
COLCHESTER ROYAL GRAMMAR SCHOOL

BOOKIPEDIA

GABRIELE CORSO

CANDIDATE NUMBER 3112



Table of Contents

Analysis

Introduction to the problem.....	6
Description of the app	7
Critical path.....	7
Analysis of existing systems.....	8
Reading activity.....	8
Library	9
Books database.....	10
Prospective users.....	11
Survey	11
Results.....	13
Specific objectives.....	16
Reading activity.....	16
Library	17
Backend	17
Data Flow Diagram	18
Initial Entity Relationship Diagrams	19
Local Database	19
Backend database.....	19
Data Dictionary	20
Objects.....	22
Other research.....	23
eBook file format	23
File Security Methods	24

Design

UI – User Interface Design	25
Register activity	25
Login activity	26
Home activity.....	27
Library activity	28
Browse activity.....	29
Genre activity.....	30
Search activity.....	31
Book activity	32
Author activity	33
Reading activity.....	34
System Flow Diagrams.....	37
Register and Login process	37
Library	38
Reading	39
Classes Inheritance	40
Classes description.....	42
Local Database Structure	45
Table creation SQL.....	45
SQL Queries.....	46
Server database structure	47
Final entity relationship diagram	47
Tables creation SQL	48
Books retrieval SQL query.....	50
Authors retrieval SQL query.....	52
Reading update SQL query.....	52
User control SQL query.....	53
Webserver structure.....	54
Security	55

Notes.....	56
Books file	57
Test plan	58
RegistrationActivity.....	58
LoginActivity	59
HomeActivity	60
LibraryActivity.....	60
BrowseActivity.....	61
GenreActivity	62
SearchActivity	63
BookActivity.....	63
AuthorActivity.....	64
HelpAndFeedbackActivity.....	64
SuggestionActivity.....	65
ReadingActivity	65
RatingActivity.....	67
Epub to txt converter.....	68

Testing

RegistrationActivity.....	69
LoginActivity	74
HomeActivity	78
LibraryActivity.....	81
BrowseActivity.....	84
GenreActivity	87
SearchActivity	89
BookActivity.....	92
AuthorActivity.....	95
HelpAndFeedbackActivity.....	96
SuggestionActivity	98
ReadingActivity	101
RatingActivity.....	109
Epub to txt converter.....	112

Evaluation

Evaluation against specific objectives.....	113
Reading activity.....	113
Library	115
Backend	117
User Feedback and Discussion.....	118
Overall Evaluation.....	122
Extension	123

ANALYSIS

Introduction to the problem

In recent years the music industry has taken a clear path. First, songs were sold on their own as CDs. In 2001 the first iPod and other mp3 players were presented and people liked the idea of having a single device that can store an incredible amount of songs. Finally in recent years most of the customers have switched to mobile apps which allow them to have their music directly on the smartphone so they do not have to use and carry another device.

The same is happening to the book industry. While they used to be sold all on paper, after the Kindle launch by Amazon in 2007 more and more people have switched to read books on a single hardware that can contain thousand of books in its memory. The last step, however, has not taken place yet and people are reluctant to the idea of reading on their phones or tablets. I think that it is mostly a problem of habits and we will start reading books on our phones soon.

The lack of mobile applications online that offer a way of legally download and read eBooks in public domain for free drove me to the decision of making an eBooks streaming android app. Users will therefore be able to choose books between the ones that the app would suggest them according to their preferences or just by searching them and read them directly on the app.

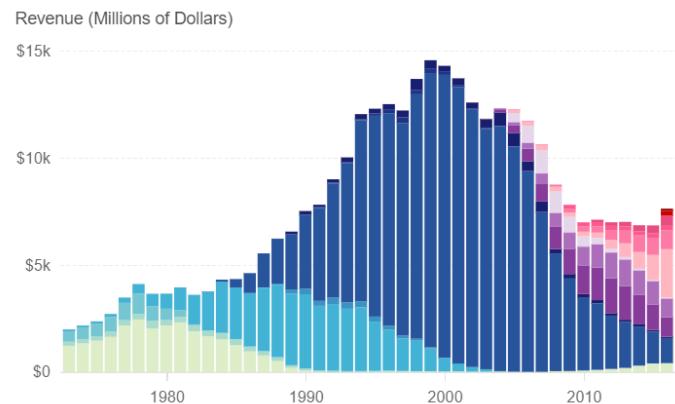


Figure 1 – U.S. Recorded Music Revenues by Format
Light blue represent cassettes, blue CDs, purple download and pink streaming. Graph and data by RIAA.

