

Oleggo

DESIGN DOCUMENT



Falci Angelo
Gallego García Julián David

Politecnico di Milano January 26, 2018





CONTENTS

1.	Chapter 1 - Introduction.....	- 5 -
1.1.	Purpose	- 5 -
1.1.1.	Intended audience	- 5 -
1.2.	Scope	- 5 -
1.3.	Definitions, acronyms, abbreviations Reference documents	- 6 -
1.3.1.	Definitions	- 6 -
1.3.2.	Acronyms	- 6 -
1.3.3.	Abbreviations	- 6 -
1.4.	Goals.....	- 7 -
1.5.	Framework for app development	- 7 -
1.6.	Overview	- 8 -
2.	Chapter 2 - Architectural Design.....	- 9 -
2.1.	Overview	- 9 -
2.2.	Architectural Style and Patterns.....	- 9 -
2.3.	Use case functional requirements analysis.....	- 11 -
2.3.1.	Insert new book	- 11 -
2.3.2.	Update bookmark	- 12 -
2.3.3.	Search a specific word in dictionary.....	- 12 -
2.3.4.	Insert a note	- 13 -
2.3.5.	Search word definition.....	- 14 -
2.3.6.	Change state of book	- 14 -
2.3.7.	See all favorite notes	- 15 -
2.3.8.	See General dictionary.....	- 15 -
2.3.9.	See the notes of specific book.....	- 15 -
2.3.10.	See the words definitions of a specific book.....	- 16 -
2.3.11.	Remove a word	- 16 -
2.3.12.	Remove a note	- 16 -
2.3.13.	Remove a note from favorite notes	- 17 -
2.3.14.	Add a note to favorite notes.....	- 17 -
2.3.15.	Modify an existing note	- 17 -
2.4.	Sequence diagrams.....	- 18 -
2.4.1.	Add new books.....	- 18 -
2.4.2.	Update bookmark	- 19 -
2.4.3.	Add new note	- 20 -
2.4.4.	Search word definition.....	- 21 -
2.4.5.	Change the state of a book	- 22 -
3.	Chapter 3 - Class diagram.....	- 23 -
3.1.	Database	- 23 -
3.2.	Pages	- 23 -
4.	Chapter 4 - User Interface Design.....	- 26 -
4.1.	Home Page	- 26 -
4.1.1.	Library	- 27 -
4.1.2.	Favorite Notes	- 27 -
4.2.	Book	- 28 -



4.2.1.	<i>Information</i>	- 28 -
4.2.2.	<i>Notes</i>	- 30 -
4.2.3.	Dictionary	- 31 -
4.2.4.	Update progress screen.....	- 32 -
4.3.	Notes	- 33 -
4.4.	New Book	- 35 -
4.5.	Side Drawer menu	- 36 -
4.6.	Complementary screens.....	- 37 -
5.	Software System Attributes	- 38 -
5.1.	Reliability.....	- 38 -
5.2.	Availability.....	- 38 -
5.3.	Security	- 38 -
5.4.	Maintainability	- 38 -
5.5.	Usability.....	- 38 -
5.6.	Portability.....	- 38 -
6.	References	- 39 -



1. Chapter 1 - Introduction

1.1. Purpose

This document is intended to help understand and evaluate design and prototyping steps taken into account for the development of the mobile application call “Oleggo” (named from olè exaltation of being good at something and leggo from the verb leggere Italian word for read), it explains both the application domain and the system for further understanding. It explains the features of Oleggo, the main components we used and the interface we implemented for the user.

This project is implemented as an outcome of the course “Design and Implementation of mobile applications”.

1.1.1. Intended audience

This document is realized for all people participating in and/or supervising the project:

- Expected audience of this document is the developers and anyone who intends to develop on this program
- Developers who can review project's capabilities and more easily understand where their efforts should be targeted to improve or add more features to it (it sets the guidelines for future development).
- Project testers can use this document as a base for their testing strategy as some bugs are easier to find using this document. This way testing becomes more methodically organized.
- End users of this application who wish to read about what this application can do.

1.2. Scope

The goal of Oleggo is to create an application that can be used by students to take notes or look for a word in a fast way to reduce the dead times. An other advantage is that the students can have always with them their notes and a personal dictionary with the words that were more complicated for them. They also can save like favorite the most important note in order to find them in the home page.

But it can also be used by anyone that want to save in own phone all books he/she is reading or read in the past and, for each book, save the notes inserted or word searched and also



the bookmark. To take notes with Oleggo will be fast mostly because it is possible insert note or word using a speech recognized alternatively to write it.

1.3. Definitions, acronyms, abbreviations Reference documents

1.3.1. Definitions

Note: Something that write the user, it can be its personal notes or a quotes from a book.

Architecture: <system> fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution.

Bar-code: It is an identification code that include the ISBN and usually is on the back f the book.

Drawer menu: It is the menu that can appear and disappear on the left or right in an application.

1.3.2. Acronyms

UML: Unified Modelling Language.

ISBN: International Standard Book Number

MVVM: Model View View Model

1.3.3. Abbreviations

App: Application.

DB: Database



1.4. Goals

The system has to provide these features:

- [G1] User can insert a new book inserting the ISBN by writing it or reading it using the bar-code reader in the app.
- [G2] For each book inserted, user can insert and modify a bookmark.
- [G3] For each book inserted user can insert a note by writing it or using speech recognize, and he/she can also add the page of the book. The note will be added to the “main active” book (explained further) .
- [G4] For each book inserted user can search the meaning of a word and add it in her/his personal dictionary. The word will be added to the “main active” book so user have a global dictionary and many little dictionary for each book.
- [G5] User can change the state of a book in “passive”, “active” (book user is reading and can change bookmark), “main active” (user can have only one main active book, every word searching or notes added will be connected to this book).
- [G6] User can choose to see all favorite notes or all notes connected to a certain book.
- [G7] User can choose to see all words in the dictionary or all words connected to a certain book.

1.5. Framework for app development

To develops Oleggo we used the framework Nativescript.

This framework can be used for cross-platform development, so our app could be run in Android or IOS system (at the moment the changes necessary to IOS interface hasn't been done, this will be added to future work). It is possible code for Nativescript in JavaScript or in TypeScript, for Oleggo we chose the first. Other languages used for the graphics are XML and CSS.



Why NativeScript?

There are a lot of motivations.

The main reason is that it offers many powerfully plugins to implement things, like speech recognition, that in other framework are hard to do (we will show all the plugins used).

An other reason is for the graphics. NativeScript offers a really easy implementation of the components for developing the user interface also due to the fact that we have previously worked with Javascript and XML for develop of websites allows an easy way to implement the graphics for the application.

Another motive is due to the fact that Nativescript allows you to upload the changes live on the emulator or cellphone where the app is running shorts the time of developing in a consider way.

Finally, they are libraries and templates for mobile application developed by the community that allows us to implement complex interfaces that without would take much more time. We combine some of them to implement beautiful graphics components that, without these templates, are very hard to do so beautiful (for example the side drawer menu).

1.6. Overview

This document is divided in four parts with clean and non-ambiguity description of the whole system:

- **Chapter 1:** brief description of the idea, goals and objective of the document.
- **Chapter 2:** Explanation of the main functionalities with constraints, assumptions and hardware dependencies.
- **Chapter 3:** Specifications of the system functionalities and non-system functionalities possible scenarios.
- **Chapter 4:** Use cases and UML models.
- **Chapter 5:** Software system attributes.



2. Chapter 2 - Architectural Design

2.1. Overview

In this section it is provided detailed information of the product structure and the implementation of the general structure. In the following sections it is provided details of why it was decided the architecture for the system, what are the characteristics of the architecture that make it the chosen one.

In the use case section, it is provided the details about how user can interact with Oleggo and what happen (including exception) when user used the different services offer by the application.

Then it continues talking about the general components of the architecture and the subcomponents of each component and their interactions. Following by information of all the functions that do each subcomponent.

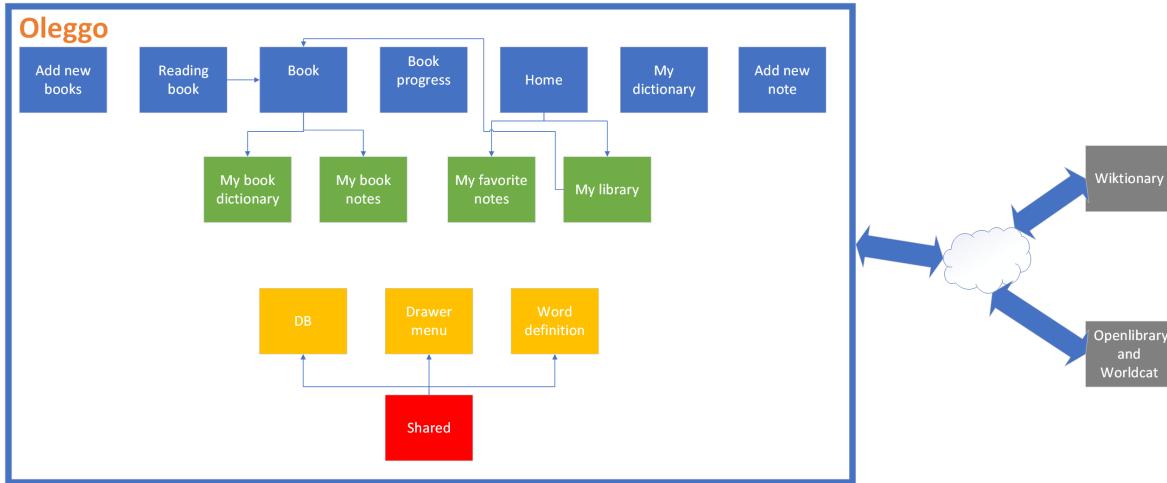
After the primary information on how it is constituted the architecture, its explained how all these components work to make the system to work effetely.

The chapter finish with an explanation about the interface and how some components interact with it.

2.2. Architectural Style and Patterns

For NativeScript is used to use the MVVM pattern (Model View View Model) where:

- *Model*: The model defines and represents the data. Separating the model from the various views that might use it allows for code reuse.
- *View*: The view represents the UI, which in NativeScript is written in XML. The view is often data-bound to the view model so that changes made to the view model in JavaScript instantly trigger visual changes to UI components.
- *View Model*: The view model contains the application logic (often including the model), and exposes the data to the view. NativeScript provides a module called 'Observable', which facilitates creating a view model object that can be bound to the view.



