT SOURCE CODES AND LICENSING	44
6.1. OPEN SOURCE PROJECTS	44
7. INSTALLATION INSTRUCTIONS	45

1. Introduction

This deliverable describes the final version of the integrated iLearnRW system prototype as it was refined by incorporating feedback from the evaluation phase of the project. A fully operational final prototype is delivered. The iLearnRW project provides tools for English and Greek linguistic experts/teachers as well as children struggling with dyslexia. There are two web pages supporting the experts/teachers while a serious game and a reader has been built especially for children with dyslexia. The purpose of the iLearnRW is to offer children a set of tools to improve their linguistic skills, especially for students of ages 9-11 with dyslexia/dysorthographia.

The big picture of the iLearnRW system¹ can be seen in Figure 1. As it is easy to see the system is based on client-server architecture.

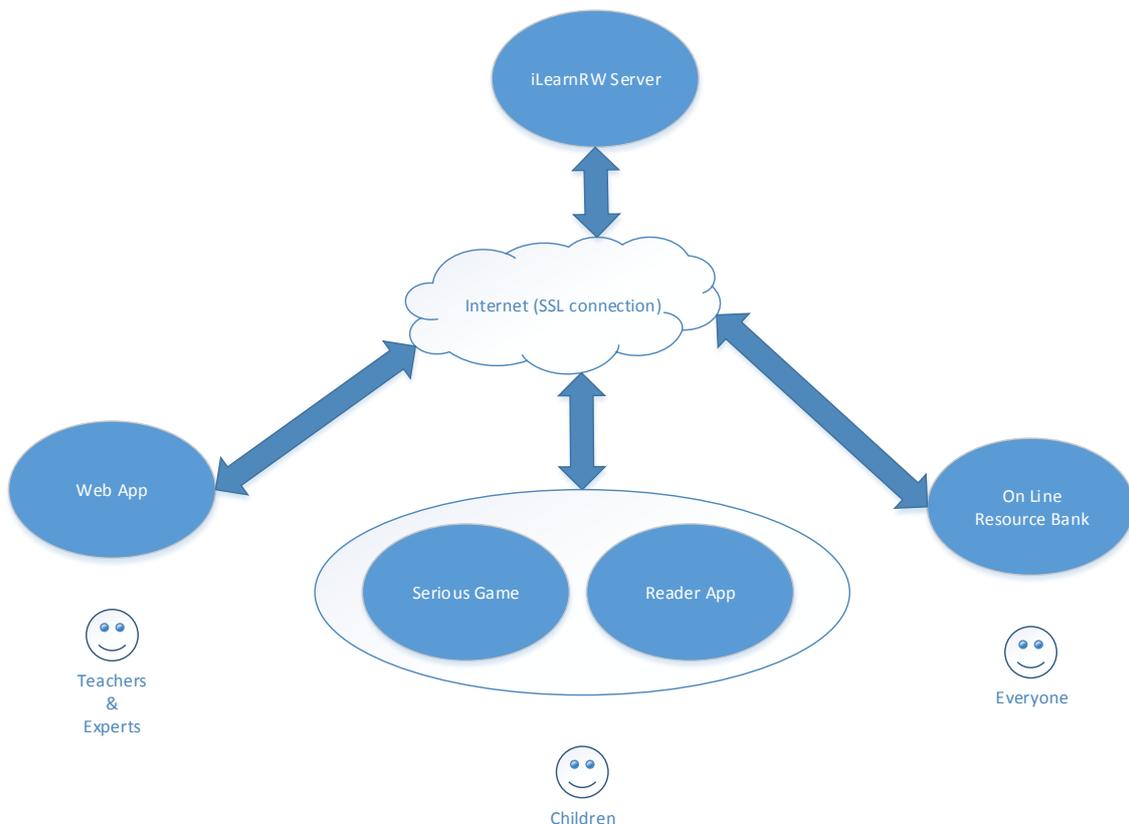


Figure 1. Overview of the Final System

Most of the iLearnRW components have already been introduced in deliverable D6.2 where the main components of the software were detailed. The activities of the last year involved:

- Development of the On Line Resource Bank (ORB) web page
- Creation of a special application for extracting statistics from the system's database and support the evaluation.
- Upgrade of the English and Greek dictionaries
- Refinement of system components based on the linguistic experts input
- Maintenance/updating of the iLearnRW central server

The iLearnRW system was evaluated in English and Greek schools (for students with dyslexia) during the last year of the project.

¹ A DVD including all the source code and resources (dictionaries and images) is attached

The remaining of this document is structured as follows: Section 2 provides a brief overview of the iLearnRW system. In section 3 we present a detailed description for each one of the system components; especially in section 3.1 we outline the system refinements compared to the Y2 version. Section 4 describes two of the most used components of the iLearnRW system, the English and Greek special dictionaries. Section 5 contains the system's overview from a technical perspective while section 6 contains information about the licensing and the source codes of the different iLearnRW components. Finally, installation instructions about the client applications can be found in section 7.

2. iLearnRW System Overview

The iLearnRW project includes next generation learning software which uses a tablet/computer to facilitate the learning process for children with dyslexia and/or dysorthographia.

The main components/features of the *Intelligent Learning Environment for Reading and Writing* (the iLearnRW system) are the following:

- *Linguistic user profile*: iLearnRW keeps track of the history for each student as well as her/his linguistic strengths and weaknesses exported from special tests designed by experts
- *Engaging learning activities (serious games)*: the system includes a profile-aware serious game which induces effective learning for children with dyslexia/dysorthographia. The last is achieved through nine profile-aware mini-games asking the user to complete several language tasks. The sequence of the games played and the displayed content was carefully selected using learning/teaching strategies.
- *Game usage logging mechanism*: a module that is being used to store all users' history in a central database. Independently used and accessible through a public API allows the games to log all user's actions.
- *Text classification based on user profile*: a module that ranks content/text appropriateness as learning material for a specific user with reading/writing difficulties (based on her/his profile).
- *Personalized content presentation*: iLearnRW contains a dyslexia-friendly reader that supports reading using content adaptation and presentation mechanisms (including text-to-speech) which, in conjunction with individual user profiles results in content presentation that suits each individual user
- *Evaluation of learning*: the system offers the experts the possibility to view the users' profiles and history (logs). An expert can evaluate her/his students' progress individually or as a group by creating several reports.
- *On-line resource data bank*: a web application including most features of the iLearnRW system making them available to the public. The user can classify or annotate texts as well as preview user profiles.

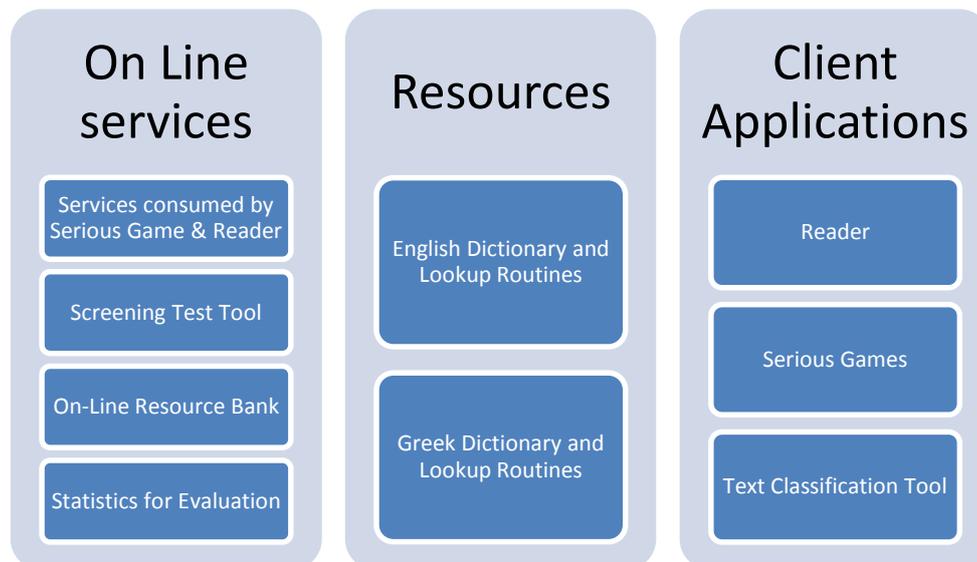


Figure 2. Project components' overview

2.1. Target Groups

There are two main groups of people taking advantage of the iLearnRW features. Students can improve their linguistic skills (especially children with dyslexia/dysorthographia) by the usage of the serious

game and the reader. The system also simplifies the life of the linguistic experts (dyslexia experts, speech therapists, etc.) since it can be used to setup user profiles, preview the students' progress and classify the learning material (based on difficulty).

The value that the software adds to each user category can be summarized as follows:

- **Students of Ages 9-11 (with Dyslexia)**
 - Screening Test: A special test created by the linguistic experts containing questions suitable to recognise any kind of dyslexia. Based on each student's answers to this test the system creates a *Dyslexia User Profile* for the student and uses it to optimize the learning experience of the user to the Reader App and the Serious Game.
 - Reader App: the text annotation feature is able to highlight word parts based on linguistic rules (eg. suffixing) so that the reader learns to recognise them.
 - Serious Game: there are 9 mini games asking the user to complete tasks concerning the word structure (eg. split word into syllables) so that the user strengthens her/his knowledge on the corresponding linguistic rules.
 - On Line Resource Bank (ORB): a user can check any text and get information about the structure of each word. There is also a map of common linguistic difficulties where the user can check which words of the text matches to each of these difficulties and why. Another feature of the ORB is the text annotation functionality which works as in the Reader App.
- **Students of Ages 9-11 (generic)**
 - Serious Game, Reader App & ORB: All three applications can be used from students to help improving their linguistic skills.
- **Experts/Teachers:**
 - Web Application: This application is the admin interface for the linguistic experts. Here the expert can create teacher and student accounts and preview the history of her/his students. There are special reports and graphs showing the progress of each student or total statistics for each school. In this site the expert is responsible to insert manually the answers of the student to the screening test in order to initialize the user profile.
 - On Line Resource Bank: The expert can login to the ORB and quickly check a student's profile. She/He can also check the difficulties of a text and quickly see if it has a suitable structure for his students. Finally, the expert can easily create linguistic exercises by hiding parts of words (using a special text annotation filter) and then ask the students to fill the gaps.

The iLearnRW system has been evaluated through extensive field-testing in two language settings (UK and Greece). The evaluation of the ILearnRW system was based on the analysis of the data collected during the implementation of the UK and Greek test-beds.

2.2. Mode of Operation

Naturally, children having dyslexia are first identified by their parents or teachers. Then a linguistic expert may work with the child.

1. **Users Setup:** A dyslexia expert creates a group of users (including students and teachers)
2. **Profile Initialization:** Each student takes a special test and the dyslexia expert (or teacher) imports the answers to the iLearnRW system (the system automatically determines the user's strengths and weaknesses and the profile is created)
3. **Serious Games and Reader:** After a profile is created for the student, then she/he can use the serious game or the reader which are both user sensitive (since the user logs in first and the apps fetch the user profile).
 - Serious game: the game asks the user to complete several linguistic tasks (presented as mini activities) and keeps track of the user's responses. If needed the game updates the user profile (redefinition of the user's strengths and weaknesses).

- Reader: the reader presents annotated texts to the user, this means that the reader highlights some specific word parts (i.e. suffixes, prefixes etc) making the student to get familiar with the corresponding language rules.
4. **Student Evaluation and Teacher Support:** A teacher can at any time preview the student's process and also predefine the future mini-activities that the system will ask the user to complete.

3. Description of the iLearnRW system

In this section we describe the iLearnRW system as it is by the end of the third year of the project. An earlier revision was previously delivered in the D6.2, describing the main parts of the system. Therefore, we start by making a development summary for the third year of the project in which we present the refinements and the new components. Then we describe each one of the iLearnRW applications in separate subsections.

3.1. Year 3 – Technical Development Overview

This section includes a summary of the components refined or developed during the third year of the iLearnRW project. We also give a brief explanation for each one of the updated applications.

Serious Game

The development of game during the last year has focused on improving its usability and accessibility for players and developers. On the player side, the mechanics and user-interface of all game scenarios, the world view and social network were polished, following the feedback collected from numerous users. In addition, a system for automatic updates was integrated into the game, allowing selected devices to receive refined versions of the software as soon as changes were implemented and tested. On the developer side, the game was refactored to reduce coupling and improve cohesion, and in general clean up the code to facilitate the task of modifying or extending the software. Furthermore, a code repository with documentation was created to distribute the source code and assets developed during the project.