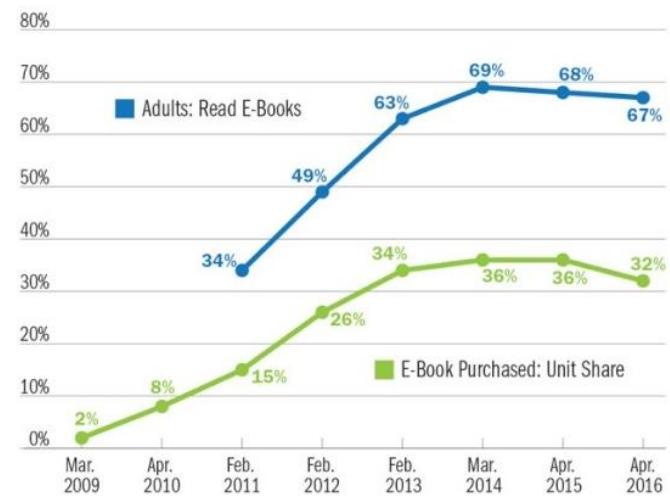


Figure 2 – EBook unit share trends
Graph and data by Codex Group.

Description of the app

The app will therefore offer a library which suggests new books to the users according to their previous choices and ratings other users similar to them gave to certain books. The app will locally store the books that the users start to read so that they can continue them even without an internet connection.

At the beginning the platform is going to have only books in public domain, which, according to the Copyright, Designs and Patents Act (1988), are all the books for which have passed more than 70 years after the death of the creator.

Critical path

The three main stages I identified for the creation of the application are:

1. Creation of an eBooks reader: the part of the application where the user will actually read the books.
2. Creation of library: the interface with which the user will be able to log in and choose a book with some algorithms for the research and suggestions of eBooks for the users.
3. Creation of the backend: which will store the eBooks library as well a database with tables relating to the eBooks, authors, genres and users and will connect directly with the app. Addition of an adequate number of eBooks to the backend.

The order is not strict as most of the stages are independent from each other, but completing them in the order that I have presented them with will help the intermediate testing process.

Analysis of existing systems

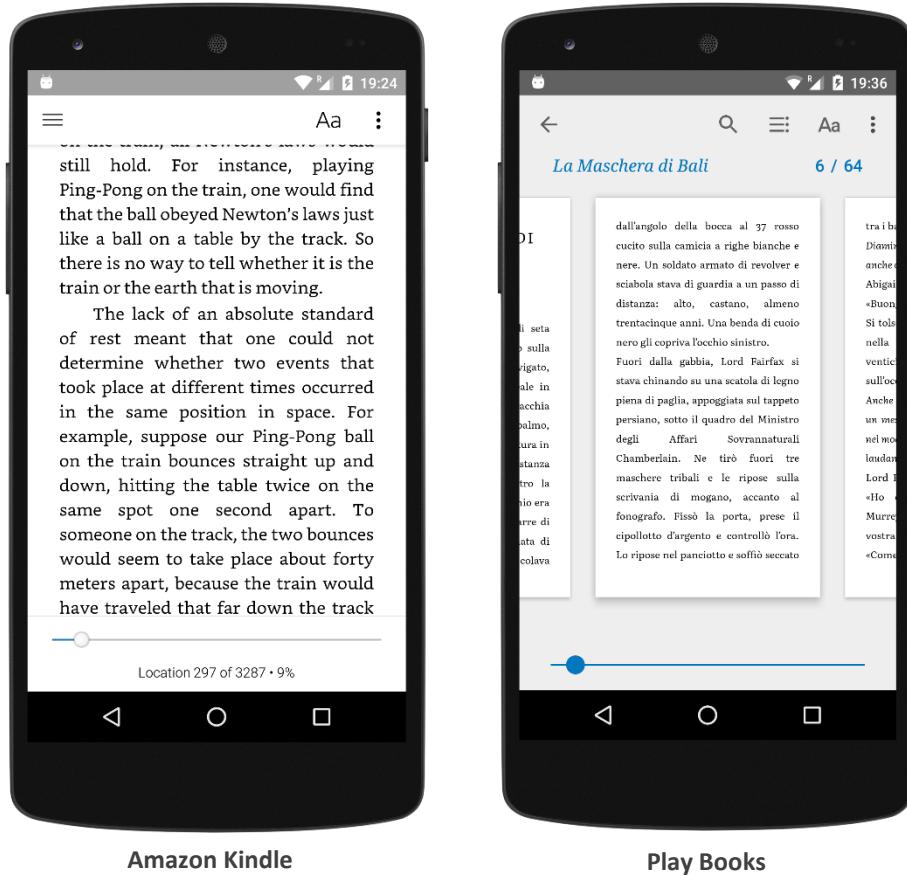
There already are few applications on the Google Play Store (official android app store) which allows users to buy and read eBooks on smartphones. In particular, there are two applications that are more popular and, in my opinion, better made than the others: Amazon Kindle (by Amazon) and Play Books (by Google). For each of the stages I identified before, I analysed how they were implemented in these apps.