In Oleggo we use MVVM but with some modifications. We preferred to have a folder for each page of our app and sub folders if a page included more pages. We choose this because each page use data in different ways and in order to have a more organization and scalability because someone can work only to a page without modify the other. It's mean that each view has only one model and each model has only one view but in order to don't have a different class in which database is called we create shared components that include all possible queries need in the pages and every page can use it when want to interact with the database. In this way if we want modify something, like the name or type of the columns, in the database we must work only in this class.

We will see all blue and green components in the chapter "User Interface". The shared components aren't visible for the user, they are only used by the pages above:

DB → contain all the query to the database, in this way if we modify something to database we must modify only this class.

Drawer Menu → it is the hide menu that appear in every page if user click button on the left-top or shift the finger from left to right.

Word Definition → it's include a set of pages that manage the searching of words in Wiktionary.



2.3. Use case functional requirements analysis

This part of the document shows how users interact with Oleggo in order to reach the different goals. We will see also the flow of actions the user needs to do but we not see the system internals.

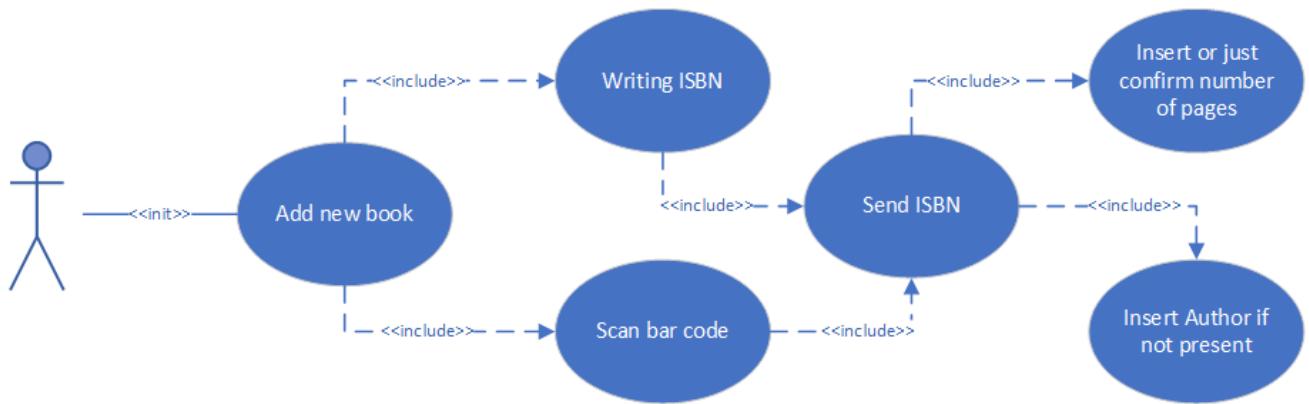
We can identify only one group of actors involved in this application:

- Users – who had downloaded and installed the application Oleggo

For the most important, and complex, functions there is the graphics representation of user case.

2.3.1. Insert new book

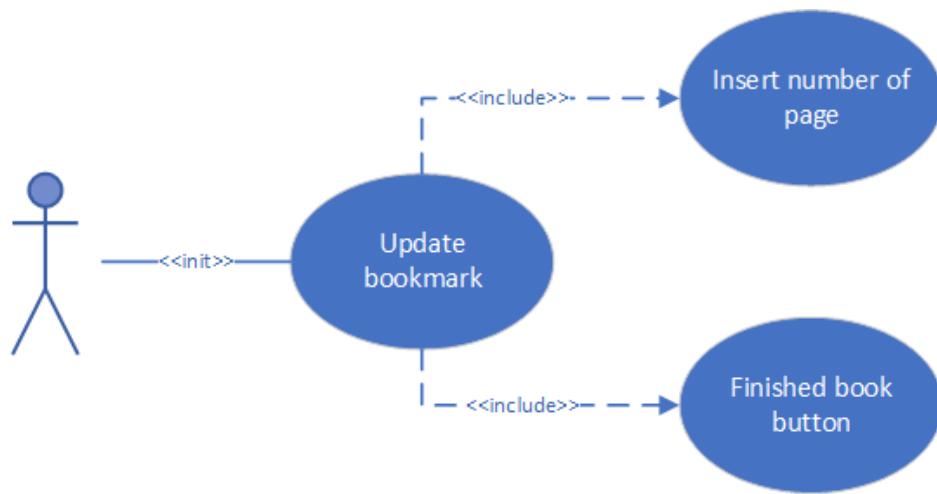
Actor	Users
Goal	Insert a new book using ISBN
Event flow	Press New Book voice in the drawer menu Insert the ISBN of book writing it or scanning the bar code Press Send ISBN If book is not present in online library insert the title and continue If author is not present in online library insert it and continue Confirm (or insert if it isn't present) the number of page in the book and continue
Output conditions	Book is inserted
Exception	User insert a not valid number of pages (for example insert letters instead of numbers) Insert a not valid ISBN





2.3.2. Update bookmark

Actor	Users
Goal	Update the bookmark of a book
Event flow	Press Home or Reading Books in the drawer menu Press in the book we want to modify bookmark Press the circle button at left side Insert a number and press update progress or press "I have finished button"
Input conditions	Book already added
Output conditions	Bookmark updated Book state changed and become "read"
Exception	/



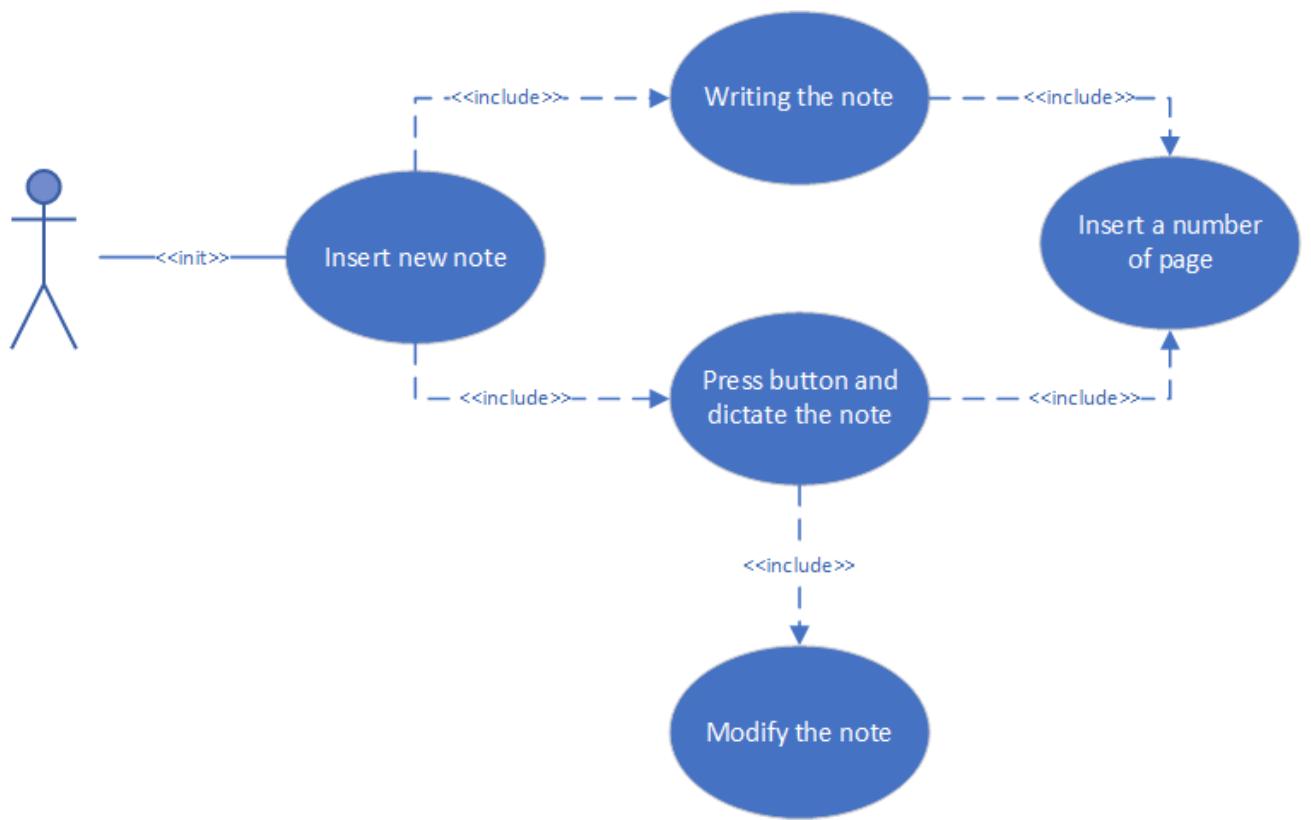
2.3.3. Search a specific word in dictionary

Actor	Users
Goal	Search a specific word in the global dictionary
Event flow	Go in global dictionary to see all the words (see table above for further details) Insert in Search box on top of the page the word you want to search the meaning or a part of this word
Input conditions	Words already searched
Output conditions	Show all the word that included what you wrote in the search box
Exception	/