Reader

The reader application was also updated in during the last year of the project. According to the update the reader now works also in off-line mode. It can also make suggestions to the user about annotation rules based on her/his profile. We also added a set of predefined annotation presets to provide fast and meaningful options for the text annotation.

Text Classification Tool

The text classification tool was updated/recreated and it is now a Java application that operates using the central iLearnRW server for the analysis of the texts. The last improves the client-side performance since the user's computer is only responsible to display the text analysis and not to perform it. The text analysis tool now supports text annotation and can also show information about the real users of the system (profile, updates, etc.). Last but not least, we also improved the presentation of the text metrics with special graphs.

On Line Resource Bank

The ORB was completely developed during the third year of the project. The full description of the application can be found section 3.4.

Application to Extract User Statistics from the Datalogger

This application was also completely developed during the third year of the project and is fully described in section 3.6.

Dictionaries

In the last year of the project we made a major update to the dictionary resources. We started by developing a special application that sorts words based on their structure, frequency, length and other characteristics. The last routine was heavily used in order to sort the game content (words) by difficulty allowing the game to ask for “easy” or “difficult” words for each particular linguistic area. For the Greek version we followed a slightly different approach; a special mechanism was developed to store words

based on their CV structure². The experts then sorted the structures by difficulty based on their experience. And the final step is that the games can get not only words based on the index-category information but also based on the CV word structure which implies difficulty.

3.2. Serious Game

Below we briefly describe the serious game included in the iLearnRW applications.

The iLearnRW serious game is inspired by the famous Mexican Día de los Muertos celebration. The core premise is that the dead have unfinished business with the living, largely regarding communication. The player has been gifted the skill of being able to see, hear, and talk to the dead, unlike most other living characters in the game world. It is therefore up to the player to help characters. Much of this help takes place through game challenges, which require the practice and strengthening of linguistic skills. The iLearnRW game is consisted of two modes (Adventure and Play mode), both of them take place in the Día de los Muertos world. There are nine mini activities available in the game world. Note that the first time a player encounters each challenge an animation explains the goal of the challenge, and how to play.

3.2.1. Game World

The game world is presented as a vast city street map where houses and other constructions are connected through paved roads. The game starts inside a house of the player which is a special location that allows the player to select an avatar and customize her/his aspect (Figure 3). Then player controls the avatar which is always centered on the screen; to move, the player has to tap on a location of the road, which will prompt the avatar to move automatically to that location. The map is not scrollable, the last forces the player to choose her/his path step-by-step to reach any location outside the view; this decision reinforces the feeling of being playing a game rather than only selecting literacy activities.



Figure 3. Start screen of the serious game

The world view (Figure 4) contains nine special locations that represent each of the game scenarios. Each special location has a main character attached to it, and that location will be locked until the player meets the representative character (i.e. until the player's profile is strong enough to start learning the corresponding linguistic area). These characters will expose the background story behind that location, introducing the player to the world's lore, mythology, and rules. The encounters with these characters are directed by the game alone to ensure a coherent and paced discovery of the game world (i.e. undesired behaviors such as meeting all characters within the first hour play can be avoided). Once the world is introduced, a procedure used to create encounters, this is, the language area, difficulty and game recommended by the server will be used to direct where a character will show up and what game will ask the player to play. After helping a character, it will hover around the same location for a while

² CV structure: replace all word letters according to the rule: consonants replaced with a 'C' and vowels with a 'V'. For example word 'hello' is converted to 'CVCCV'.

allowing the player to re-play that particular game. In general, players are allowed to ignore characters' requests and instead exploring the world to talk to other characters that they find more interesting. The pedagogical module can still be used to influence the characters that are present at any given time in the world, limiting the free play feature to a number of options.

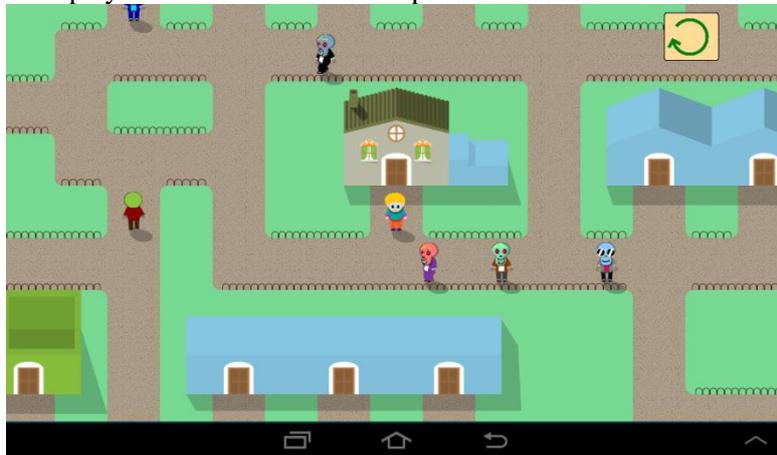


Figure 4. The game world

Interactions with characters can be initiated by tapping on them. In order to reduce the amount of text used in the game we decided to implement these interactions using voice recordings. Motivated in part by budget restrictions but also to reinforce the concept that the player is the only person that can understand the characters in the world, the characters only speak through speech bubbles that contain images or symbols. Then the voice of the avatar responds with an explanatory remark. For instance, the owner of the junkyard can initiate the conversation with a speech bubble displaying the post office, and the avatar may reply by saying "Oh, so you need a hand with your mail? Sure!". As the number of characters is large and all the narrative need to be accessible in Greek and English, the solution of having only the avatar voiced reduces the number of different voices needed to four (English and Greek male and female voices).



Figure 5. Some details of the friend just visited (after the end of the mini-game)



Figure 6. A new friend is unlocked after a successful attempt in a mini-game

3.2.2. Mail Sorter

Activity Description

In Mail Sorter, the player's objective is to assist a mailroom clerk who has become overburdened with a surplus of Día de los Muertos mail from the living to their dead relatives. Letters and packages come out of a conveyor belt, and pass through an X-ray machine, which shows a fragment of their address. Nearby, postal delivery workers sit on a line of chairs, waiting for enough mail to collect up in their mailbags to go out and do a mail-run. Mailbags may be labelled with any of the following: vowel sounds, suffixes, prefixes, letter names, letter patterns, graphemes or phonemes (for grapheme/phoneme matching), irregular/sight words, confusing letter shapes. These patterns map to ones that can be seen on the packages and letters through the X-ray machine. If letters are not dispatched within a certain amount of time, they disappear - since they are unstable objects that exist between both worlds.

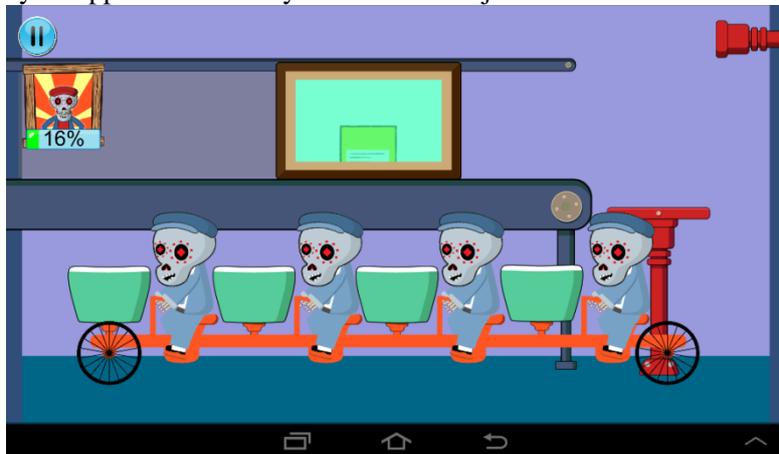


Figure 7. Mail Sorter

Feedback and reinforcement

In this game the player is forced to discover all the correct matchings before moving on to the next set of words/patterns. Additional feedback is not required.

3.2.3. Bridge Reinforcer

Activity Description

In Bridge Reinforcer, the player's objective is to assist travelers who have become stuck on a broken bridge. They do so by completing words featuring missing segments of the bridge (made of squared blocks), thus creating a stable road for the characters to travel. The characters are always moving to the right, so the player must continuously complete words to help the travelers.

Placing incorrect segments into the bridge causes the segments to drop straight through making the bridge inaccessible.



Figure 8. Bridge Reinforcer

Feedback and reinforcement

After the student has highlighted parts of the word, she/he is guided (from left to right) through the mistaken segments. Then, all correct and incorrect blocks are clearly indicated, again from left to right. Finally, the student has to fix her errors.

3.2.4. Harvest

Activity Description

In Harvest, the player's objective is to extract all the possible goodness out of words that keep flowering out of the ground, in order to obtain as much "word essence" as possible. Word essence is a very important magical substance, essentially serving as the power supply of the world. Each flowered word can be characterized in different ways: for example, "happening" has 3 syllables, begins with h, has "ing" as a suffix, etc. The player has to identify characteristics of the word in accordance with the characteristics collected by various characters/machines surrounding the flowered word. These characters/machines are labelled with certain characteristics, including numbers of syllables, prefixes, suffixes, and so on. For example, there may be 6 characters/machines surrounding the flowered word "happening", which in turn may be labelled with: "1 syllable", "2 syllables", "3 syllables", "ist", "ing", and "ism". The characters/machines can only extract word essence from correctly matching flowers, while any attempt to extract word essence from an incorrectly matching flower can incur damage on the character/machine - encouraging the player to not try and match words with every character/machine. Essence can only be extracted from a character/machine once per word. Once a player decides that none of the characters/machines surrounding the flower can perform further extractions, they can summon a next wave of characters/machines to perform different extractions. For example, a next wave for "happening" might include "begins with h", "begins with b", "contains ben", "contains pen", "ends with d", "ends with g". The player has 60 seconds to extract as much essence as possible from a word before it wilts. Wilted flowers can no longer have extractions performed on them.

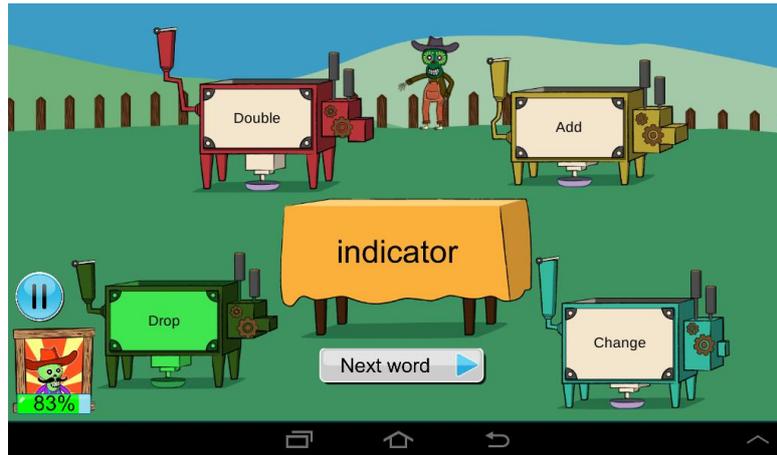


Figure 9. Harvest

Feedback and reinforcement

In this game the player is forced to discover all the characteristics of the word. Additional feedback is not required.

3.2.5. Serenade Hero

Activity Description

In Serenade Hero, the player's objective is to assist a character in singing a song to his sweetheart, who is still alive. The character and his band are singing below the window of the sweetheart. A line of text is on screen, with words missing. The player needs to fill in the missing parts from a selection of words visible at the bottom of the screen. The sentence keeps refreshing, with new words available. A musical pacer is slowly along the line, from left to right, and if the sentence is deemed correct, the character and his band will continue singing. The longer the chain of correctness, the more complex and interesting the music sounds and the animations are. For example, along a chain of correctness, the game will result in: a drumbeat, an accompanying instrument, another accompanying instrument, back up dancers, backup singers, the sweetheart appearing in the window, bystanders gathering, and the sweetheart bobbing her head to the music. As the game gets harder, the speed of the pacer will increase.



Figure 10. Serenade Hero

Feedback and reinforcement

The tally displayed after the game shows the player what she/he did wrong.

3.2.6. Moving Pathways

Activity Description

In Moving Pathways, the player's objective is to help two characters be reunited. The player is faced with a grid of hexagonal tiles, all of which contain a single letter, but many of which are letters that can be confused with one another (e.g. b, p, and q). The player must create a path of movement based on a particular letter (i.e. the target letter) to reunite the characters. But no contiguous line of tiles featuring the target letter exists. The player needs to perform rotations on some tiles in order to continue the path. Only blocks around target letters can be rotated.