Reading activity

The two apps both offer a clean reading activity with touch controls (both touch and swipe) on the left and right that allow the user to change the current page. Moreover, by selecting some words these can be highlighted, translated and notes can be added.

This can be expanded with a touch in the middle leading to the appearance of a control bar at the top and a progress bar at the bottom. The progress bar at the bottom also offers the possibility to move to another position in the book. The controls bars of both applications offer:

- A font control button through which text size, colour, font and brightness can be changed
- A button for the access to the table of contents of the eBook
- A button for the insertion of a bookmark in the current position



Library

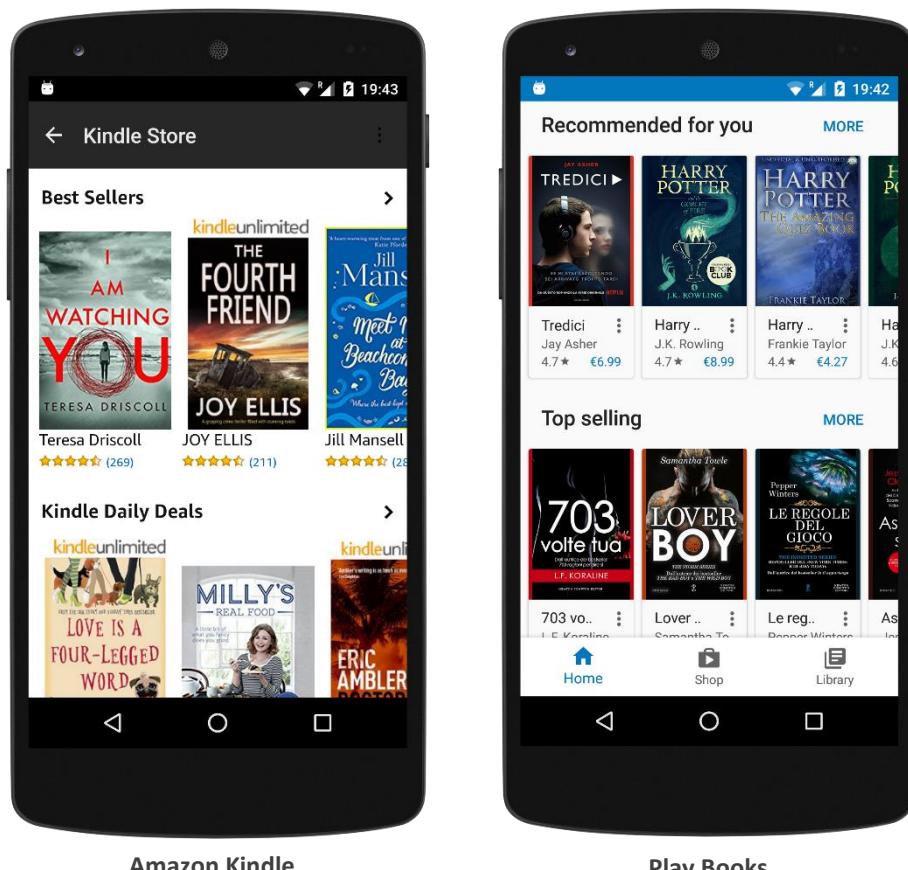
For the login both apps require their own company accounts: Kindle asks for an Amazon account while Play Books a Google one.

Once you are logged in both apps have a sort of questionnaire to collect an overview of the user's preferences in order to be able to make some suggestions of books he may like. The user is asked what genres of books he reads and particular books he liked between those genres.