2.3.4. Insert a note

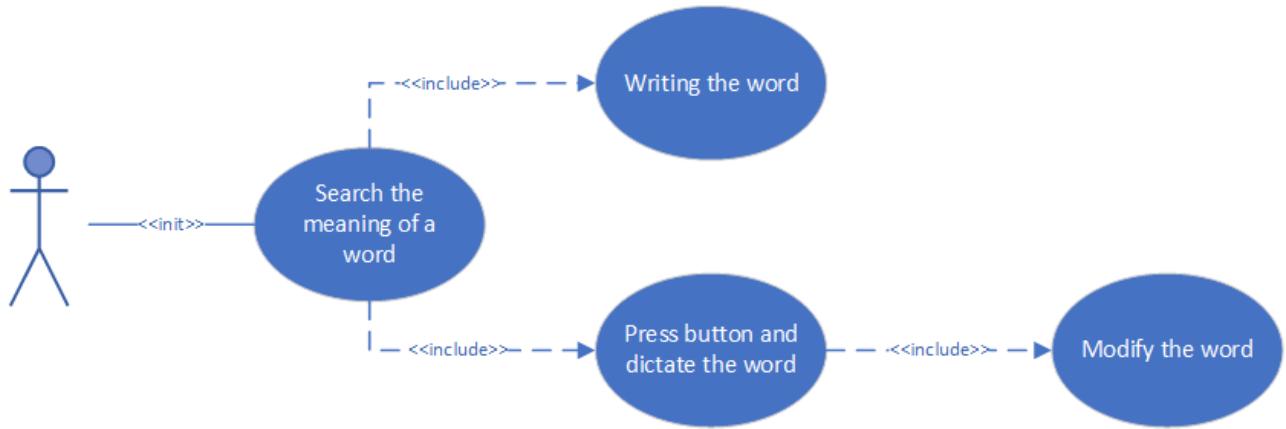
Actor	Users
Goal	Insert a note connected to “main active” book
Event flow	Press New Note in the drawer menu Insert the note writing it or using speech recognized (pressing “press and speak” and speak) and eventually modify it Optionally it possible insert a number of page Press add quote
Input conditions	One book is “main active”
Output conditions	Quote added and connected to the “main active” book (on default the quote isn’t favorite)
Exception	If there isn’t any “main active” book the quote isn’t added and an alert advice the user





2.3.5. Search word definition

Actor	Users
Goal	Search and save the meaning of a word connected to "main active" book
Event flow	Press New Note in the drawer menu Insert the word writing it or using speech recognized (pressing "press and speak" and speak) and eventually modify it Press look for the word
Input conditions	One book is "main active"
Output conditions	Word's meaning saved and connected to the "main active" book
Exception	If there isn't any "main active" book the word isn't added and an alert advice the user



2.3.6. Change state of book

Actor	Users
Goal	Change the state of a book (passive, active, main active or read)
Event flow	Press Home or Reading Books in the drawer menu Press in the book you want to change the state Press in the big, center button more times to change state in "active" or "passive" Press the little button on the left to set the "main active" state If you finish the book see the "update bookmark" table above
Input conditions	Book already added
Output conditions	The state of the book is changed
Exception	/



2.3.7. See all favorite notes

Actor	Users
Goal	See all favorite notes
Event flow	Press Home→My Favorite Notes in the drawer menu
Input conditions	There is at least one favorite note
Output conditions	Show all favorite notes
Exception	/

2.3.8. See General dictionary

Actor	Users
Goal	See all words searched and saved
Event flow	Press My Dictionary in drawer menu
Input conditions	There is at least one word
Output conditions	Show all words in the dictionary with their meaning in alphabet order
Exception	/

2.3.9. See the notes of specific book

Actor	Users
Goal	See all notes connected to a specific book
Event flow	Press Home or Reading Books in the drawer menu Press in the book you want to see the connected notes Press My Notes in the tab menu on top of the page or shift the finger from left to right
Input conditions	Book and notes for this book added
Output conditions	Show all notes connected to a specific book
Exception	/



2.3.10. See the words definitions of a specific book

Actor	Users
Goal	See all words, and their meaning, connected to a specific book
Event flow	Press Home or Reading Books in the drawer menu Press in the book you want to see the connected notes or shift the finger from right to left Press My Dictionary in the tab menu on top of the page
Input conditions	Book and word searched for this book added
Output conditions	Show all words and their meaning connected to a specific book
Exception	/

2.3.11. Remove a word

Actor	Users
Goal	Remove a word from dictionary
Event flow	Go in the page to see all words in dictionary or only the words connected to a specific book (see the tables above for details) Press remove button near the word we want remove
Input conditions	Word already searched
Output conditions	If you are in global dictionary you remove from dictionary all words equal the word you remove If you are in the dictionary connected to a specific book you remove only the word connected with this book (it means that if a word is connected to 2 books you remove only one occurrence of this and the word is present yet in the global dictionary)
Exception	/

2.3.12. Remove a note

Actor	Users
Goal	Remove a note
Event flow	Go in Home→My Favorite Notes or in the notes of a specific book (see the tables above for details) Press delete button near the note you want to remove
Input conditions	Note already inserted



Output conditions	The note is removed
--------------------------	---------------------

2.3.13. Remove a note from favorite notes

Actor	Users
Goal	Remove a note
Event flow	Go in the notes of a specific book (see the tables above for details) Press “not favorite” button near the note you want to remove from favorite notes
Input conditions	Word already searched and added to favorites
Output conditions	The note is removed from favorite notes and the star color change
Exception	/

2.3.14. Add a note to favorite notes

Actor	Users
Goal	Remove a note
Event flow	Go in Home→My Favorite Notes Press “favorite” button near the note you want to remove from favorite notes
Input conditions	Note already added
Output conditions	The note is removed from favorite notes and the star color change
Exception	/

2.3.15. Modify an existing note

Actor	Users
Goal	Modify an existing note
Event flow	Go in Home→My Favorite Notes or in the notes of a specific book (see the tables above for details) Modify what you want in the note you want to modify Press the button save near the chosen note
Input conditions	Note already added
Output conditions	The modification of this note is saved
Exception	/



2.4. Sequence diagrams

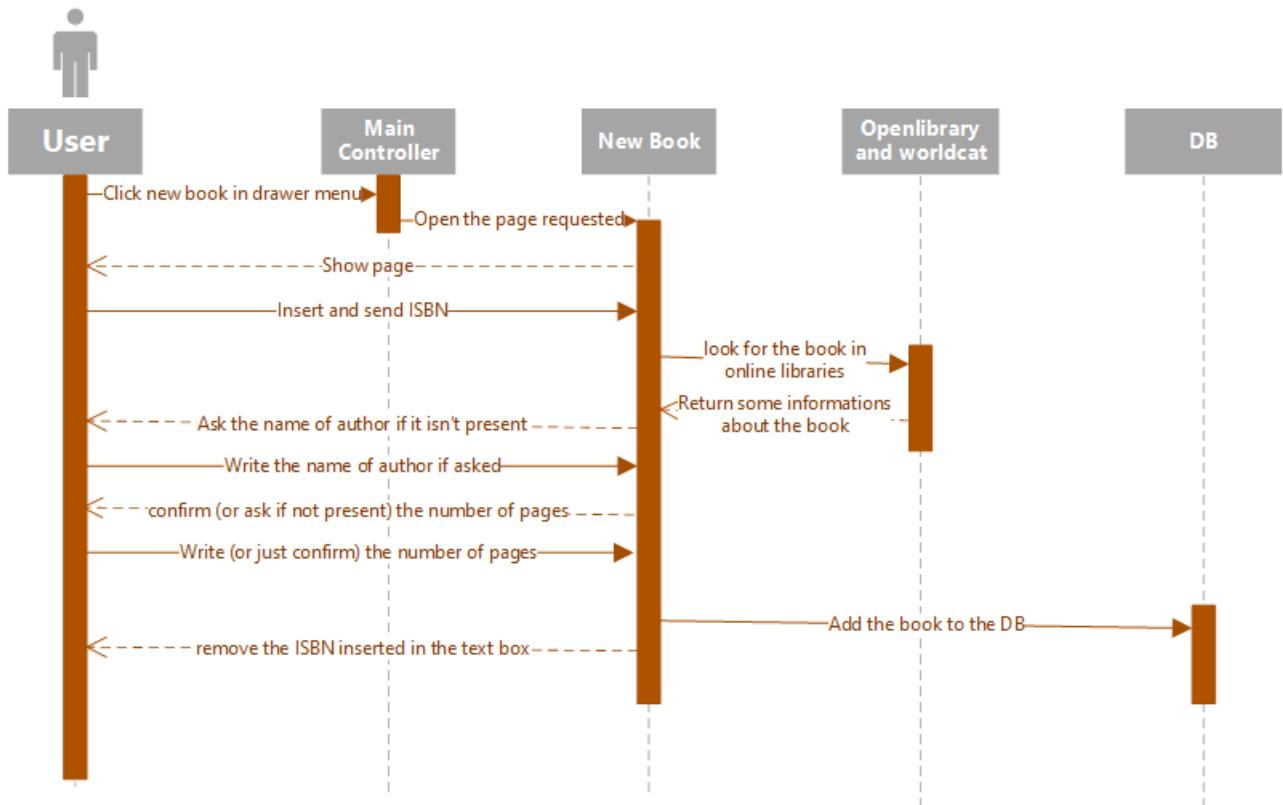
In this chapter we will model the flow of logic within our system in a visual manner that will enable us to document and validate our logic for further analysis and design purpose. We will present the most important interactions between the system and the user.

2.4.1. Add new books

User can add a book press “new book” voice in drawer menu. Then user can add ISBN writing it or scanning bar code.

Online libraries don't have all information about books, for this reason sometimes app ask to the user to insert the name of the author. It also asks to confirm, or insert if it isn't available, the number of pages that contain the book.

When the book is added an alert advice user and the text box with ISBN will be cleaned.