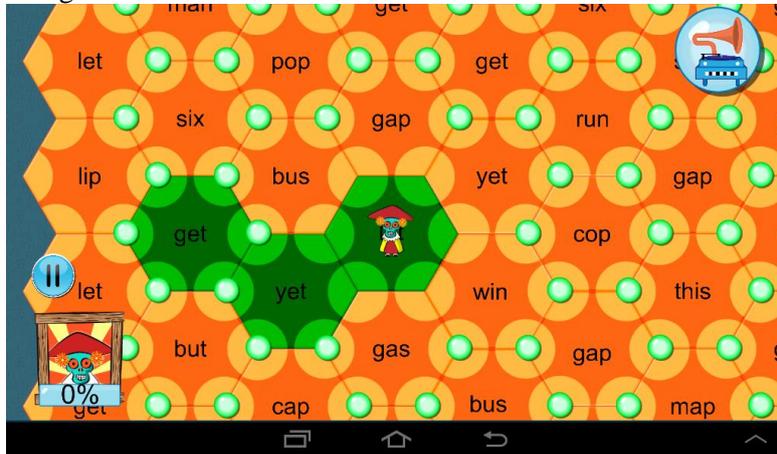


Figure 11. Moving Pathways

Feedback and reinforcement

The path of correct patterns tapped is maintained in green, facilitating that the student keeps track of all the right decisions made so far.

3.2.7. Solomon's Junkyard

Activity Description

In this activity the player's objective is to perform segmentation on words that are dropping down the screen. When correctly segmented, words break up into groups of cubes. For example, "watermelon" when correctly segmented turns into: wa-ter-me-lon, which after a couple of seconds, turns into four objects of size proportional to the size of each syllable. When in block form, block groups can be shunted around. Incorrectly segmented words, and also unsegmented words, cannot be broken up and fall straight down. When a complete line is made (of blocks and unsuccessfully segmented words or word parts), the line clears. The round is over when the top of the screen is reached by a block, a word, or a word part.



Figure 12. Solomon's Junkyard

Feedback and reinforcement

Once the segmentation time is up, the player can observe that the syllables of the word are transformed into objects one by one, in the correct order. Furthermore, if some splits are missing, a skeleton hand splits those syllables for the player.

3.2.8. Whack a Monkey

Activity Description

A pattern is presented on a prompt card (e.g. sound, letter, word or picture) and several monkeys appear holding cards with letters or words either matching the pattern or not matching the pattern. Player has to hit the monkey matching the pattern and not a monkey not matching the pattern.

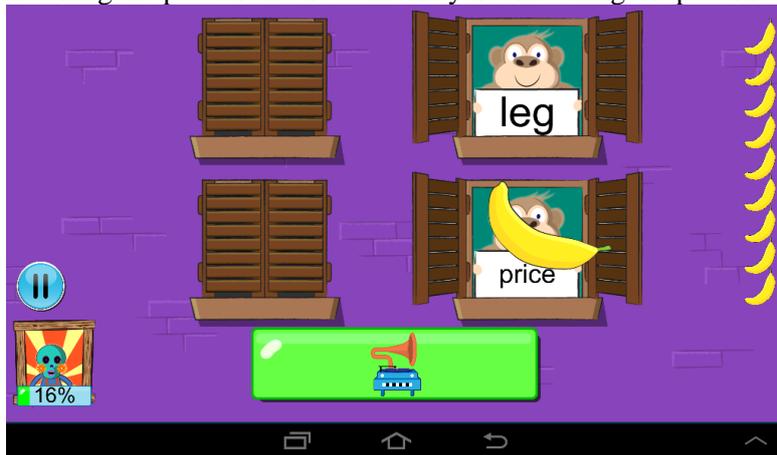


Figure 13. Whack a Monkey

Feedback and reinforcement

The fast pace nature of this game does not allow for immediate feedback other than right/wrong.

3.2.9. Mail Delivery

Activity Description

In Mail Delivery, the player has to help a postman to deliver his parcels while avoiding playful monkeys that try to capture the parcels. The player has to first avoid monkeys roaming the streets to reach a door; once at the door, the player should knock on the door as many times as the number of syllables.

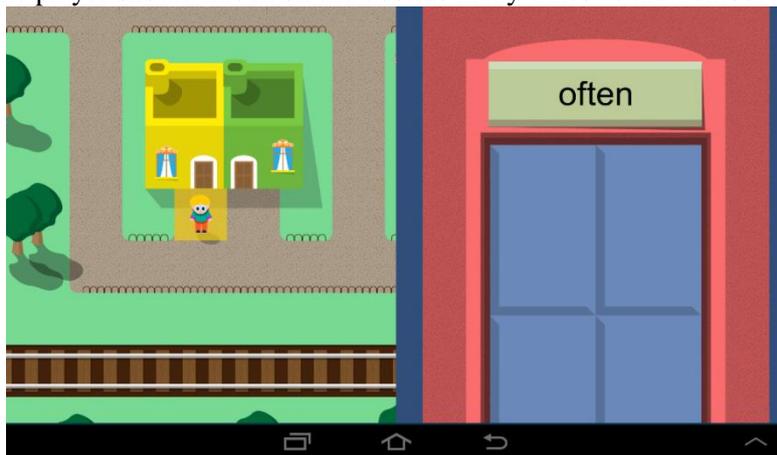


Figure 14. Mail Delivery

Feedback and reinforcement

The target pattern can be written or spoken text (played whenever the player hits the speech gadget). In addition, this activity adds touch as an additional sense, as the player uses tapping not to select an item on the screen but to express a number and rhythm. A knocking sound is played with each tap, further reinforcing this action.

3.2.10. Typing Train Dispatcher

Activity Description

The player is shown a train station and given the task of loading and dispatching trains from it as fast as possible to avoid collisions. To dispatch a train, the player has to load it with the appropriate cargo. To load the cargo, the player uses the keyboard to type the appropriate language into each carriage. For instance, if the player is presented with the word "syllable" (by voice, writing or both), they tap the first carriage and type in 'syl', then the second, type in 'la', etc. This same thing can be done with letters, sounds (i.e. all letters representing a sound would fit in the carriage), suffixes or whole words.

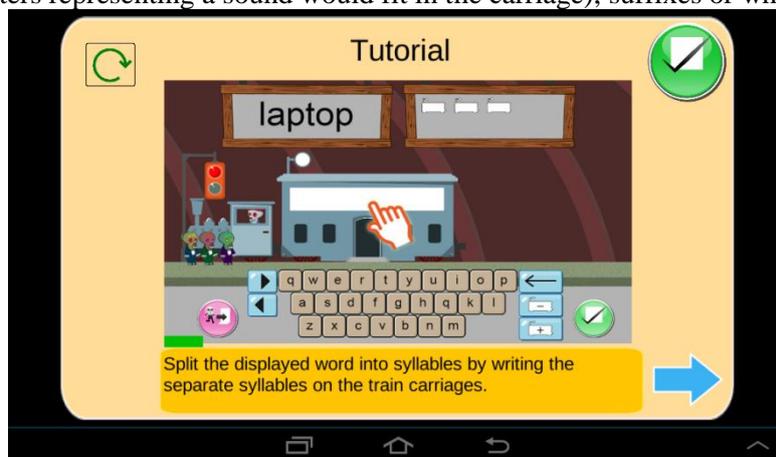


Figure 15. Typing Train Dispatcher

Feedback and reinforcement

In this game the player is forced to type all correct syllables of the word. Additional feedback is not required.

3.3. Reader

The iLearnRW also offers a special dyslexia-friendly reader application. This can be considered an “auxiliary” application compared to the Serious Game.

The reader is an application that allows a user to read digital texts. The user first imports the texts to the system and then he/she is allowed to use a set of text annotation rules that address certain dyslexia problems while reading. Annotation rules include a set of possible language difficulties (i.e. suffix problems on ‘-ed’ endings, or problems on specific syllabification patterns, etc.) along with the annotation style that the user prefers to them inside the text.

The application contains three main activities, a library, an annotation rules manager and the main reader, which helps the student manage his/her books in an orderly fashion and reading in accordance with the student’s needs.

Library

The library is the place where the user’s books are displayed in alphabetical order. The user can manage her/his books by adding or removing items from her/his library. When a user adds a book then he/she can tap on it and read it on the tablet’s screen.



Figure 16. Reader Library View

Presentation (Annotation) Rules

Within the ‘Library’ screen you also have the option to adjust the presentation rules of the text. These allow you to customize how specific word segments and/or whole words are colored or highlighted. For instance you can choose to highlight a specific letter combination or a particular prefix or suffix.

The annotation rules correspond to linguistic areas and help the reader to emphasize on special structures words while reading. The user has the possibility to setup annotation rules or to adopt the system suggestions on rules based on her/his profile. Another possibility for the user is to select sets of rules that are predefined by the iLearnRW dyslexia experts (containing common dyslexia problems colored

in appropriate style). If the rules are applied, then all words in the text which have structure that matches to an active rule characteristics will be highlighted in accordance with the specified colors.