Then the user enters in the library where books are suggested to him according to different criteria, browsing between these two and other apps (Spotify for example) I found the following criteria used in some of these:

- a. Best sellers or most popular
- b. Special deals: eBooks apps sell books individually every week/month some of them are on sale
- c. New releases
- d. Recommended for you: based on previous choices

Both apps offer than a shelf section where the user's recent books are displayed. However in none of them the shop/library is really integrated inside the app: Play Books opens the Google Play app for selecting an item while Kindle has a Web View that renders to the Amazon website.



Books database

The two apps I am analysing are supported by a stores (Amazon and Google Play) with thousands, maybe millions, of eBooks behind in many different languages. Books mainly sold individually, the only exception is a ‘kindleunlimited’, a subscription offered in the Kindle app which allows users, paying £7.99 a month, to have many free eBooks available and discount on others.

Both apps then offer a function of research for a particular book and for each eBook there is a small description. However none of the apps have pages for each author with a small description and their works.

The backend structure of both apps is obviously hidden to the user.

Prospective users

The prospective user for this application is anyone with an android smartphone or tablet, which will be able to download the app from the Google Play Store once it will be published. However, in order to have more feedback during the development, I decided to identify some specific users in Eesha Irfan, Daniel Gibbons, Jeffrey Yuen and Nikini Kottegoda, which showed interest in the project and would be available to discuss functionalities of the project at any time.

Survey

In order to collect general impressions, select which functionalities are key to be developed and find some new ideas, I decided to make the following online survey using Google Forms:

How do you currently read books?

- Paper
- eBook reader (like kindle)
- I don't read books
- Other: _____

How many books a month do you usually read?

- More than 3
- 3
- 2
- 1
- Less than 1

Would you read books on your smartphone?

- Yes
- No, the screen is too small
- No, I prefer the paper feel
- Other: _____

Functionalities you would like the app to have (that you would use) while reading a book on the smartphone/tablet?

- Change font
- Change characters dimensions
- Highlight text
- Take notes
- Rate a book once you have read it
- Add bookmarks
- Other: _____

How would you like to choose your books?

- Directly search the title/author
- Suggestions according to your previous books
- Navigate between genres or themes
- Most popular or best rates books by other users
- Read brief description of the book
- Other: _____

Any suggestions for the app name?

Your answer _____

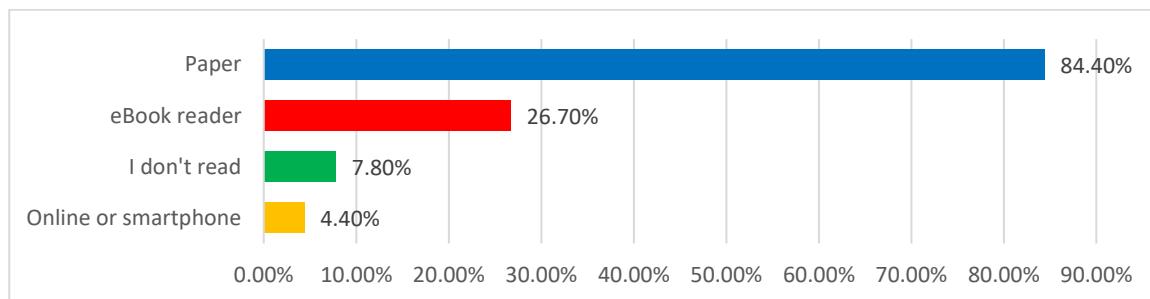
Any other suggestion of features you would like to see in this kind of app?

La tua risposta _____

Results

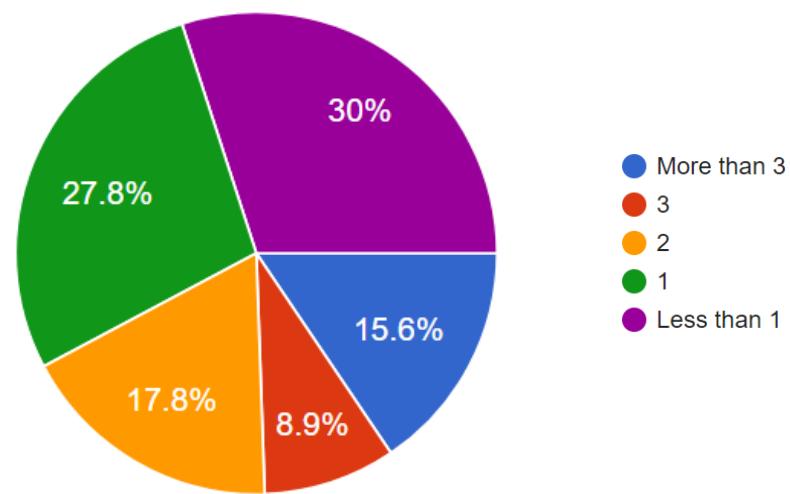
I sent the form to all the sixth formers in my school and it was answered by 91 students in a week. The following were the results:

How do you currently read books?