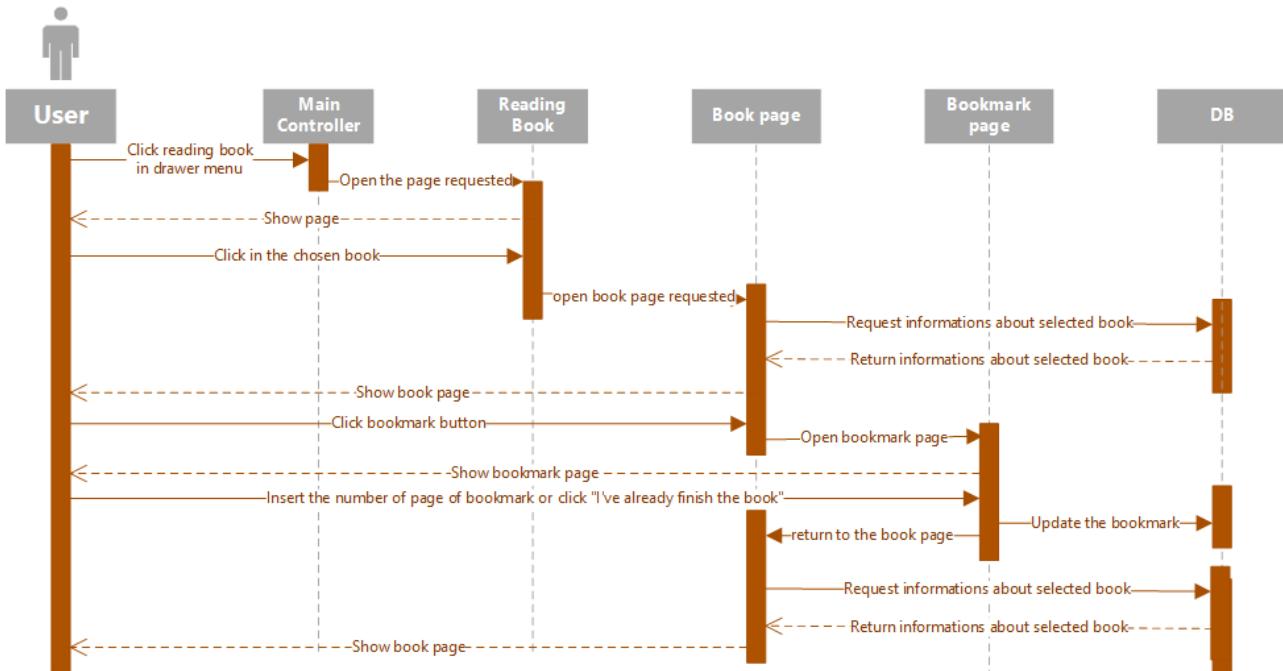


2.4.2. Update bookmark

In the personal page of book user can update the bookmark of a specific book.

User must click the little button at left and insert the number of page where put the bookmark.

Finally press “update progress” to confirm the new bookmark or “I have finished already the book”. If user insert a number greater than the number of pages that has the book the bookmark is set to the last page of book.





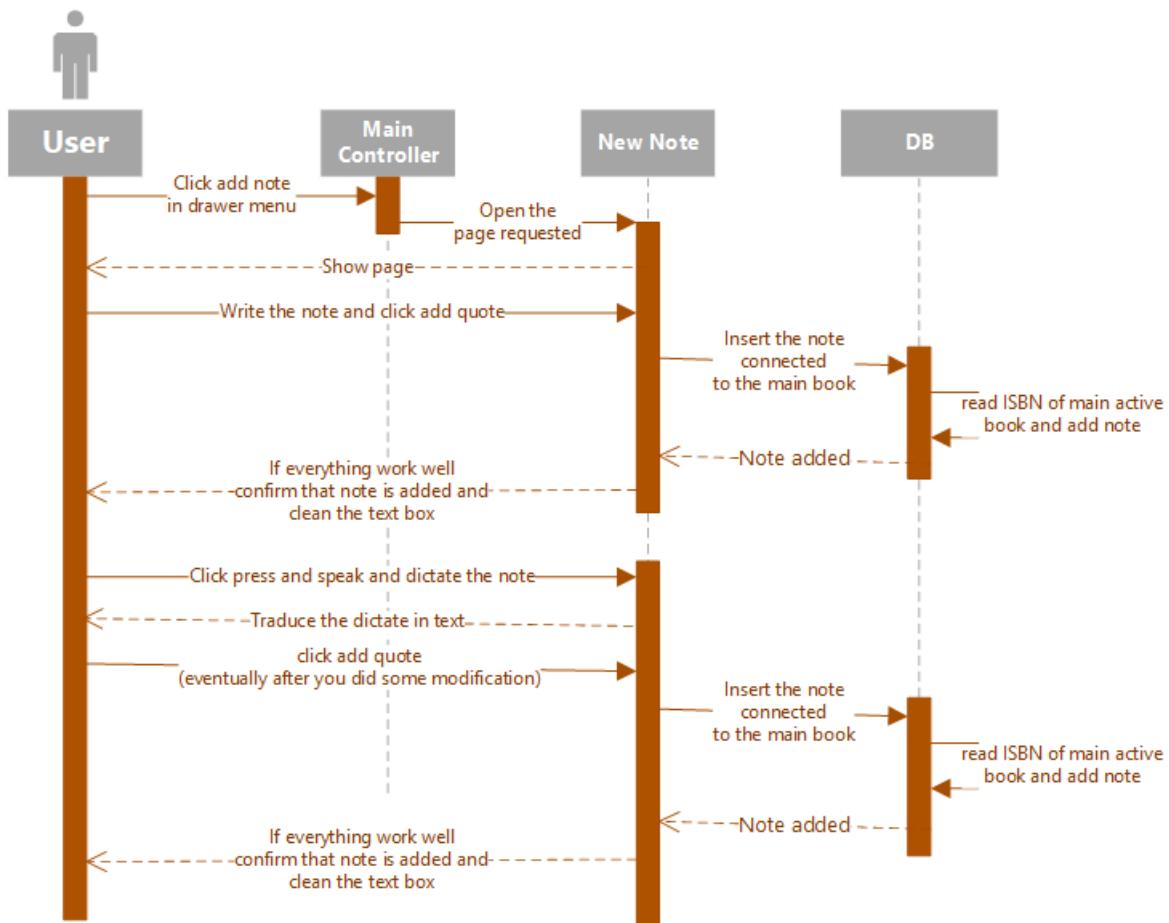
2.4.3. Add new note

User can add a note pressing “new note” voice in drawer menu.

It can write the note or dictate it using speech recognize. After dictate the note user can modify the text that appear after the translation from voice to text.

If there isn't a “main active” book note can't be added to database.

If the note is added successfully an alert advice the user and the text box will be cleaned.





2.4.4. Search word definition

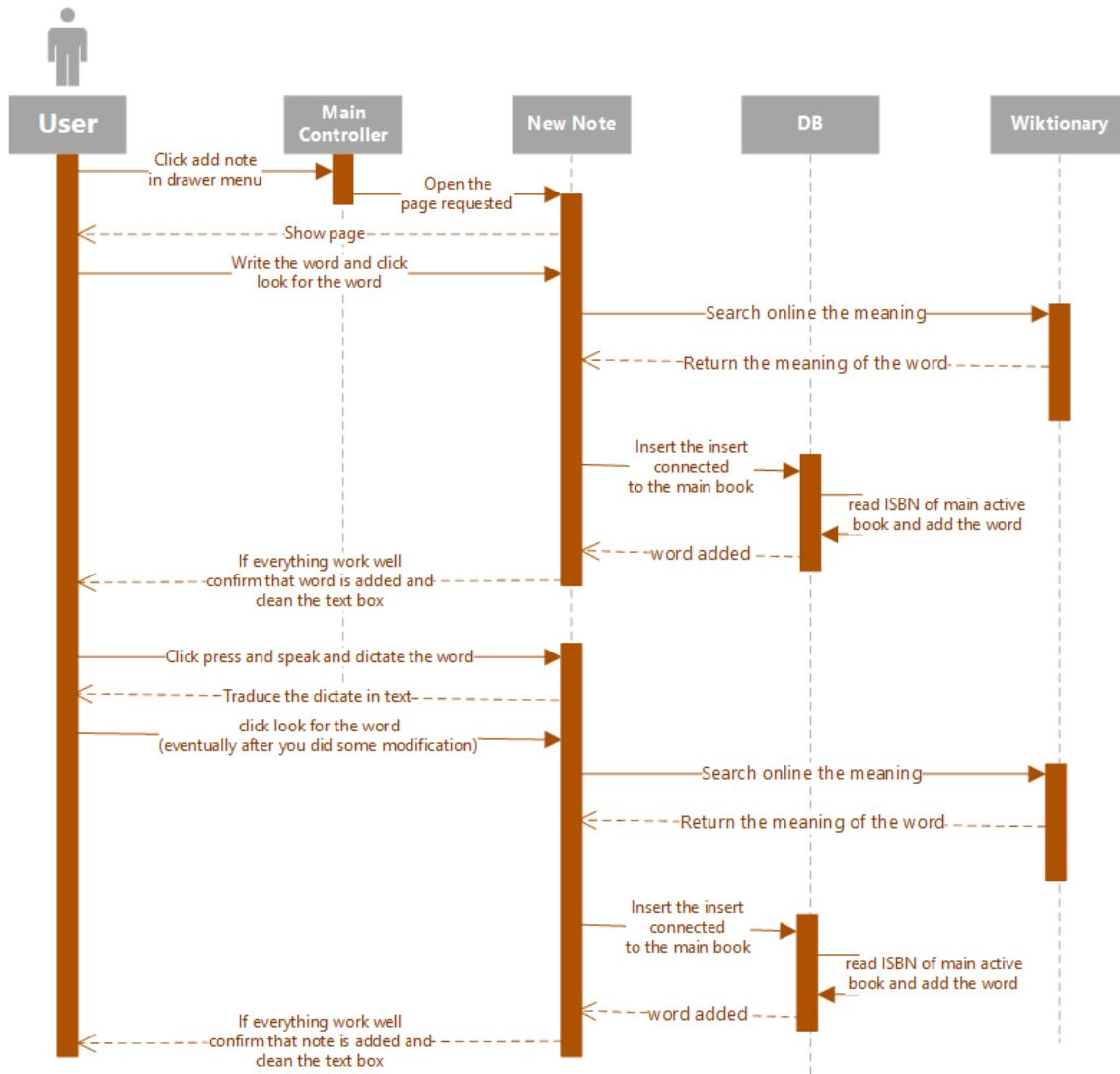
User can search a word pressing “new note” voice in drawer menu.

It can write the word or dictate it using speech recognize. After dictate the word user can modify the text that appear after the translation from voice to text.

If there isn't a “main active” book word can't be added to database.

If the word is added successfully an alert advice the user and the text box will be cleaned.

Finally, if user look for a “strange” word, or a word that doesn't exist in Wiktionary, the word is saved in dictionary without the meaning.