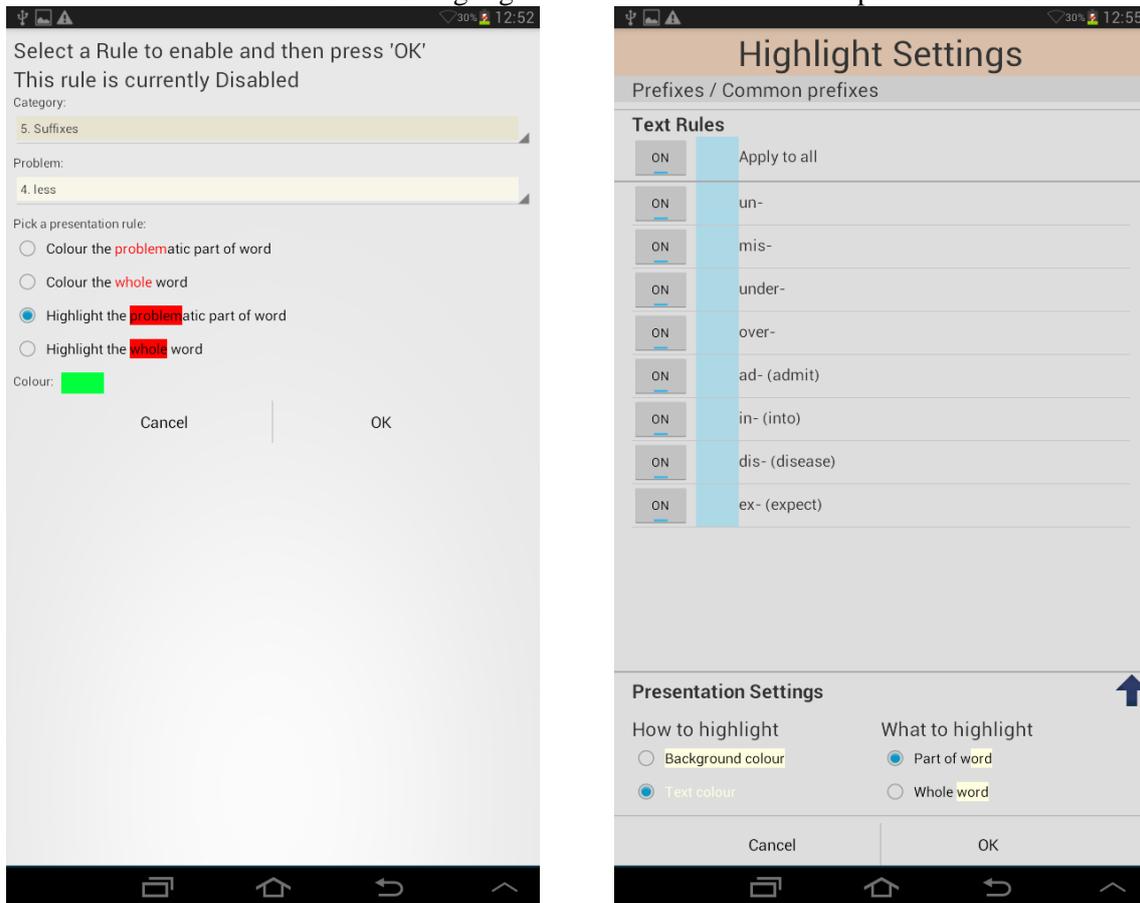


Figure 17. Annotation Settings pt.1

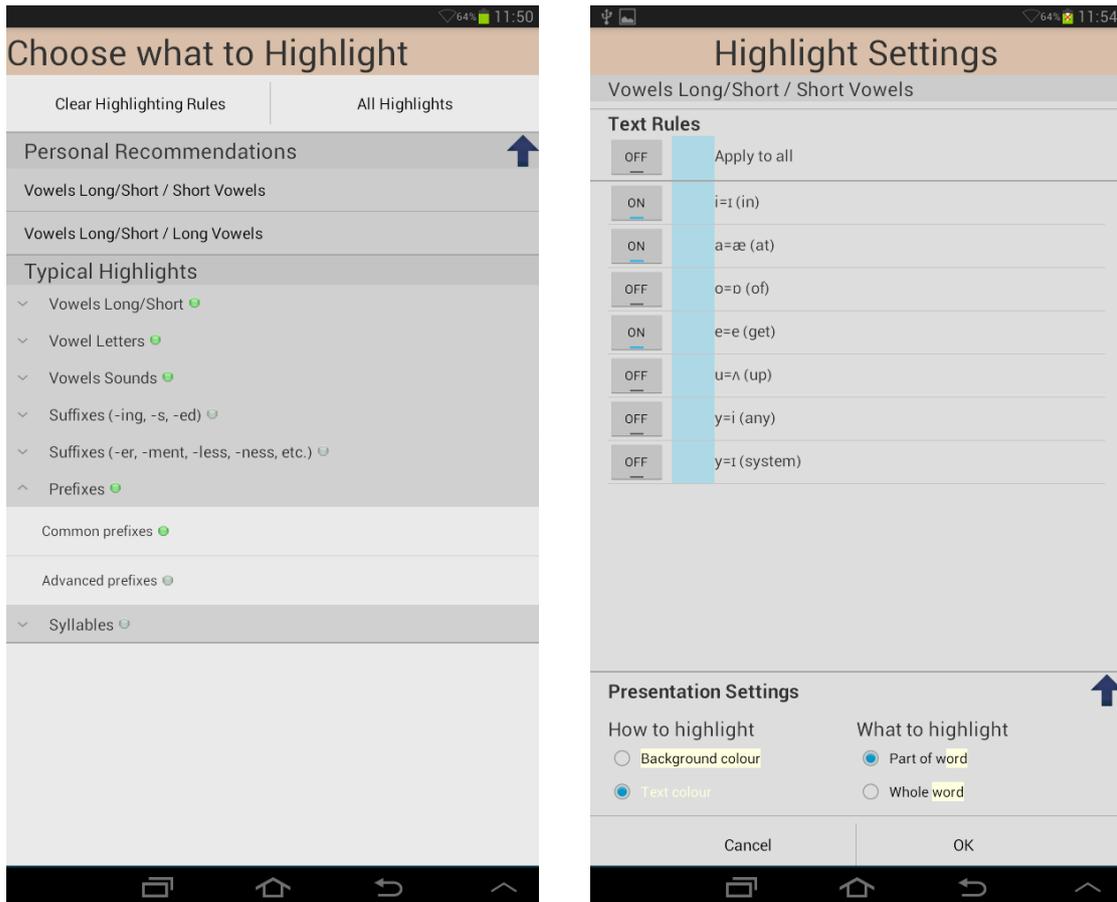


Figure 18. Annotation settings pt.2

Reader

The reader is the screen where most of the user's time will be spent. This is the view that shows the selected text. Note that if a user chooses to apply the special rules, highlighting will appear on words that match the linguistic issues that the rules are about.

The reader supports the following functionality:

- Mode switching: Two different modes are supported by the reader, listen and guidance. These modes are used to help the user with the process of reading and allow him/her to focus on a small part of the text.
 - Listen: Allows the user to listen to the text by using the text-to-speech functionality of the Android operating system.
 - Guidance: Guides the user forward with the use of a highlighted word which moves forward according to a speed slider.
- Search: Allows users to search within the text for given search string and also lets them search between the next and previous occurrence of inputted string.
- Play/Previous/Next: Depending on which mode the user has selected, the buttons will work differently.
 - Listen mode:
 - Play – Starts or pauses text-to-speech read
 - Previous – Moves highlighting to previous sentence
 - Next – Moves highlighting to next sentence
 - Guidance mode:
 - Play – Starts or pauses guidance highlighting
 - Previous – Moves highlighting to previous word
 - Next – Moves highlighting to next word.

- Tap & Hold on a word: If a user tap & holds on a word a popup will appear that gives more information about the selected word.

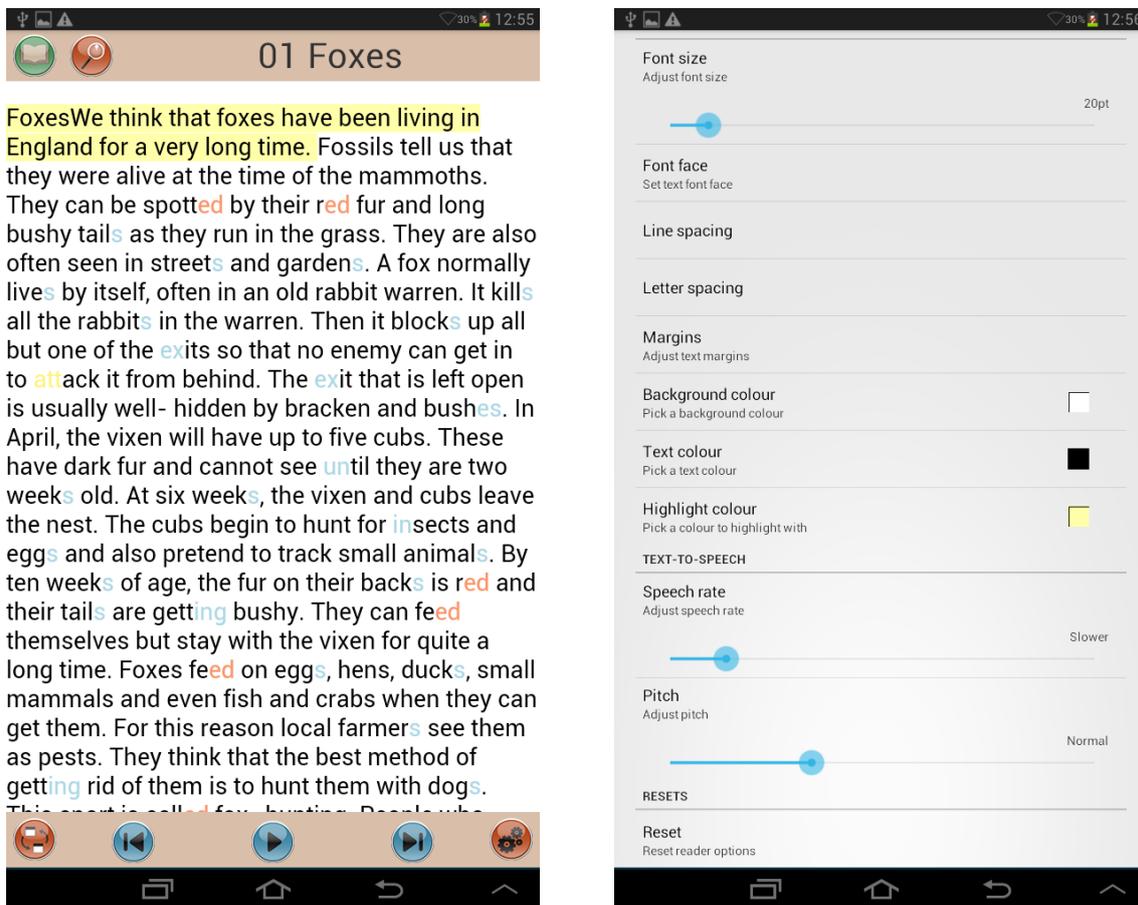


Figure 19. The reader and the settings screen

Text Settings

Within the 'Text Settings' menu the user can adjust the appearance of the text in reader. A list with all available options follows:

- Select font size
- Select font type (there are 5 different dyslexia-friendly fonts).
- Adjust how much space there is between the lines of the text
- Adjust the space between individual letters by tapping on the Letter spacing
- Change the amount of space on either side of the text
- Select your preferred colors for the background, text and highlight (when text is being read aloud)
- Adjust the speed and pitch of the voice that reads the text

3.4. On Line Resource Bank

The ORB is an application offering the iLearnRW functionality and features to the public. A user can chose between working as an iLearnRW member or as a guest. The last allows the public to work in guest mode and use all the features of the software. The user can preview a set of student profiles or analyze/annotate texts. In the next sections we go through the different features of this app. The application can be accessed on line under the URL <https://ssl.ilearnrw.eu/orb/>.

3.4.1. Profile Viewer

Selecting the first tab of the menu the user can select her/his preferred profiles and view the details of each. Note that there are 3 different roles when viewing this page related to the type of the logged in user:

1. Logged in as an expert/teacher: The system displays on the left the full list of expert's/teacher's students
2. Logged in as a student: The system displays on the left only the current student's name
3. Not logged in: The system displays on the left two generic profiles, one English and one Greek profile.

In Figure 20 we present the page contents when an expert is logged in and after selecting a user to view his profile.

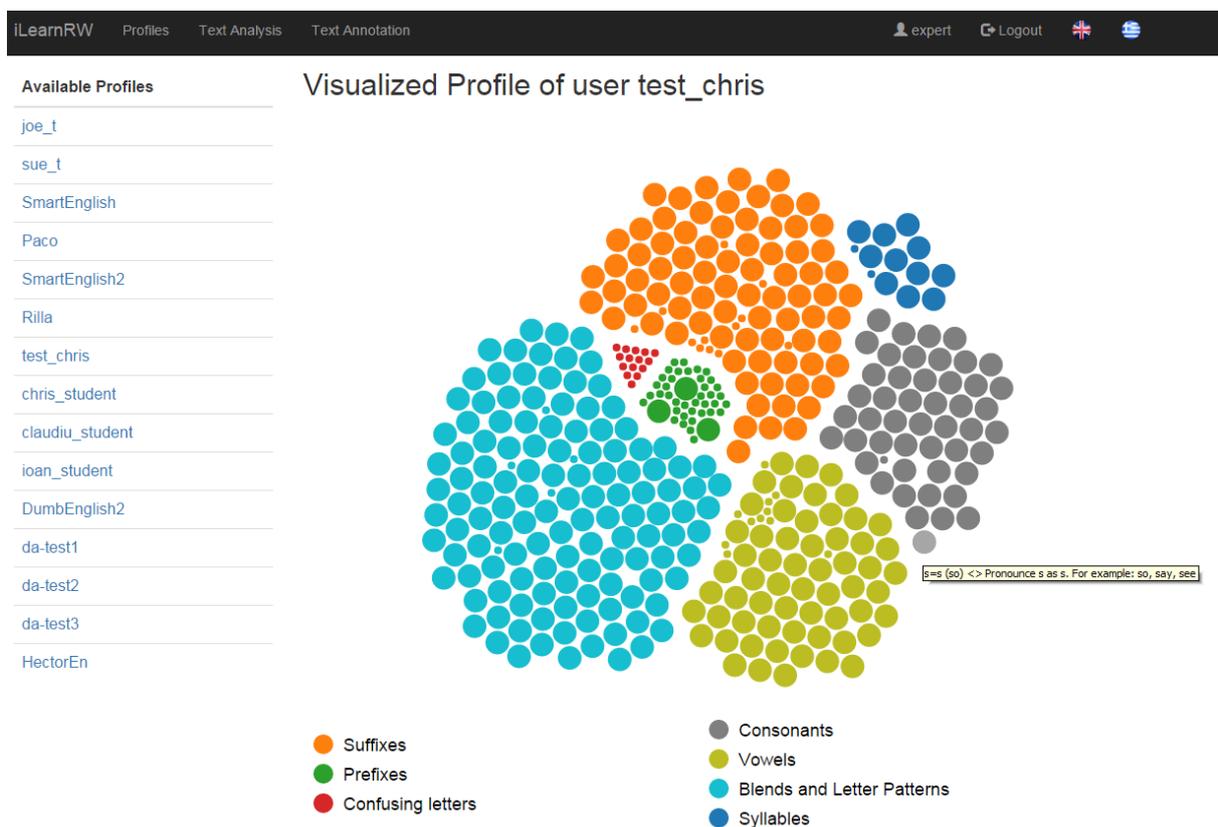
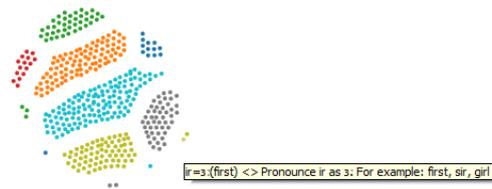


Figure 20. On Line Resource Bank - Profile Visualization

The expert can have a quick look to the user's profile through a special visualization. We display each language difficulty by a circle with size proportional to the severity. The circles are grouped by colours to indicate the language category they belong. When a user's mouse hovers a circle then a pop-up with details of the corresponding problem opens.