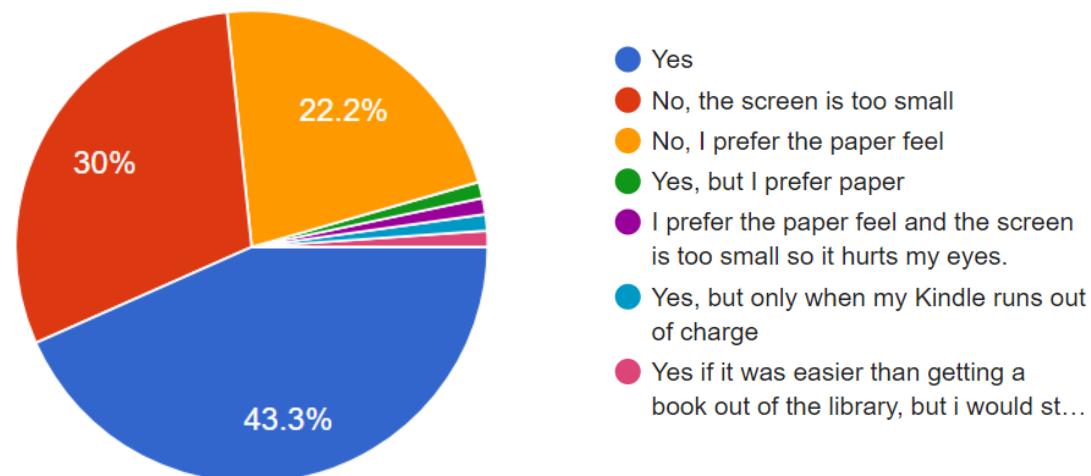


As expected the results show that the vast majority of students still read their books on paper, a small proportion of eBooks readers and less than 1 every 20 use a smartphone.

How many books a month do you usually read?

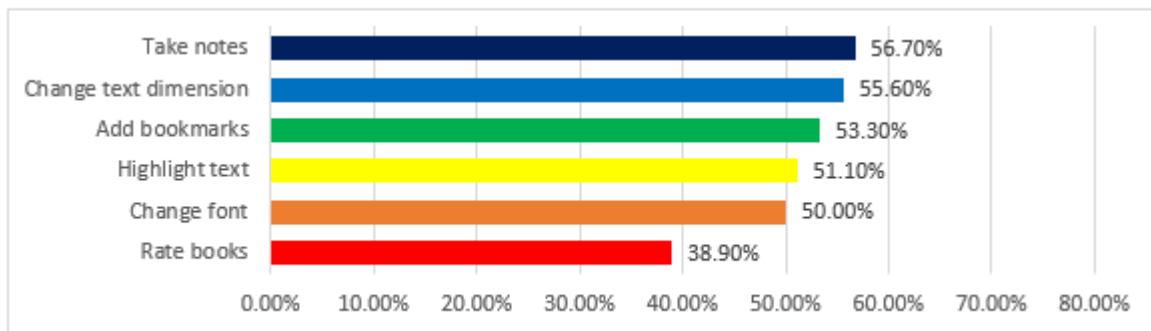


Would you read books on your smartphone?