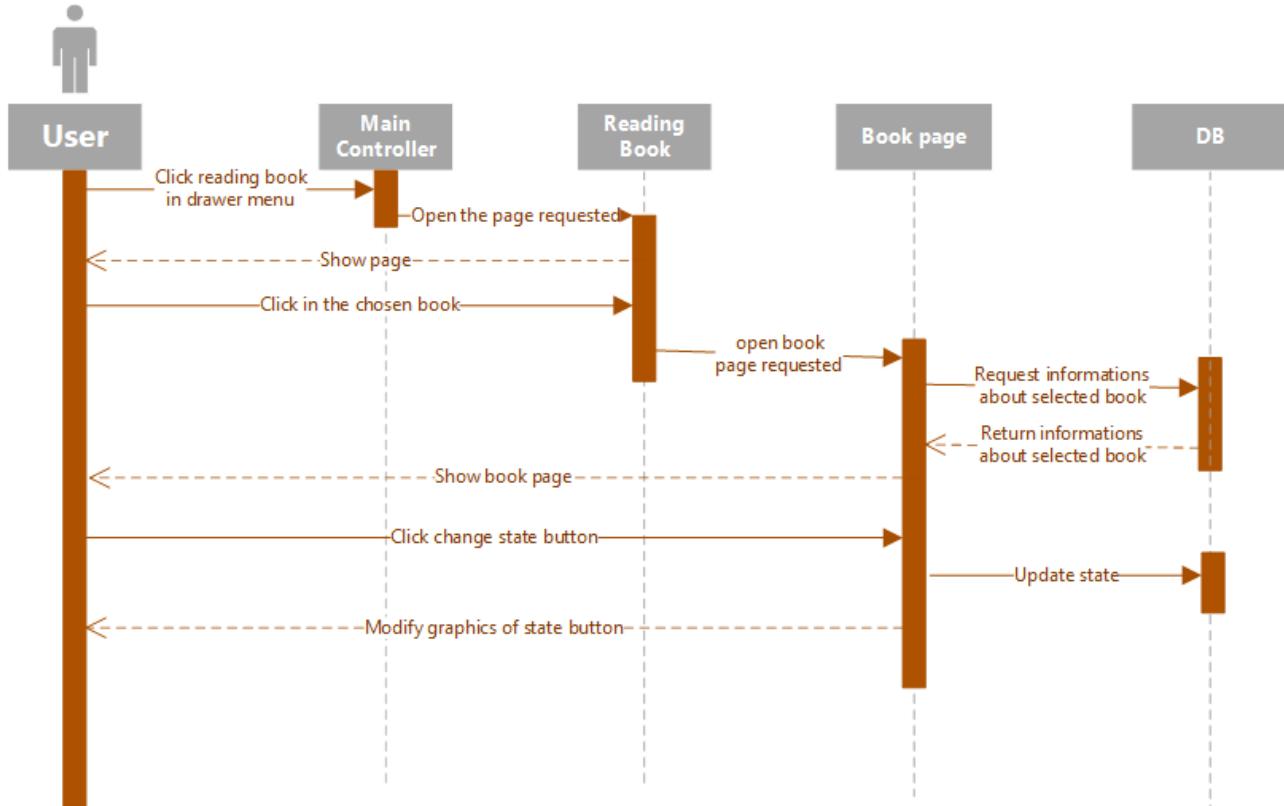


2.4.5. Change the state of a book

In the personal page of book user can change the state of the book.

It can be:

- passive → book doesn't appear in "reading book" page
- active → book appears in "reading book" page
- main active → book appears in "reading book" page and if user add new note or word they will be connected to this book; only one book for times can be "main active"





3. Chapter 3 - Class diagram

In this part of document, we will see the structure of the database and details about the tables architecture.

3.1. Database

The database is used to organize all the information important for the application for the management of the books and notes, for that reason we created a local database using sqlite and we divided the information into tree tables one for the books, one for the dictionary and one for the quotes.

Books	Dictionary	Quotes
<ul style="list-style-type: none">- id- ISBN- Title- Author- Pages- Bookmark- State (0=passive, 1=active, 2=main active, 3=read)- Imagelink	<ul style="list-style-type: none">- id- ISBN- word- meaning	<ul style="list-style-type: none">- id- ISBN- quote- page- favorite (0=Not Favorite[default], 1=Favorite)- when

3.2. Pages

In the model graph we can see all pages that has Oleggo. For each page it is visible:

- Title.
- Visible data in that page.
- Methods that user can call interacting with that page.

Moreover, in each page is possible use drawer menu shifting the fridge from left to right or pressing the button on top left.

Now, for each page, we see a brief description:

- **Home** → This is the first window appear when user open Oleggo, it includes two pages selectable using the tab menu. In “Library” we can see all books inserted in Oleggo in alphabet order independently of their state. In the other page, “Notes”, we can see all favorite notes inserted in Oleggo order by the date of insertion.
- **Add New Book** → In this page we have ad edit text where we can input the ISBN of book we want inserted. We can also use the bar code reader pressing an other button to read the ISBN.
- **Add Note** → In this page we can add a note connected to the “main active” book. We can write the note or using the speech recognition (and eventually modify the text

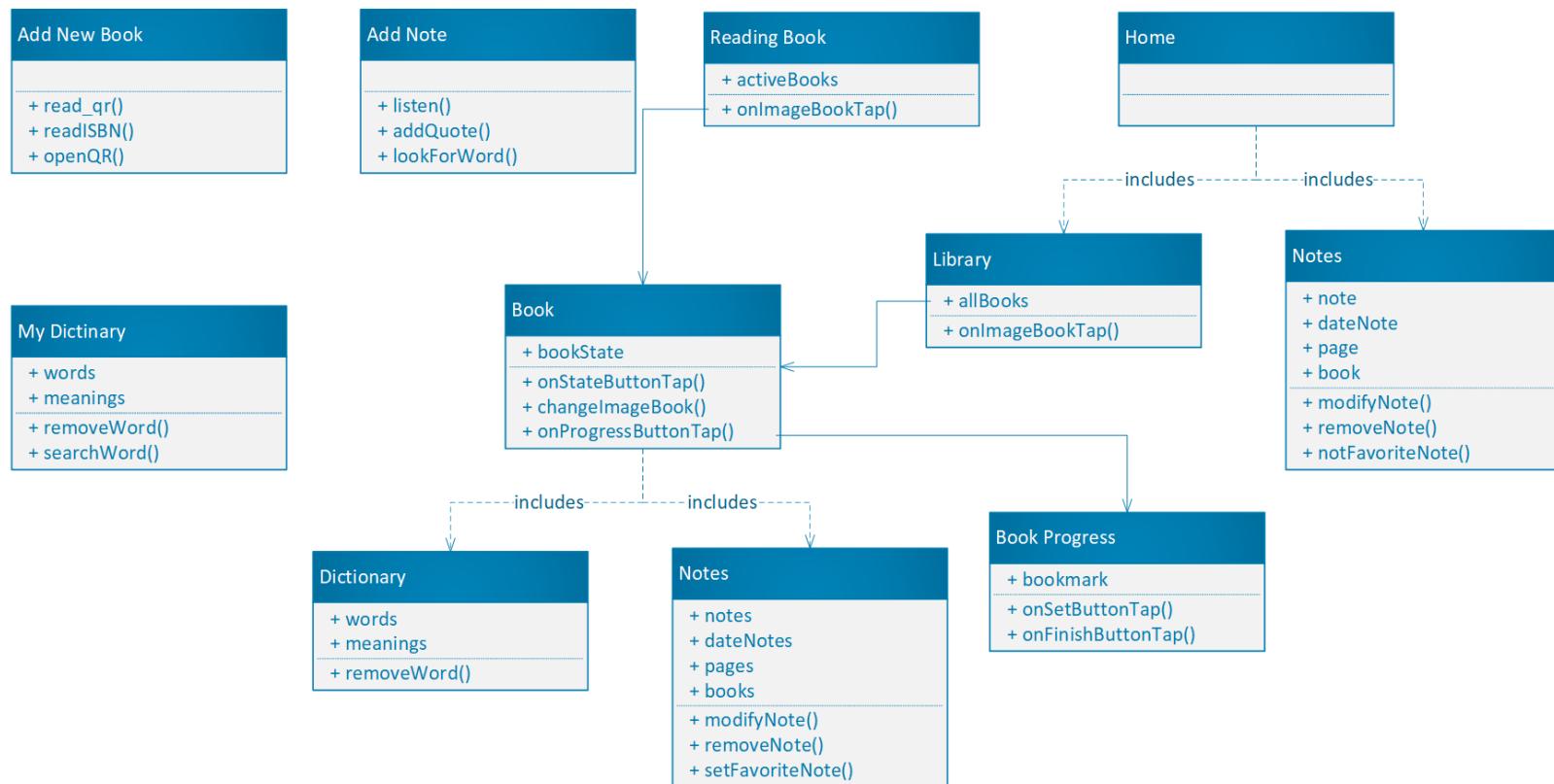


translated from the voice). When the note is added successfully an alert advice us and the text box will be cleaned while nothing happen if we don't have a "main active" book.

- **Book** → In the book page we can see the information about a single book, the page where is the bookmark, how many notes and word are connected to this book and the state of this book. Moreover, we can change the image of book from this page, it's useful if the online library didn't found any image. The last thing we can do in this page is open the "Book Progress" page.

Then from this page we can move to other two pages using the tab menu on the top of the page, "Dictionary" and "Notes". The first page contains all words in dictionary connected to this book and we can chose to remove one of them (but if this word is connected to other words in the global dictionary we will continue to see it). The second page contain all notes connected to this book, in this page we can do the same things we can do in the "Notes" page included in "Home". The only difference is that here we can add a note in the preference notes rather than remove it from preference notes.

- **Book Progress** → In this page we can modify the page where is the bookmark.
- **My Dictionary** → In this page we have all words added in the every book with its meaning order in alphabetic way. Moreover we can search a specific word or a set of words writing a part of it or them. Finally from this page we can remove a word definitely, it's means that if we remove here a word we will not find them in the "Book" page because we remove each occurrence in the database of this word.
- **Reading Books** → In this page we can see the list of all "active" book and, if we press in a book, we can reach the "Book" page with all information about the page chosen. This page is useful because we can see only the "active" book and not all book added in the past and find quickly a specific book we are reading in this moment.