In Figure 21 we show a visualization of a non-dyslexic profile. Note that the area covered by the circles is negligible compared to the dyslexic profile presented in Figure 20.

Visualized Profile of user sue_t



- Suffixes
- Prefixes
- Confusing letters
- Consonants
- Vowels
- Blends and Letter Patterns
- Syllables

Figure 21. Non - Dyslexia Profile Visualization

In the same page we also present a table with the user profile entries. We show all language problems, grouped by category, linked to the corresponding severity of the student.

3.4.2. Text Analysis

In this section the user can analyze texts based on a selected user's profile. The user should select the preferred profile, insert the text and then press the "calculate metrics" button. Then the main metrics (generic and user related) are displayed in a table on the right side of the page (Figure 22).

In the same page a special heat map showing the amount of words matched to each language difficulty is presented (Figure 23). The box colors indicate the user's frequency on showing a particular problem (Red: Very Often, Yellow: Often, Blue: Rarely, Green: Never). Also, the numbers inside boxes show the number of words matched to a particular user difficulty. If the problem is green colored (i.e. no problem for the user) then nothing is matched. Finally, by clicking on a box you can see details about the corresponding difficulty as well as the text words matching to this difficulty.

Select a profile:
 test_chris

Insert your text here:

iLearnRW is a 3 year research project funded by the European Union's 7th Framework Programme: Information and Communication Technologies (ICT). The project started in October 2012 and will finish in October 2015 (36 months). There are 7 participants from 4 countries involved in the project.

Aims
 The iLearnRW project aims to develop next generation tablet-based software that will support and motivate children with dyslexia aged 9-11 with their reading and writing. The software will incorporate an assistive reader, providing tailored reading support as well as literacy-based activities targeted at each child's specific difficulties. It will also incorporate a learning game providing further opportunities for children to practice specific areas of reading and writing within a fun and engaging game environment.

It is intended that the iLearnRW system will support the following specific features:
 user profiling.
 learning/teaching strategies.
 classification of learning material based on user profile.
 personalized content presentation.
 engaging learning activities.
 the evaluation of learning.
 an on-line resource data bank.

After development the software will be evaluated through extensive field-testing in different language settings (UK and Greece) to establish its overall success in promoting the reading and writing skills of children with dyslexia.

Calculate Metrics

Generic Metrics

Property	Value
Number of Words	198
Polysyllabic Words	46
Number of Sentences	14
15 or more word Sentences	6
Number of Paragraphs	4
Average Word Length	5.78
Average Number of Syllables	1.85
Average Words per Sentence	14.14
Flesch-Kincaid	11.80

User Related Metrics

Property	Value
Text Difficulty	2,625.50
Difficult Words	126
Very Difficult Words	103

Figure 22. Text Analysis 1 of 2

Consonants

52 1 20 4 53 0 26 1 5 31 0 0 35 0 14 1 7 0 23 10 19 2 0 6
 0 3 15 0 20 1 2 13 15 2 0 0 1 4 5 0 0 0 0 21 6 0 18 0 2

Vowels

61 22 14 8 17 2 3 0 4 0 1 0 2 6 0 4 0 0 0 0 7 0 0 0 1 1 2
 1 0 0 6 0 0 0 0 0 1 0 0 0 0 0 5 1 0 0 0 0 0 0 1 0 0 0 0 0 0
 0 1 0 0 0 3 0 0 0 0 0 0 0 0

Blends and Letter Patterns

0 5 2 3 0 0 0 8 0 0 0 0 0 11 0 1 0 0 3 1 0 0 0 0 0 0 0 0 0
 0 0 8 1 0 0 0 2 0 0 0 0 0 1 19 1 0 0 0 0 0 3 0 0 0 0 0 0 5 0
 2 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 1
 0 0 0 5 0 0 1 8 4 0 0 0

Syllables

95 40 17 15 15 0 10 0 1 57 25 16 5

Suffixes

10 4 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 5 0 0 0 0 0 4 2
 0 0 3 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 2

Prefixes

0 0 0 1 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 5 0 0 0 0 0 0 1 0
 0 3 0 0 0 2 0 0 1 0 1 0

Confusing letters

4 5 2 2 0 0 5 0 0 3 5 0 0 0 0

I=I (let) <> Pronounce I as I. For example: let, old, like

Word	Syllables	Phonetics
language	-lan-guage-	'læŋ.gwɪdʒ
development	-de-ve-lop-ment-	dɪ.'ve.ləp.mənt
line	-line-	laɪn
establish	-e-sta-lish-	ɪ.'stæ.blɪʃ
evaluated	-e-va-lua-ted-	ɪ.'væ.lju:ɪ.tɪd
learning	-learn-ing-	'lɜ:n.ɪŋ
material	-ma-ter-i-al-	mə.'tɪər.i.əl
field	-field-	fi:ld
child's	-child's-	tʃaɪldz
difficulties	-dif-fic-ul-ties-	'dɪ.fɪk.əl.tɪz
involved	-in-volved-	ɪn.'vɒlvd
profiling	-pro-fil-ing-	'prəʊ.faɪ.lɪŋ
children	-chil-dren-	'tʃɪl.drən
evaluation	-e-va-lu-a-tion-	ɪ.'væ.lju:'eɪʃ.ən
develop	-de-ve-lop-	dɪ.'ve.ləp
profile	-pro-file-	'prəʊ.faɪl
technologies	-tech-no-logies-	tek.'nɒ.lə.dʒɪz
also	-al-so-	'ɔ:l.səʊ

Figure 23. Text Analysis 2 of 2

3.4.3. Text Annotation

The last feature of the ORB is responsible for annotation texts based on a selected profile. The user should select the preferred profile, enable her/his preferred annotation rules and then insert the text and then press the “annotate text” button. Then a new page opens containing the annotated text (Figure 25).

Select a profile: ▾

test_chris, Difficulties List

Consonants

Vowels

1. i=ɪ (in) <> Pronounce i as ɪ. For example: in, it, if
2. a=æ (at) <> Pronounce a as æ. For example: at, as, and
3. o=ɒ (of) <> Pronounce o as ɒ. For example: of, on, not
4. i=aɪ (find) <> Pronounce i as aɪ. For example: find, child, hi
5. e=e (get) <> Pronounce e as e. For example: get, let, when
6. u=ʌ (up) <> Pronounce u as ʌ. For example: up, but, much

Blends and Letter Patterns

Syllables

Suffixes

Prefixes

Confusing letters

Insert your text here:

This site was developed as part of the [iLearnRW](#) project. It is a resource for teachers, parents and students who want help with phonics. Unlike standard text readability tools online, this site not only displays a total score for any text but also gives a [heatmap](#) of difficulties and can link any individual word to potential problems. It can also help by giving examples of words with similar features. You can check any word. Furthermore, you can learn more about phonics by seeing the big picture.

Active Rules

Quick Selections ▾

a=æ (at) <> Pronounce a as æ. For example: at, as, and

Highlight Problematic Word Parts ▾

ar=ɑ (far) <> Pronounce ar as ɑ. For example: far, car, art

Highlight Problematic Word Parts ▾

aCe=eɪC (make) <> Pronounce aCe as eɪC. For example: make, take, same

Highlight Problematic Word Parts ▾

a=ɒ (what) <> Pronounce a as ɒ. For example: what, want, watch

Highlight Problematic Word Parts ▾

Figure 24. Text Annotation, Rules Setup

The page displays a detailed list of the user’s profile and by clicking on a red circled cross the linguistic problem enters to the “Active Rules” list. The active rules are linguistic phenomena that are going to be marked in the final text. The user can also modify the way each rule is displayed, for this purpose there are the following options:

- Nothing: The rule makes no changes to the final text
- Paint Problematic Word Parts: All words matched to this rule will be painted (only the letters, not the background) using the selected colour.
- Paint Problematic Words: All parts of the words matched to this rule will be painted using the selected colour.
- Highlight Problematic Word Parts: All words matched to this rule will be highlighted (change the background colour) using the selected colour.
- Highlight Problematic Words: All parts of words matched to this rule will be highlighted using the selected colour.
- Hide Problematic Parts: All parts of words matched to this rule will be left blank using the selected colour (can be used to create exercises).

This site was developed as part of the iLearnRW project. It is a resource for teachers, parents and students who want help with phonics. Unlike standard text readability tools online, this site not only displays a total score for any text but also gives a heatmap of difficulties and can link any individual word to potential problems. It can also help by giving examples of words with similar features. You can check any word. Furthermore, you can learn more about phonics by seeing the big picture.

Figure 25. Annotated Text

3.5. Text Classification Tool

The text classification tool is another tool of the iLearnRW ecosystem; note that this is the only application that runs on a PC (Linux, Windows or Mac). The purpose of the application is to provide a way to evaluate texts' difficulties for a specific user. An expert can also use this tool in order to facilitate the selection of appropriate reading resources for children with dyslexia. The text classification tool loads the profile of an individual user and based on this it supports the following functionality:

- Profile viewer: Displays the user profile as well as a graph showing the profile updates.
- Word analysis: This feature allows an expert to insert words and then check the dyslexia problems (i.e. profile entries) that are relevant to both the word and the specific user.
- Text analysis: This panel displays information about a text compared against the problems of a particular user profile. It contains information on how many words have been matched to each one of the user profile entries and shows the part of the word that causes the problem. Also information about special metrics on the particular text is presented.
- File explorer with text ranking The file explorer displays a list of texts (text files loaded from a fixed directory) and displays some important text metrics to the user (number of words, number of sentences, difficulty based on the above algorithm, etc.). The user has also the ability to sort the texts by his/her chosen metric.

3.5.1. Home Screen

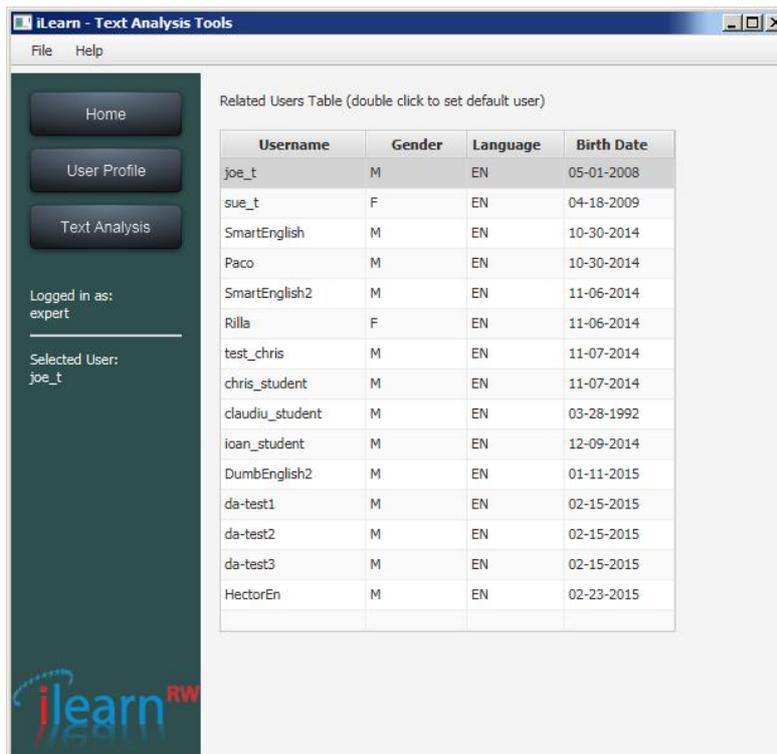


Figure 26. Home Screen Showing Linked Students

When the user starts the application she/he is first prompted to insert her/his credentials in order to fetch the full details from the iLearnRW server. Then a window opens with a small menu on the left and the list of the available profiles. Note that for the experts it displays the list of her/his students (Figure 26) while for a student it displays only the one corresponding profile. The user then can select a student's profile to work with by double clicking on the corresponding row of the student's table. Then, based on the selected profile, the user is allowed to i) view list of students (home), ii) view the details of the selected student's profile, and iii) analyse texts (classify or annotate). The two later options are explained in the next paragraphs.