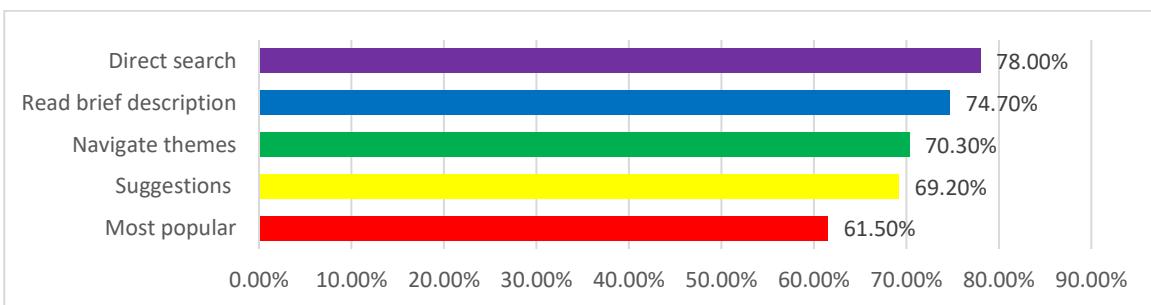
The fact that almost half of the students answered that they would read on their phone, considered that the less than the 5% already use another mobile platform, is an encouraging indicator of the presence of a lot of potential users but at the same time the need for some kind of differentiation with the other apps.

Functionalities you would like the app to have (that you would use) while reading a book on the smartphone/tablet?



Results show that the only functionality that less than the 50% of the users say they would use is the books rating, which, however, will be useful also when suggesting the user new books and therefore will be implemented in the app. All the other factors have been selected with a fairly high proportion and therefore will be included in the project.

How would you like to choose your books?



All the options received a high percentage suggesting me that the best would be to develop them all.

Any suggestions for the app name?

Bookify (6)	B00K<3	Smartbook (3)	Bookworm
Lumberjack (2)	Bookery	Bookie	Digital Read
The Library	BookStreamer	Ready	Dotify
Read Stream	All you can read	Tome-y	The Bookcase
BookBashers	Just Books	Delphi	Binge reader
MareBooksEU	Bookity	Recurring Reading	Books on hand
BookReadingApp	bookbook	Book-it	Readit, just read it
The mobile library	Boookz	Bookface	Bookipedia

Any other suggestion of features you would like to see in this kind of app?

- Access from computer and tablet (2)
- Temporary book downloads for use offline.
- Being able to report or comment
- Auto-reject calls
- Yellower colour theme so that its less painful for the eyes to concentrate (2)
- Page flipping animation
- Podcasts or audiobooks
- Customised for mobile (so there is not a need to zoom)
- Access with Facebook
- Availability across multiple devices for the same account
- See what friends are reading
- Customise-able themes (2)
- A way to change the contrast of the page (3)
- Ability to see what friends are reading or create book clubs
- Possibility to get the book read automatically
- Offline functionality
- Night-Time reading mode (3)
- Include audio books
- Be able to purchase paper copy of the book ie link to Amazon
- Page numbers

Specific objectives

With the help of the survey's results, I began to decide for each stage which function should I implement in my app:

Reading activity

1. Read the customised .txt file and display it clearly in an activity, with a text dimension that is absolute and not relative to the size of the page (very large for tablets and too small for small smartphones) in order to make the books easily readable on any device.
2. Memorise the last position that will be retrieved whenever a book is reopened.
3. Navigation between pages with swipes or touches on left (previous page) and right (next page) sides of the screen and access to the control bars with a touch in the middle.
4. Quick navigation through the book pages using a bottom bar that compares with the controls after a touch in the middle of the screen.
5. Display of the progress on a book (page).
6. Taking notes while reading, the note will be linked to the position of the book where it was taken.
7. Highlighting of text, which has to stay highlighted every time the user opens that book again. Users will also be able to retrieve all the highlighted pieces of text in a book and the notes in a summary page.
8. Saving bookmarks, where the user can easily come back
9. Font control through which you can change:
 - a. Text dimension
 - b. Text font
 - c. Light / Dark / Sepia themes.
10. For audio books (registration of books read out loud by a reader) while a storage of a registered voice would be really heavy in term of space and in particular to create in the first place, a function of the app that would automatically read (using some external API) the text would be a good compromise.
11. Possibility to rate a book once the user has finished reading it.