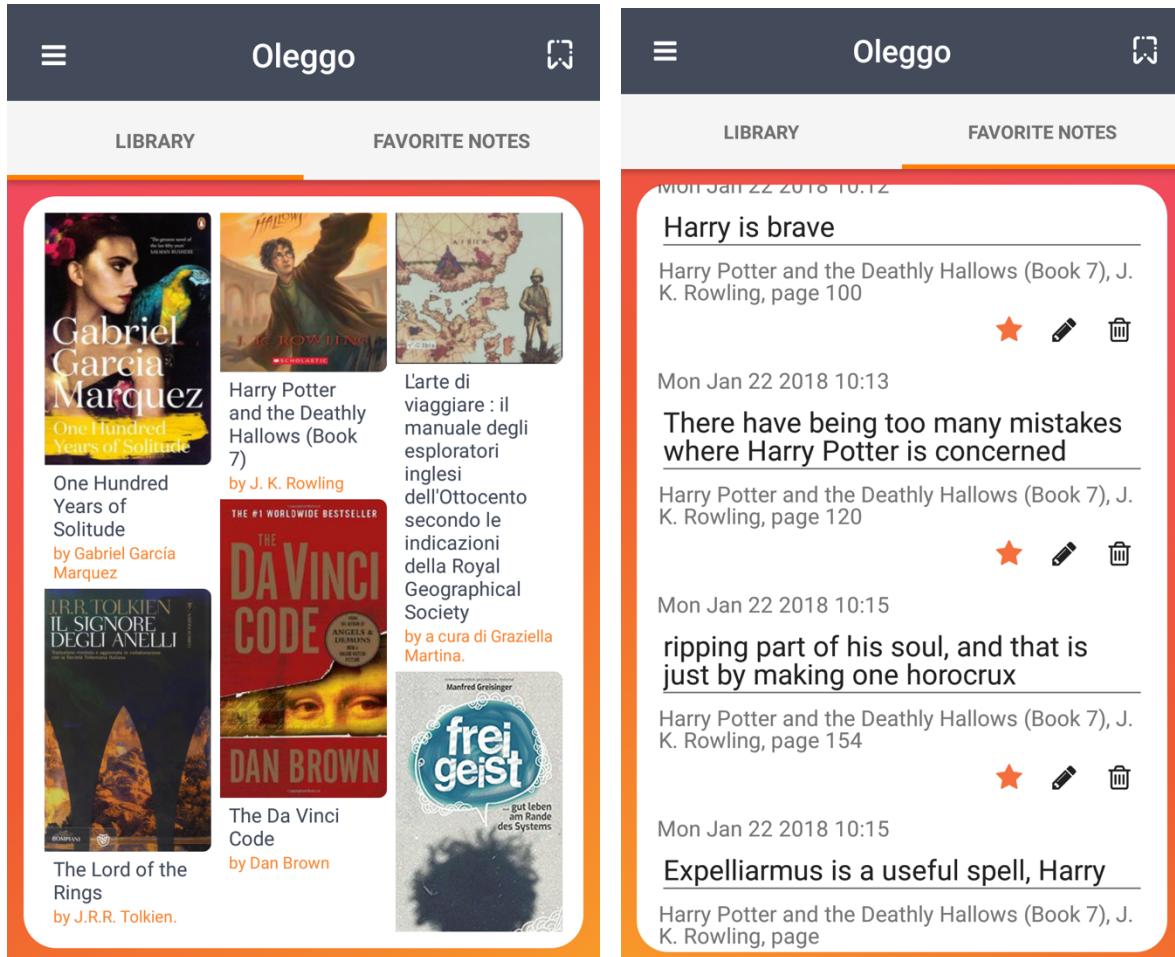
4. Chapter 4 - User Interface Design

In this chapter we will see the user interface of each page showing how use the functionalities of Oleggo.

To major details about how work the different interfaces and how happen if user wrong to insert data is explained in the section 2.3 and 2.4. Here we are interest to explain the functionalities from the point of view of the user.

4.1. Home Page

This is the first page that user sees when open Oleggo. It is divided in two sub-pages using a tab menu:





4.1.1. Library

This page shows all books inserted in Oleggo independently from their status. For each book we can see a cover image (if it is present on internet when user adds the book or if user adds after a personal image for the book), the title of the book and the name of the author.

All books are ordered by title.

4.1.2. Favorite Notes

In this page we can see all favorite notes of all books order from the newest to the oldest. For each note we have the date when we inserted it, then the title, the author and, optionally, the page of the book which the note is connected.

Below the note we have three buttons:

- Star – to remove, or add again, the note to the favorite. When we remove the note from the favorites note, it doesn't disappear immediately but only when we recharge the “favorite notes” sub-page, in this way if we press the star by mistake we can press again to add the note to the favorite and not need to go to the book page.
- Pencil – to save the modifications of the note if we want to change something. However, the date of insertion doesn't change.
- Trash bin – this button simply removes the note.



4.2. Book

This is simply the personal page of a book.
It is divided in three tabs and a full screen dialog:

4.2.1. *Information*

When we select a book in the home page or in the reading book page this is the first sub-pages that appear. It shows general information of the book:

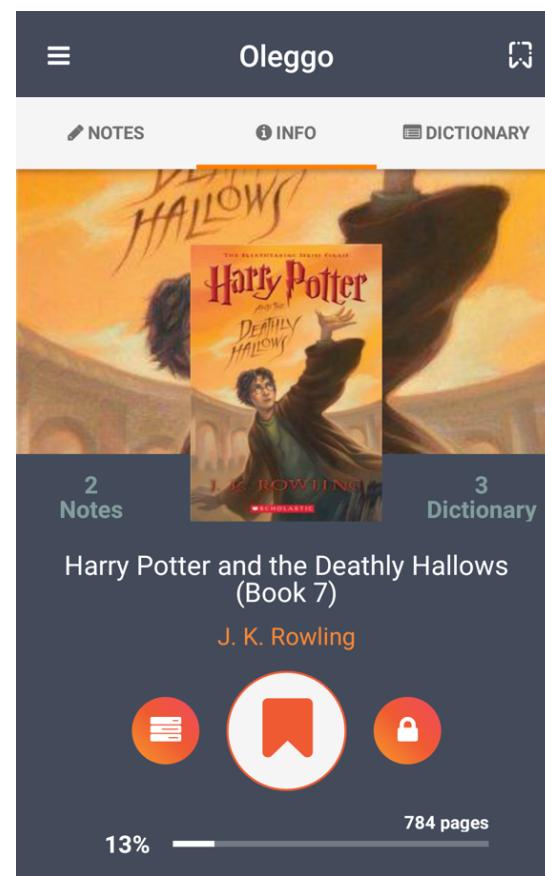
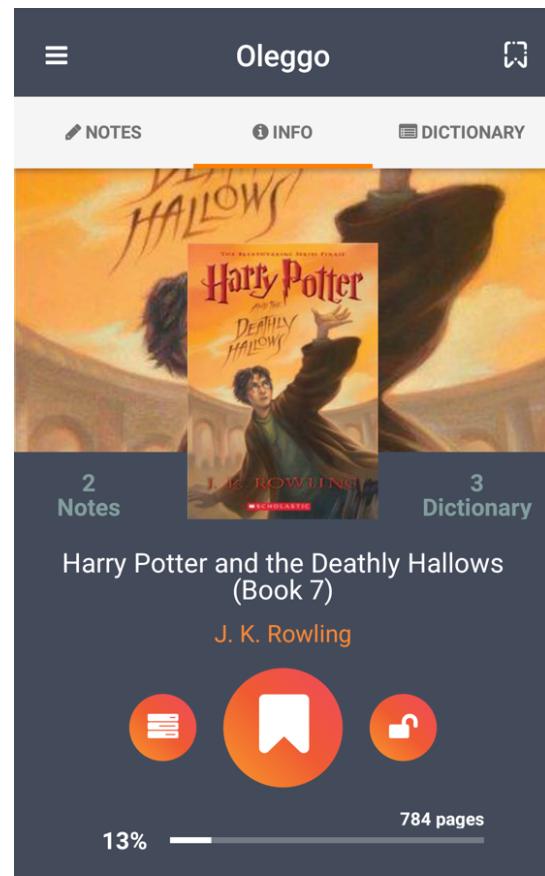
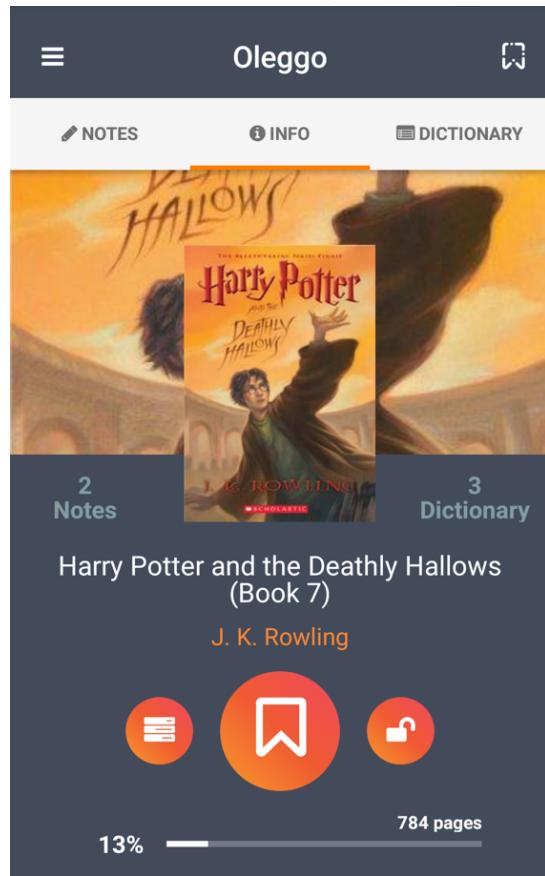
- Title.
- Author.
- How many words and notes are linked to this book.
- Status of the book (we will see it in details below).
- How many percentage of book we already read of this book.

In this page we can some information about the book:

- Pressing in the central image we can select a new picture for the cover of the book.
- Pressing the left button, it appears the update progress screen (we will see it in details below).
- Pressing the central button, we can change the status of the button.
- Pressing the right button, we can change the status of the button in main active.

A book can have three statuses:

- Passive – it is the default state when book is added. With this state the book is showed only in the library list in the main page.
- Active – when we put active one book it is showed also in reading books page. We put active the book we are reading.
- Main Active – the main active book is the book which are linked all the notes and words that will be added by us. It is showed in reading books page like the active.
- Obviously we can have more passive or active books but only one main active book.