3.5.2. Profile Viewer

By pressing the second item of the left menu a new view opens and displays the profile and a timeline graph with the updates (based on the serious game logs). The profile is represented by a two dimensional table. Each row of this table contains all linguistic problems that belong to a specific category while each cell represents a special problem of this linguistic area. In Figure 27 for example an English user profile is displayed, the fourth row represents the syllabification issues and the fifth cell represents the linguistic issue of syllabifying a word that follows the <vowel>-<consonant, vowel> syllabification pattern (e.g. a-ny, o-ver, a-bout). The severity of the user to each particular problem is displayed with a number inside the cell.

On the bottom part of the screen one can see the overall progress of the user expressed by the profile updates she/he gets while using the serious game. Each point on the graph represents the sum of the user severities at a specific date. Consequently, profiles with points closer to zero can be described as dyslexia-free while the problematic ones contain points away from zero.

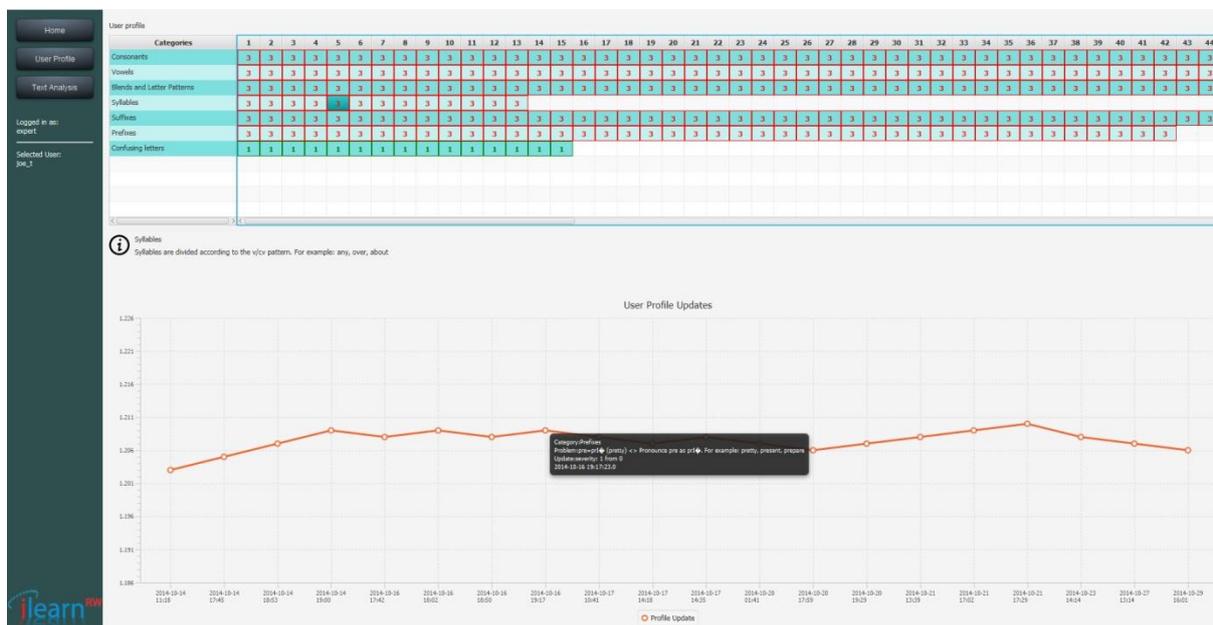


Figure 27. User Profile with Profile Updates Time Chart

3.5.3. Text Analysis

When the user selects the third item of the left menu a text analyser opens.

On the right part of the Figure 28 we can see the file manager of the application. This component makes use of the content classification module. So, we can see the different metrics (generic or user related) presented on the screen. By clicking on a text it opens in the text editor (center) and the user can press the “Analyze Text” button to get a deeper text analysis. After a text is analysed the user can navigate through three extra tabs, the “Phonics Heatmap” tab, “Text Analysis” tab and “Text Annotation” tab. The Figure 29 shows the phonics heatmap of the previously selected text. It is represented as a two dimensional matrix which displays either blue or variations of “cold” or “hot” colours. The rule to add colours to the matrix is, roughly, the following: the more the words of the text having structure that matches to the cell’s corresponding problem, the “hottest” the colour of the cell is

By clicking a cell one can see the description of the corresponding problem and also the words of the text matching this problem. An alternative analysis of the text problems against the user profile can be found in the “Text Analysis” tab where a pie representing the number of problematic words per linguistic category exists (Figure 30).

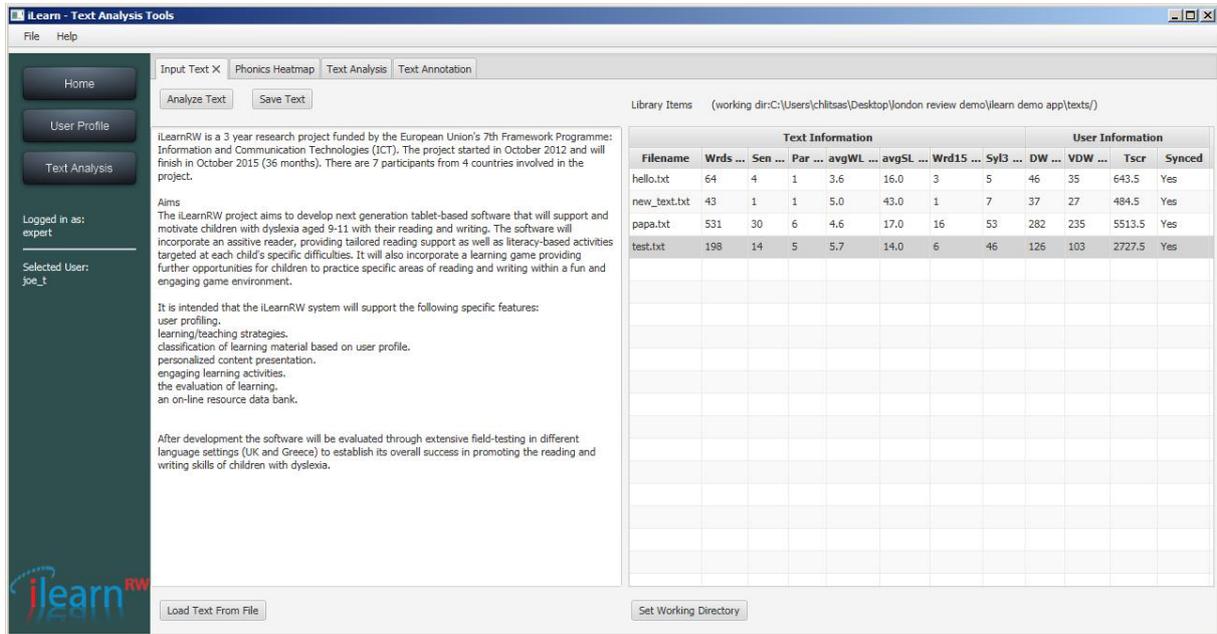


Figure 28. User Library of Texts

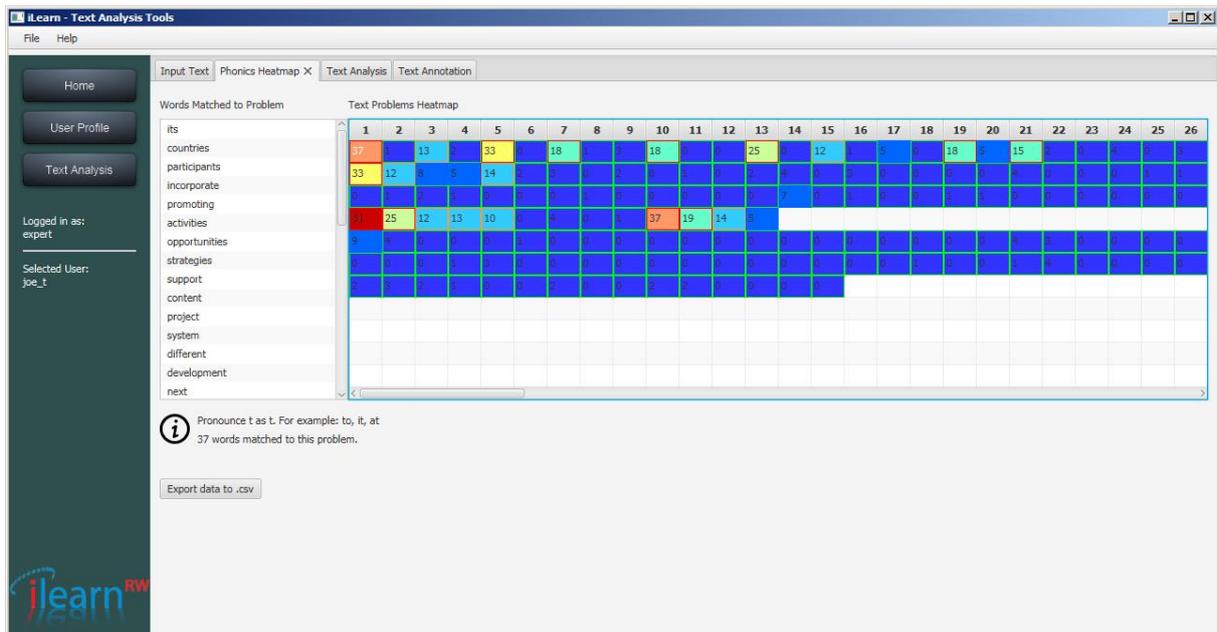


Figure 29. Heatmap of a text on a User Profile

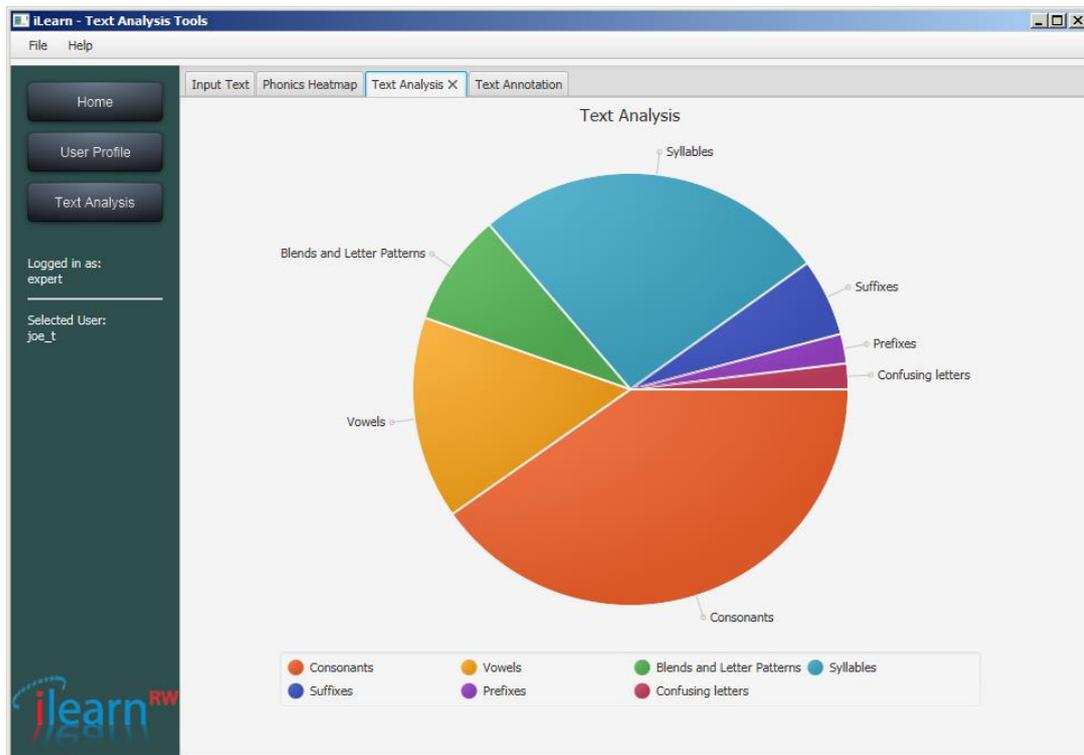


Figure 30. A Pie that shows the amount of User problems per Category

3.6. Data Analysis Console Application

During the 3rd year of the project the evaluation phase took place at English and Greek schools. Controlled groups of children were using the iLearnRW software (mostly the serious game and the reader) and the experts were able to monitor the children progress after that period. All collected data was stored in the central iLearnRW server since the games made use of the data logger service.

There are two kinds of data that we can examine a) the user profile and b) the game logs. The most valuable (and also most difficult) data to analyze is the game logs because they contain all the history of each student. An expert can keep track of a specific problem and see the student's answers on games presenting words taken from this problem. Also an expert may want to see the user's progress session after session, day by day etc.