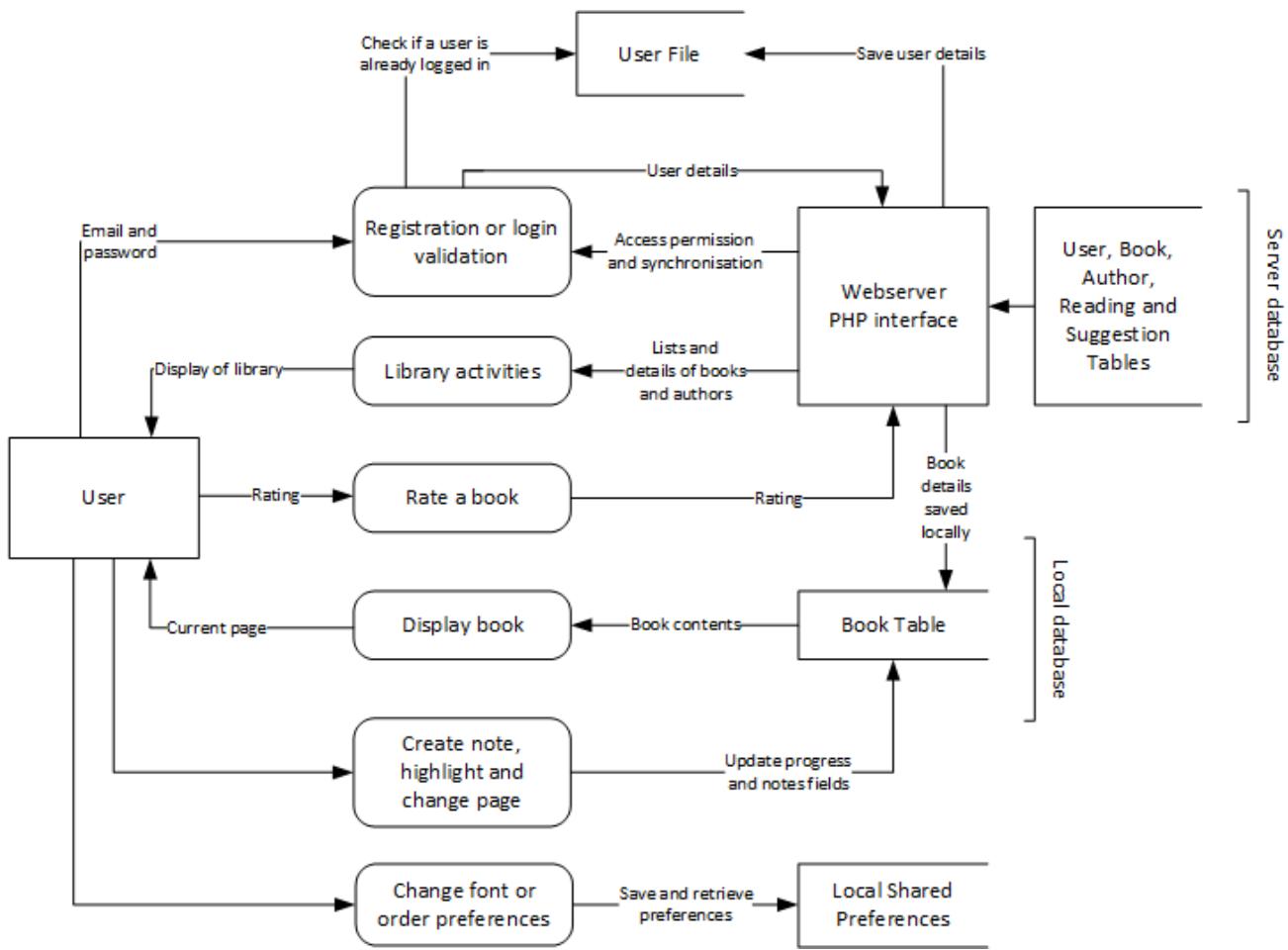
Library

1. Registration with a new app account, giving email, password and favourite genre.
2. Log in with email and password.
3. Logged user registration to avoid having to log in every time.
4. Shelf section where the books read are displayed.
5. Home section where the users are suggested books with algorithms that consider the following criteria:
 - a. Most popular: based both on the number of downloads and on the ratings of other users.
 - b. Recommended for the user: based on favourites genres.
6. Browse section where the user can navigate in the library according to genres.
7. Search section where a book or an author can be searched
8. Book overview: when a user selects a book from the library a window with info on the book is opened. This shows title, author name, year, rating, genre and description with a button to start the reading.
9. Author page: displays author's name, books and brief biography.
10. Creation of a local database that will store the data of the books the user read or is reading on the device so no internet connection is needed to finish a book started or look back at a book finished.
11. Choice for the user of the order in which books are displayed in some lists in the library
12. Possibility to suggest a book that should be added.
13. Synchronization between multiple devices so that the user can start reading on a certain device and continue on others and notes, highlights and progress are memorized.