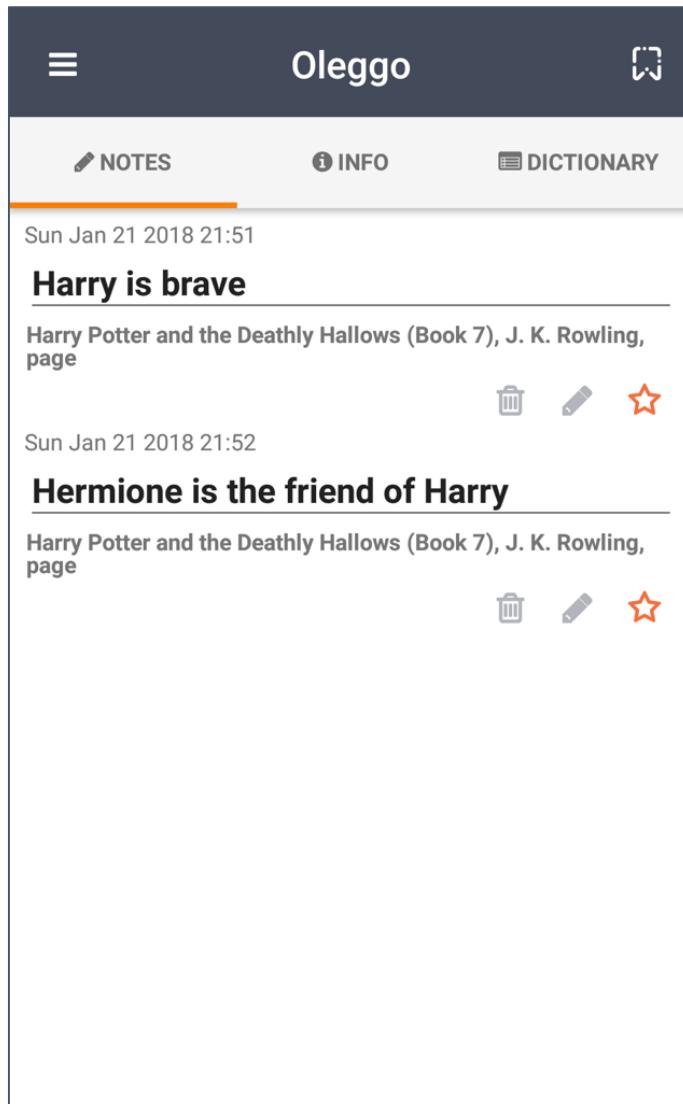


4.2.2. Notes

If we press notes in the tab menu or if we shift the finger from left to the right, we arrive in the notes sub-page.

Here we can see all notes linked to this book. Like in the home page here we can modify (and save) and remove any note.

We can also add or remove a note from the favorite. If we add it obviously we will find it in the home page the next time we open it.





4.2.3. Dictionary

If we press dictionary in the tab menu or if we shift the finger from right to the left we arrive in the notes sub-page.

Here we can see all words linked to this book with their definitions in alphabetic order. We can't search a specific word like in the dictionary page and if we remove a word here we just remove the word linked to this book. It's mean that if the same word is linked to different book if I remove it from here I will continue to see it in the dictionary page or in the personal book page of all books that are linked with this word but not anymore in this book.

The screenshot shows a mobile application interface for a book titled "Oleggo". The top navigation bar includes a menu icon, the book title "Oleggo", and a bookmark icon. Below the title, there are three tabs: "NOTES", "INFO", and "DICTIONARY", with "DICTIONARY" being the active tab. The main content area displays two entries:

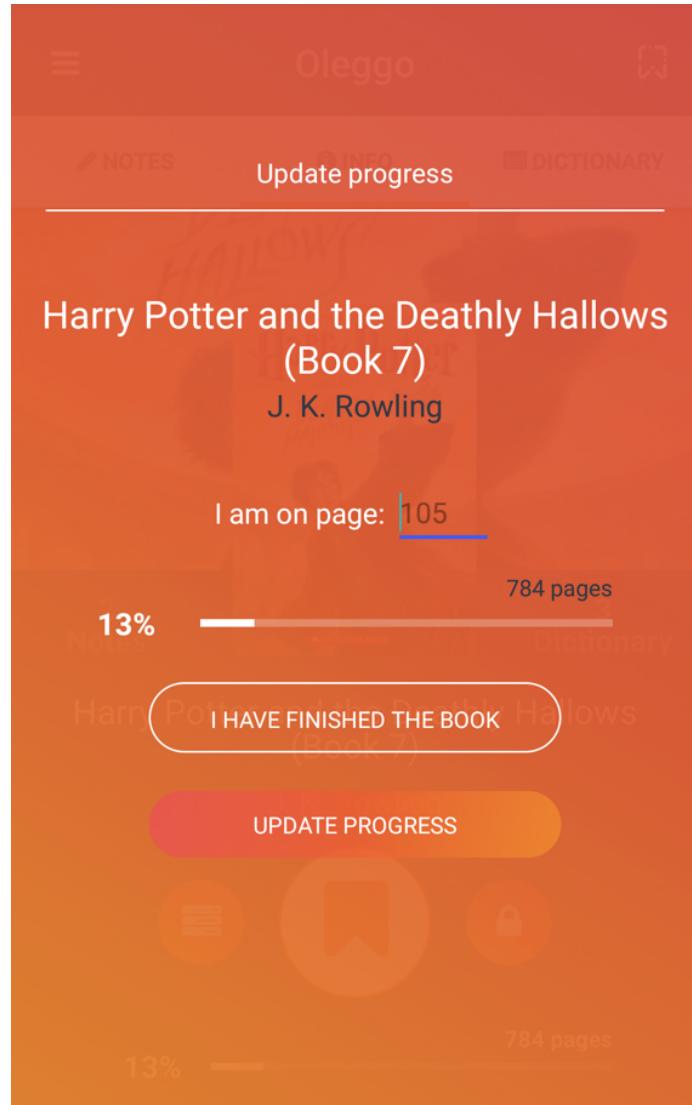
- Brave**
Strong in the face of fear; courageous. Delete icon
- Mage**
fantasy A magician, wizard or sorcerer. Delete icon



4.2.4. Update progress screen

In this screen we can update the bookmark of the book inserting the page where we are arrived to read and pressing the update progress button.

If we finish the book, we can directly press “I have finished the book” button.

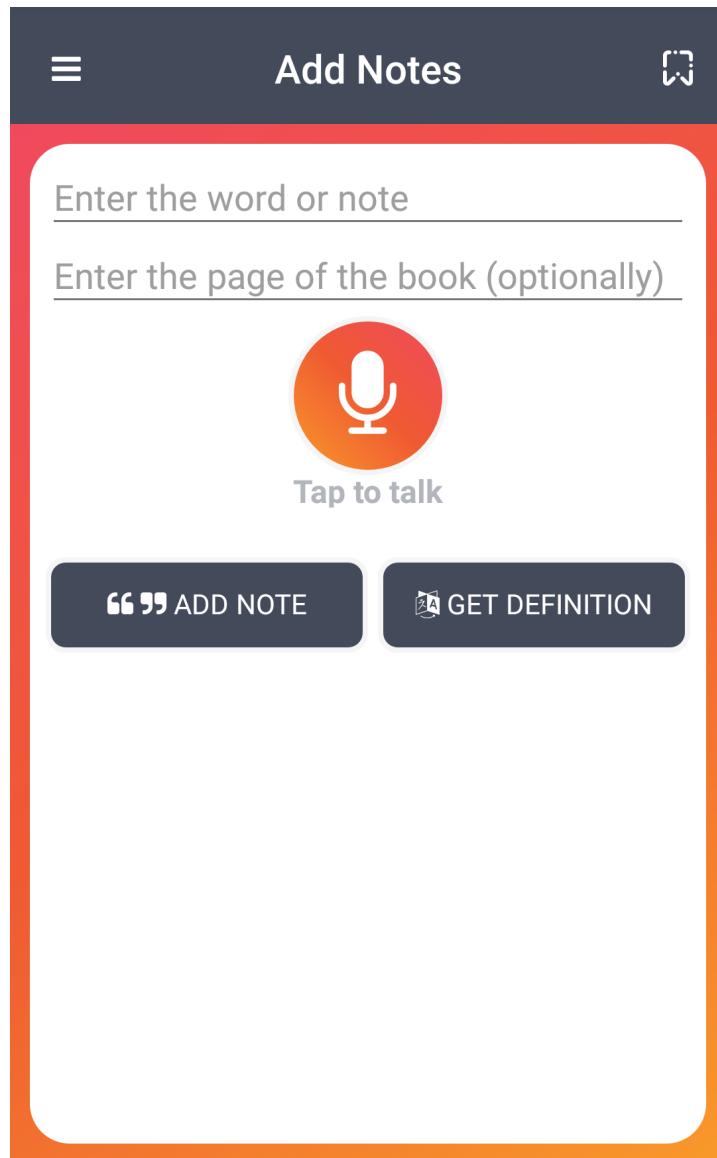




4.3. Notes

First of all, we must know that we can reach quickly this page from any page using the icon on the top right page. In this way we don't lost time if we must write a note and we have a few time.

In this page we can insert a new note or search the definition about a word that will be added to the dictionary.





- *Note*

We can write the note directly in the text box on top of the page or we can press the microphone and speech. Our voice will be translated in text and printed in the same text box, so we can modify it. We can also insert a page that will be showed together the note, but it is optionally.

Then we must press “add note” and note will be added and connected to the main active book.

- *Word*

Like to the note we can insert the word we want to know the definition writing it or using the speech recognition.

Then we must press “get definition” to look for the word online, in Wiktionary, and add it the dictionary.

If the application doesn’t found the word in Wiktionary it adds the words to the dictionary without definition.

If we don’t set a main active book, we can’t add word or note and if we try to add one of them an alert advice us to choose a main active book before.