Not surprisingly the data collected from the games in cooperation with the data logger were not ready for analysis. The above is mostly a consequence of the following errors that came up during the gameplay:

- Lost internet connection
- Game crashes
- Student stops (kills) the game without finishing the task

All above errors caused the inability of automatically identifying the *sessions* from the database because for some sessions the "session end" tag was missing. The solution to this problem was to create a special Java parser that is able to read all the data from the database and decide, out of the context, when a session starts, ends etc. We also used this console application to export the parsed data into special csv formats suitable for analysis in MS Excel or any other similar tool. Next we describe the data exported from the serious game logs and then the data from the screening test.

3.6.1. Parser for the Serious Game data

The console application is responsible to automatically create CSV files containing user logs. There are three different files for each language containing data grouped by i) Session, ii) Day or iii) by Week. Each CSV file contains the following information/columns:

- Username
- Teacher
- School
- Week
- Day
- Timestamp
- Time Played
- Complete Session
- Total Number of Words Played
- Number of Unique Words Played
- Success Rate
- Total Profile Entries Played
- Description of Profile Entries Played
- List of Words Played
- List of Apps Played
- List of Positive Profile Changes
- List of Negative Profile Changes
- Description of Positive Profile Changes
- Description of Negative Profile Changes

A screenshot of how the data looks like can be found in Figure 31.

- First row: Titles of all possible questions asked in the screening test. Note that each question contains a title and a set of linked words; we only show the title here.
- A.OK: number of words successfully answered for the particular question on the first screening test.
- A.All: the number of all words tested under the question on the first screening test.
- B.OK: number of words successfully answered for the particular question on the second screening test.
- B.All: the number of all words tested under the question on the second screening test.

3.7. Web Application – Admin Panel

The iLearnRW system includes also the Admin-Panel web-app where the experts can setup user accounts, see student reports and setup screening tests. This web page is currently available at <https://ssl.ilearnrw.eu/ilearnrw/apps/panel>. The application is fully described in D6.2 and only minor updates happened through the last year.

The Admin panel is a server side application intended to allow different types of users to have access to the iLearnRW system's data in an easy, accessible way. Through the Admin panel, students and teachers can be added to the system, student profiles can be analysed, usage reports can be viewed and data from screening tests can be entered. Therefore the Admin panel application is a Performance Monitoring and User Management tool.

Each user of the iLearnRW system can have one of the following roles:

- Administrator: this user is allowed to do everything in the Admin panel: he can create other types of users, create roles, manage permissions, etc. There is currently only one user defined with the Administrator role.
- Expert: this user is allowed to create teachers and students. He can modify data for existing teachers and students. He is also permitted to create, modify and perform screening tests.
- Teacher: a teacher can view the profiles of the students assigned to him and perform screening tests.
- Student: students are currently not allowed to log into the Admin panel.

All users can view reports containing statistical data of the iLearnRW system's usage in a user friendly manner based on graphical charts (see Figure 33 and Figure 34), but also more technical data from the Data Logger.

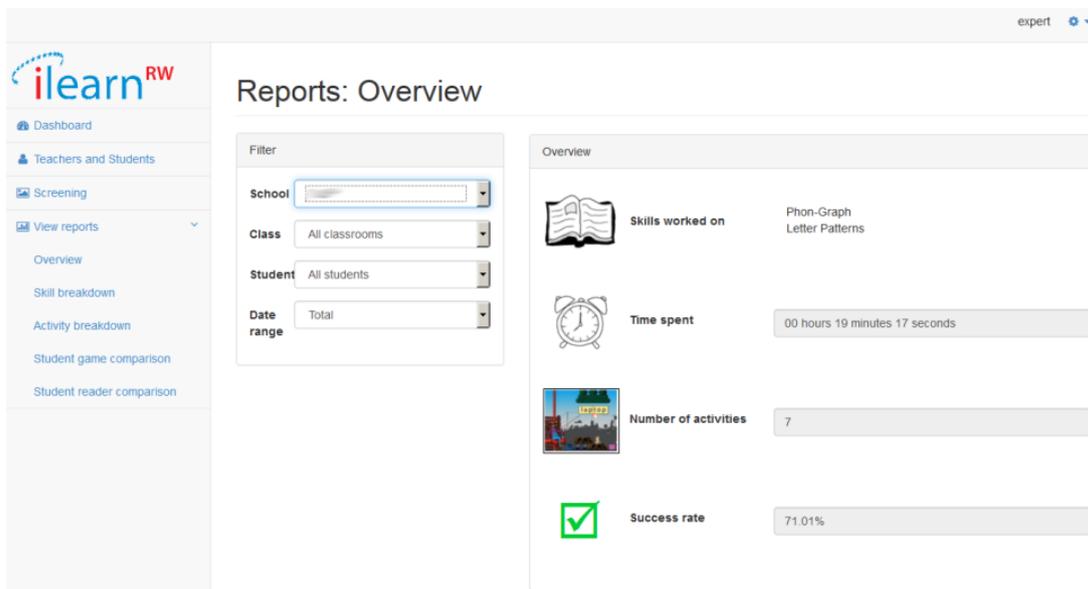


Figure 33. Admin-App, School Overview

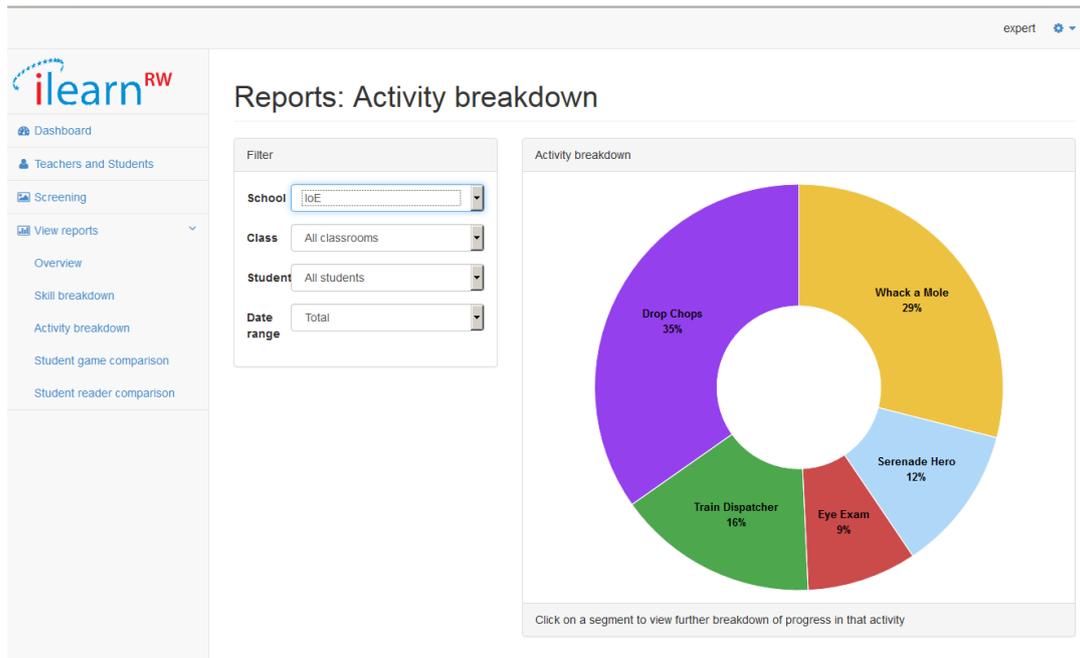


Figure 34. Admin-App, Activities Breakdown

Screening Test

The Admin-Panel contains also a tool that the users can use to setup tests and initialize user profiles. The Screening Test tool consists of two parts:

1. The tests administration panel
An expert is responsible to create and delete tests as well as insert and delete questions on them. Experts are also responsible to pick one of the tests they have created to be the *default test*, that is, the linguistic test that will be used from the teachers to test their students.
2. The student's answers form
Tests are printed on paper and are given to the students. After a student has finished with a test, an expert or an authorized teacher should enter the child's answers into our system. Then the system initializes the user profile based on the answers.

An advantage of using the screening tool generator as an interface for creating linguistic tests is that all the data of each test (questions and answers) follow a common format. Consequently, all data collected from the screening tests are stored in the central iLearnRW database and we are able to analyze this data.

4. iLearnRW Dictionaries

One of the core components of the iLearnRW system is the set of the special annotated dictionaries. The dictionaries were developed based on the specific needs of dyslexic readers and teachers providing support to them in the analysis of words into their phonic components. This includes division into syllables both in spoken and written form, mapping of graphemes to phonemes and highlighting of common spelling patterns. To be able to provide the appropriate guidance we designed an annotated dictionary, list of phonic features with associated look up routines and dictionary annotation algorithms. The resulting phonics engine was then used by the reader application and the serious games.

The iLearnRW project developed dictionaries and related tools for two languages English and Greek. However, the vast majority of the work done was on English due to its peculiar level of complexity in the area of phonics and orthography. Therefore, unless specified, the issues discussed pertain to English. Greek is mentioned only as relevant.

The main aims of this phonics engine for the iLearnRW project were:

- provide automated guidance to students and teachers reading texts (using highlighting as well as explicit information)
- generate more extensive word lists for practice activities
- provide information about word structure to the game engine

After investigating different options, including using a phonological component from an English text-to-speech engine, it was decided to build an English wordlist. We started with a list of 5,000 most frequent words in English from word frequency data provided by the Corpus of Contemporary American English and used the open source tool hunspell to create their possible inflectional and derivational forms. Then we used an online tool to generate their pronunciation. This resulted in over 15,000 lexical forms for most of which we had pronunciations and syllabification information.

From then on we had to develop algorithms for:

1. Identifying phoneme-grapheme mappings
2. Identifying orthographic syllabification
3. Labelling suffix and prefix types
4. Adding number of letters, phonemes and syllables

The final form of the dictionary required significant work in manual checking, editing and refining the annotation algorithms.

Building a dictionary of Greek was much easier. Since the pronunciation of Greek is mostly regular, it was not necessary to come up with grapheme phoneme correspondences and algorithms exist for syllabification of Greek. As a result, the Greek dictionary could contain many more lexemes (500.000) than English. In addition, Greek contained labels of parts of speech which had been identified as important by Greek literacy intervention experts. Parts of speech were omitted in the English dictionary due to the level of difficulty of identifying them automatically in a free text. Also, since the same form can belong to multiple parts of speech, this would complicate the structure of the dictionary unduly. Solving this issue will be the task of future work.

Console Application (Word List Builder)

The serious game should be able to update its content depending on the linguistic area and the level of the user. For that reason we had to create special word lists containing words grouped per category and sorted by difficulty.

The solution to the above is to work in the two stages; first we automatically picked words from the dictionary and then the experts manually checked the above lists for refinement, if needed

For the first step we created a console application (the word list builder), an application that reads the dictionary (English or Greek) and works as follows:

1. Put each word of the dictionary to the linguistic category that it matches

2. For each word list sort words based on the structure (consider the number of syllables, number of letters, word frequency sequence of consonants and vowels etc.)
3. Create csv files (ready to be used in MS Excel) showing all lists so that the experts can check the lists easily

The above work produced one word list per profile entry; an example of a list follows:

- | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">• add• odd• odds• middle• address• wedding• buddy• added• sudden• ladder• suddenly• addition• kidded• etc... |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Table 1. List of words for problem “dd=d (add) ⇔ Pronounce dd as d. For example: add, kidded”

5. Technical Details

In this section we explain the basic architecture of the iLearnRW system and we also outline the features of each component.