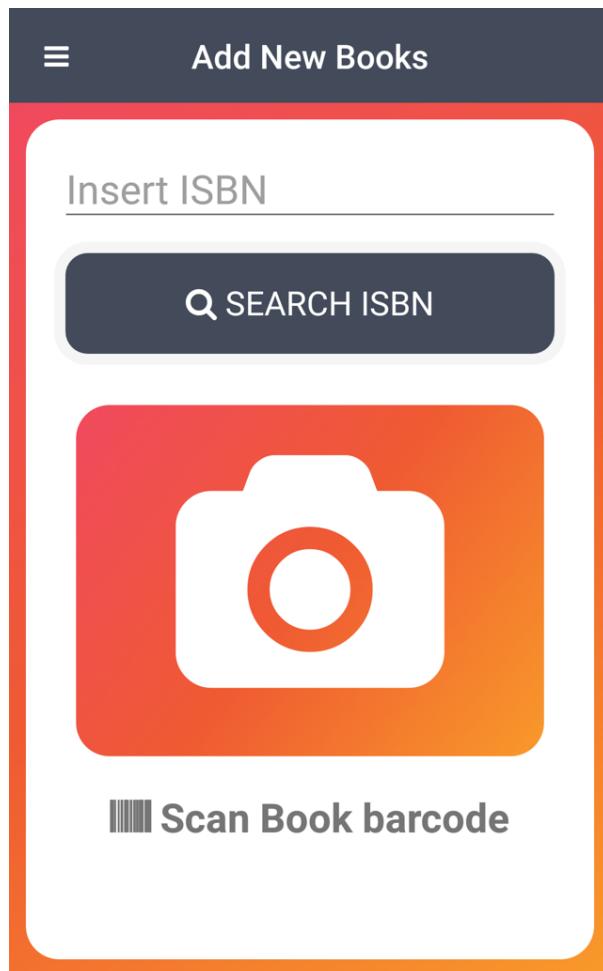
4.4. New Book

In this page we can add a book inserting its ISBN. It can be inserted using the text box or reading the bar-code of the book. To use the bar-code, we must press the image of photo camera and point the smartphone on the bar-code of the book checking to center it.

After inserting the ISBN we must press the search ISBN button that look for information about the book in two different online library.

Also the book is found can happen that it isn't present in the information the author or the number of pages. In these two case application adverts us with a dialog and asking us to enter the name of the author or/and the number of the pages. In any case application asks us to confirm the number of the pages found in the information if it is.

In other case, if the ISBN isn't present in the online libraries used, application asks us to insert title, author and number of pages.

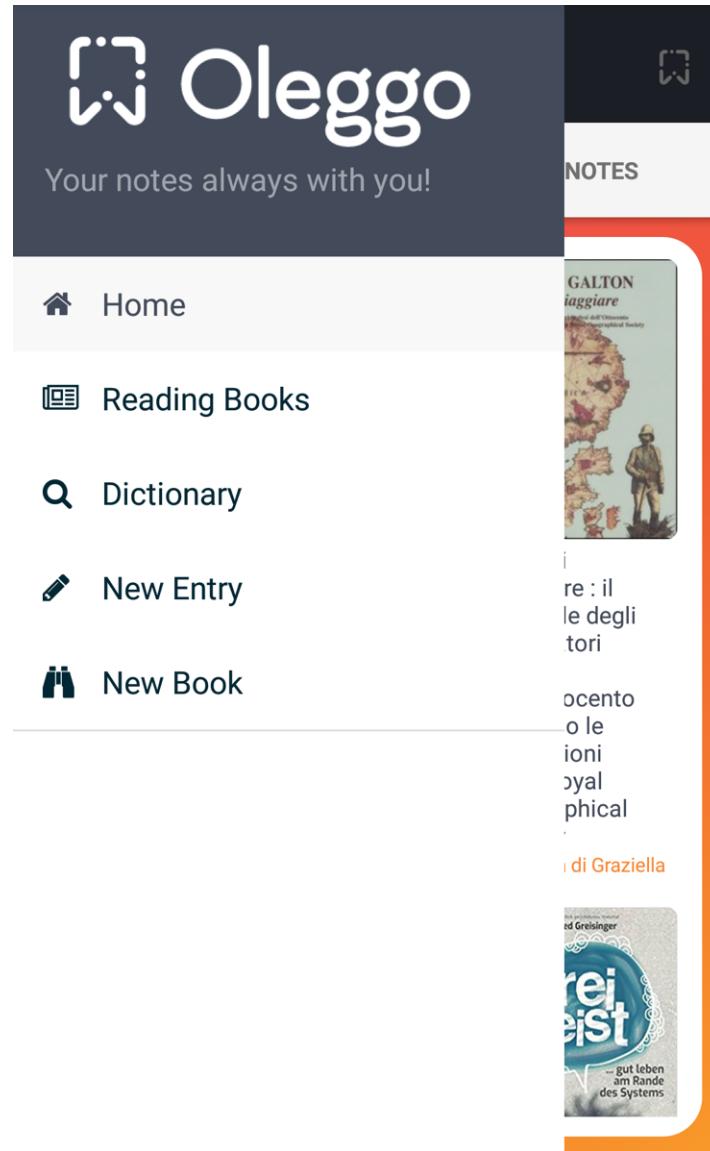




4.5. Side Drawer menu

We can access to the drawer menu pressing the icon on top left of the page or shift the finger from left to the right.

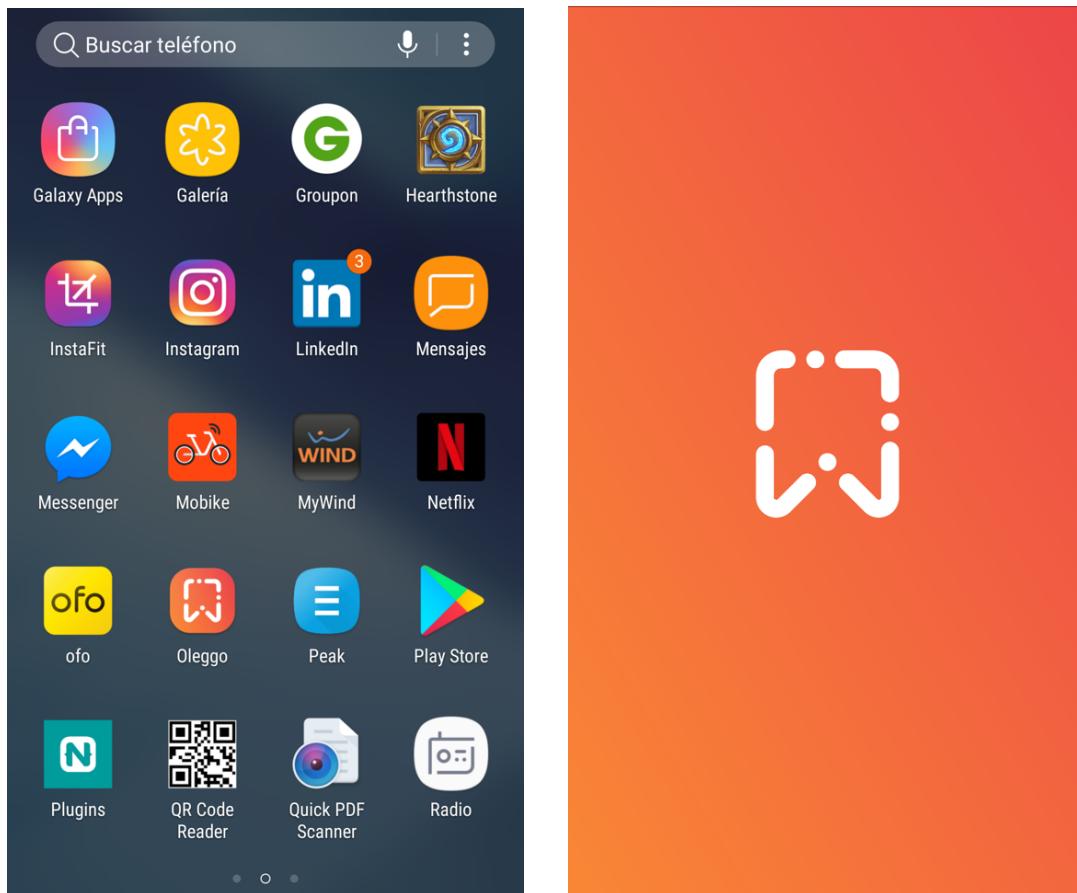
Using it we can navigate between the different main pages of the application.





4.6. Complementary screens

Here we can see the icon of the application in the main menu of smartphone. And next to we can see the splash of the app that appears when we open Oleggo, before the first page is ready.





5. Software System Attributes

5.1. Reliability

The system saves the data of user only in the local device, so Oleggo doesn't have any type of reliability for the moment.

5.2. Availability

The notes and words searched are always available for the user because they are saved in the local device.

For the online services obviously don't depend by Oleggo. Anyway we use Wiktionary for search the definition of the words and it is very difficult that it isn't available for a problem. For the other online service, the ISBN search, we use two external services, in this way not only we can have more probability to find a book but it is also less probability that each services are down in the same moment.

5.3. Security

The app doesn't save or share anything online so the personal information that user inserts are inviolable.

5.4. Maintainability

The system will be very maintainable thanks to a modular structure. The application will be documented in a way that will truly help the future developers, explaining how the application works and how it has been developed.

5.5. Usability

The application is meant to be as easy and intuitive to use as possible. To get it easier we also put icons in each button.

All the users can provide feedbacks to improve the app rating it in Play Store.

5.6. Portability

The software for the system will be developed using NativeScript. It is a cross platform that use JavaScript, XML and CSS. It's mean that we can build our application for Android and IOs.



6. References

- (1) Design and Implementation of Mobile Applications course material [Online],
<http://home.deib.polimi.it/baresi/dima.htm>
- (2) Official website of NativeScript with useful guides and plugins,
<https://docs.nativescript.org/>
- (3) Repositories when we took different pieces of code for graphics and template components, <https://github.com/EddyVerbruggen> -
<https://github.com/TobiasHennig>
- (4) API used for search the books using ISBN, <https://openlibrary.org/developers/api> -
<http://xisbn.worldcat.org/xisbnadmin/doc/api.htm>
- (5) Plugin for search the meaning of the word, <https://www.npmjs.com/package/word-definition>
- (6) Icons, <http://fontawesome.io/>