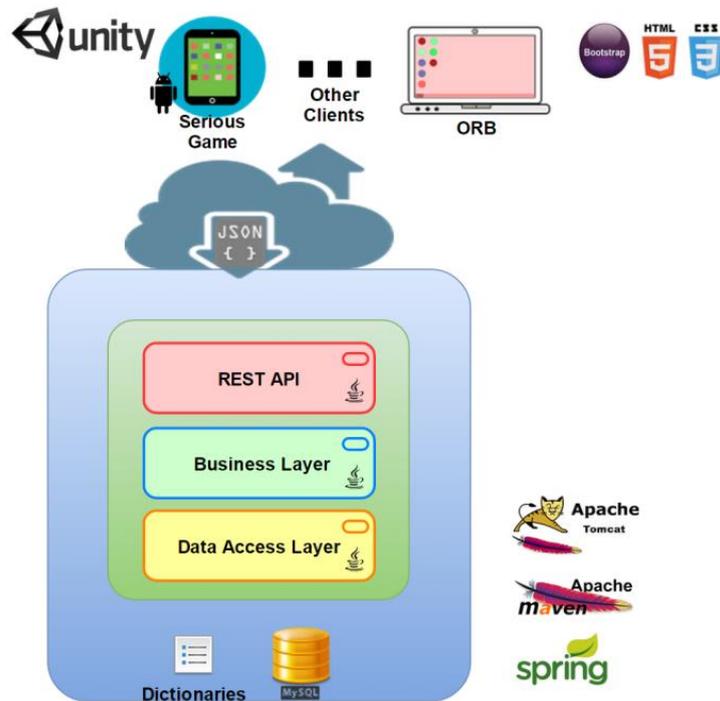


Figure 35. iLearnRW Technical Details

5.1. System Components

The iLearnRW is based on the classic server-client architecture; Figure 35 shows the main parts of the architecture. Our system uses a set of special dictionaries (English and Greek) as well as a database containing user profiles. Firstly we have a data access layer (DAO) for the communication with the database. Then the business logic uses the DAOs and the communication with the outer world is achieved through a RESTfull service. Currently, there are four applications using the services (Serious Game, Reader, ORB and the Text Analysis Tool). There is also one application that is served directly from the iLearnRW server (the Web App). A description of the system components follows in the next two subsections. First we explain the server side components and the client applications follow.

5.1.1. Server Side

The iLearnRW server contains the central parts of the system that can be used by the client apps. The main modules hosted by the server are: a) the *dictionaries*, b) the *user profiles*, c) the *data logger mechanism* and d) the *web-application*. The connection of all client apps to the server is achieved by the usage of a set of special web-services exposed by the server. A brief description of the above parts follows in the next five paragraphs.

Dictionaries: Text files containing words grouped by linguistic category/difficulty as well as a full dictionary of the language (see section 4 for a detailed description). The full dictionary contains a list of words with extra attributes like syllabification, phonetics, frequency etc. The last information is being used along with special routines to analyse or annotate texts.

Database with User Profiles: In a MySQL database we store the user severities against each linguistic difficulty. This information is essentially a snapshot of the user skills/weaknesses at the current time.

Data logger Mechanism: This is a special module that allows each external application (client) to store user logs to a MySQL database. The information that is being stored by the data logger can be considered as an extension of the user profile since it contains the user history.

Web Application: This is the admin interface for the linguistic experts. An expert can create teacher and student accounts and preview the history of her/his students through tables or graphs.

Web-Services: All client apps need a way to connect to the central server and use its features. The last task is achieved through a set of RESTfull web-services exposed by the server. Next we list the main iLearnRW services exposed under the base URL <https://ssl.ilearnrw.eu/ilearnrw/>:

- User authentication path: *user/auth?username=<username>&pass=<user_password>* (get request): this service returns a pair of tokens that can be used in the next services allowing the user to have access to restricted methods.
- User profile path: *profile?userId=<user_id>&token=<token>* (get request): returns the user profile.
- Default profiles path: *public/profile?lc=<GR or EN depending on the language>* (get request): returns a default user profile for the selected language.
- Profile updates path:
logs<username>?applicationId=PROFILE_UPDATER&tags=PROFILE_UPDATE (get request): returns the profile updates of the user defined by the selected username.
- Words for activity path:
activity/new_data?count=<number_of_words>&probId=<category_index>&activity=<activity_id>&userId=<user_id>&token=<token> (get request): returns a set of words special annotated for the specified activity.
- Text analysis path: *text/classify?userId=<user_id>&token=<token>* (post request): the post request requires the text to be analysed in the body of the request. It returns information about the text like number of words and other text metrics as well as lists of words matched to each problem difficulty. Furthermore, the response contains user related metrics like *iLearnRW text score* and others based on the text analysis against the user profile.
- Generic Text analysis path: *public/text/classify?lc=<GR or EN depending on the language>* (post request): works like the text analysis service but it uses a generic profile.
- Text annotation path: *text/annotate?userId=<user_id>&lc=<language>&token=<token>* (post request): the post request requires the text to be annotated in the body of the request. It returns an html with the initial text but contains all words inside special tags with unique ids. It also returns a list of words along with problems matched (and positions of the matched linguistic phenomena) with references (by the id) to the html words.
- Generic Text annotation path: *public/text/annotate?lc=<GR or EN depending on the language>* (post request): works like the text annotation service but it uses a generic profile.

A detailed list of all web-services exposed can also be found at: <https://ssl.ilearnrw.eu/ilearnrw/>

5.1.2. Client Applications

By the end of the 3rd year of the project there are four client applications available for experts, teachers and students. Next we describe each of these applications.

On Line Resource Bank: The ORB is a web page allowing access to the main features of the core iLearnRW system. The user can check student profiles as well as analyse and annotate texts. Although this is a web page, this app is also a client of the main server since it consumes the web-services exposed by the central iLearnRW server.

Serious game: The serious game contains nine activities which mainly display annotated content asking the user to find special characteristics of words (e.g. suffixes, syllables etc). All users' answers are logged using the data logger component. The last operation produces a detailed history of each user.

Reader: This is an android application focused on presenting annotated texts to the user. The student can create a library of texts and setup a set of annotation rules. Then, the texts will be displayed annotated

based on the enabled rules. The reader offers also a text-to-speech functionality making the reading easy even for students struggling with it.

Text Analysis Tool: This is a java application that allows the user to create a library with texts and analyse/annotate them based on a user's profile. It works on Windows, Linux or Mac but it requires internet connection since it uses the web-services of the central system.

5.1.3. Scalability of the iLearnRW eco-system

We have already explained the basic architecture of the iLearnRW system as well as the communication of the system with client apps. Note that the system offers all the basic functionality via special web-services. The last means that anyone can build her/his application and connect it to the iLearnRW system. Bellow we give the basic steps a developer should follow in order to connect her/his application to the iLearnRW system.

```
private static String baseUrl = "https://ssl.ilearnrw.eu/ilearnrw/";
private final static String USER_AGENT = "Mozilla/5.0";
private static String username = "joe_t", password = "test";
private static int userId = 68;

public static void main(String[] args) {
    try {
        // PART A: AUTHENTICATE USER
        // A.1: authenticate user
        String tokens = sendGet("user/auth?username=" + username +
"&pass=" + password);
        // A.2: get auth token to use it
        String auth = getAuthToken(tokens);

        // PART B: PLAY A 'GAME'
        // B.1: get word list for some problem (e.g. 1st category, 1st
// problem)
        int category = 0, index = 0; // java counting ;-)
        String jsonList = sendGet("wordlists/problem_data?probId="
+ category + "_" + index + "&userId=" + userId + "&token=" + auth);
        // do something with the words
        System.out.println(jsonList);
        // B.2: send some logs to the server
        String jsonLog = new Scanner(new File("logEntry.json"), "UTF-8")
            .useDelimiter("\\A").next();
        String response = sendPost("logs?token=" + auth, jsonLog);
        // B.3: update profile
        response = sendPost("profile/update?userId=" + userId
+ "&category=" + category + "&index=" + index + "&token=" + auth, "");
        // print the updates
        System.out.println(response);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figure 36. Java Code Snippet Using the iLearnRW Services

Bellow we explain the extra methods called by the java code:

- sendGet(String url): sends a get request to the specified URL

- `sendPost(String url, String data)`: sends a post request to the specified URL using the *data* variable as the body of the request
- `getAuthToken(String jsonData)`: takes as input a String in JSON format of type `{“auth”:”auth_token_goes_here”, “refresh”:”refresh_token_goes_here”}` and exports the auth token as a String.

The example shown in Figure 36 essentially does the following:

1. Retrieves tokens for the user
2. Asks and gets words for the “Pronounce t as t. For example: to, it, at” problem (1st category, 1st index problem of the English language)
3. Sends a log back to the data-logger
4. Asks the central system for profile updates

Note also that the log should be in the following JSON format, describing the details of the game event and the user:

```
{
  "username":"joe_t",
  "applicationId":"YourAppID",
  "tag":"WORD_SUCCESS",
  "problem_category":0,
  "problem_index":0,
  "level":"some level",
  "mode":"normal",
  "value":"1"
}
```

The above example shows that the iLearnRW system allows anyone to build a ‘dyslexia-friendly’ application. The developer should only be concerned about the aesthetics and the mechanics of her/his application and let the iLearnRW system decide about the user profiling and the appropriate content (words for the games).

5.2. iLearnRW Server - Hardware Specifications

To support our server-client architecture we use a server accessible via HTTPS under the domain <https://ssl.ilearnrw.eu/>. The hardware and software specifications of the server are as follows:

- Hardware
 - Processor: Intel(R) Pentium(R) CPU G3220 @ 3.00GHz, 64 bit architecture, dual core
 - RAM: 11 GB and 18GB of swap area
 - Hard drive: 450 GB
- Software
 - Operating System: Debian GNU/Linux 7, Kernel 3.2.0-4-amd64
 - Server version: Apache Tomcat/7.0.28 (amd64)
 - JVM Version: 1.6.0_32-b32 (Vendor: Sun Microsystems Inc.)
 - Database: MySQL

6. Project Source Codes and Licensing

The iLearnRW system contains several source code projects and resources. We decided that we license everything under an Open Source license and all content under Creative Commons. We believe that our choice gives to the software the greatest chance of use.

Software - BSD license: This is completely unencumbered license that allow full commercial exploitation without any limits by any one (other than giving attribution). This makes the code created most useful for all and most likely to be implemented.

Content - Creative Commons license: Any content generated (text, dictionary, graphics, etc.) is licensed under Creative Commons. This ensures that all partners have full use of the content without worries about the limits of the license.

6.1. Open Source Projects

After our decision regarding the licensing we published all GitHub repositories. The iLearnRW project consists of 5 different modules. The list of these modules and the description for each of them follows:

1. iLearnRW Library: A Java project containing the core functionality of the iLearnRW project. Classes to support user profile, language categories/problems, dictionaries etc. are implemented there. Also, this project contains the classes responsible for text classification and text annotation.
GitHub URL: <https://github.com/mpakarlsson/ilearnrw>
2. iLearnRW Admin Panel (web): A Java Web-Project using the Spring MVC framework along with Maven. This project contains all the web-services exported by the iLearnRW and the web application.
GitHub URL: <https://github.com/mpakarlsson/ilearnrw-service>
3. iLearnRW On Line Resource Bank: A Java Web-Project using the Spring MVC framework along with Gradle. This project contains the web pages forming the ORB, it makes use of the services exposed by the web app.
GitHub URL: <https://github.com/mpakarlsson/ilearnrw-orb>
4. iLearnRW Reader: An Android Project which contains all the screens and the functionality of the reader app.
GitHub URL: <https://github.com/mpakarlsson/ilearnrw-reader>
5. iLearnRW Serious Game: A 3D-Unity project with all the source code and the assets of the serious game.
GitHub URL: <https://github.com/TAPeri/WordsMatter>

7. Installation Instructions

Both the iLearnRW Serious Game and the Reader have been packaged as Android applications. The installation of the apps is straightforward and is described below:

Word Matters Game Installation

1. Start the Android device
2. Open your browser app
3. Go to the page <https://github.com/TAPeri/WordsMatter/releases/tag/2.5>
4. Click on the item “WordsMatter.apk” under the “Downloads” section to download the installer
5. When the download is complete android will notify this in the notification area
6. Press the “Download Complete” notification
7. Android will now ask you to install the app, press “Install/Yes”
8. The app is now installed and ready to be used

Word Matters Reader Installation

1. Start the Android device
2. Open your browser app
3. Go to the page <https://github.com/mpakarlsson/ilearnrw-reader/releases/tag/V2.0>
4. Click on the item “ilearnrw-reader.apk” under the “Downloads” section to download the installer
5. When the download is complete android will notify this in the notification area
6. Press the “Download Complete” notification
7. Android will now ask you to install the app, press “Install/Yes” (if not, use an external application like *apk-installer* to locate and install the app)
8. The app is now installed and ready to be used

Accounts To User the Applications

Note that both applications require full connection to the internet throughout the usage (gaming or reading). Also, only, registered users can access the content of the application. In case you don't have valid credentials you can use the following generic accounts:

English Accounts:

1. Generic Student
 - Username: SmartEnglish
 - Password: test
2. Generic Teacher
 - Username: englishteacher
 - Password: test

Greek Accounts:

1. Generic Student
 - username: SmartGreek
 - Password: test
2. Generic Teacher
 - Username: greekteacher
 - Password: test

The above accounts can also be used in order to get access to the ORB or to the web page of the iLearnRW project.