Backend

1. Working backend with a database with books, authors and users information and all the actual eBook files.
2. Strong and secure encryption for the passwords.
3. Actively respond to calls from the app for login, registration, query of eBooks, query of authors, download of eBooks
4. Creation of a program through which book can be converted from ePub to .txt with the correct mark-up (see page 23)
5. Insert about 100 eBooks with all the related information

Data Flow Diagram



There are four main ways the application will store its data:

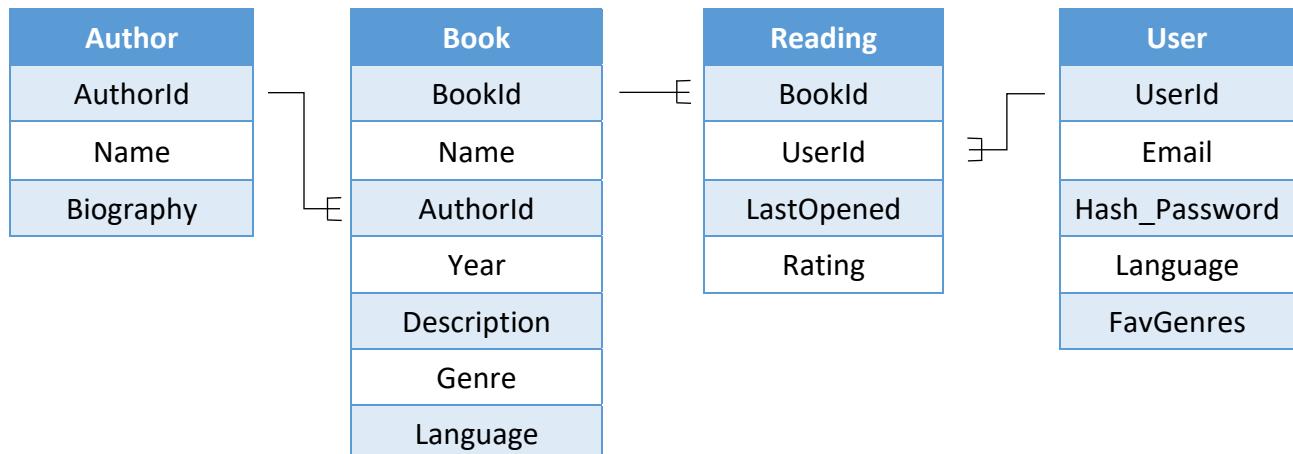
- Server database: hosted online where all the data of books, author, users and readings are stored
- Local database: that will store the data of the books the user has downloaded on the device
- User file: saved locally where the app stores the information about the user logged in so that he doesn't have to log in again every time
- Local shared preferences: of the application will store some data about font, layout and order preferences of the user

Initial Entity Relationship Diagrams

Local Database

Book
BookId
Name
AuthorId
Year
Description
Genre
Language
File
Progress
Notes

Backend database



Data Dictionary

Field name	Data type	Example	Description
User			
Email	String	google@gmail.com	Valid and verified email address
Hash_Password	String	dnjflsnlcdjcnksadn	Sequence of random characters which correspond to the hash of the password of the user
Language	String	English	Favourite reading language of the user chosen from an array of available
FavGenres	String	Romance,Action	Array of favourites genres of the user stored as single string with commas
Book			
BookName	String	Pride and Prejudice	Title of a certain book
Year	Integer	1814	Year when the book was published
Description	String	...	Brief description of the plot of the book
Genre	String	Romance	Name of the genre chosen form a definite set
File	String	...	Actual text file of the book
Progress	Integer	123	Last position of the user in the book
Notes	String	...	List of all the string one after the other concatenated by special characters
Author			
AuthorName	String	Jane Austin	Name and surname of the author
Biography	String	...	Short biography of the author
Reading			
LastOpened	Date	07/10/2017	Date when the user opened last the book
Rating	Integer	4	Rating the user gave to the book after finishing it out of 5

For the user point of view the main data he will see or have to insert is similar:

- He will have to add all the data in the user table apart for the password hash which is derived in the server from the password he inserts
- He will have access to all the information of the book table: name, author, year and description from the library and file, his progress in the book and his notes from the reading section
- Also the authors information will be available in a specific page in the library section where the user will be also able to see all the books of a certain author
- The last opened date won't be directly visible to the user, but the books he has started reading in the library will be in this order

Objects

Class name	Description
Entities	
Book	Define the properties each book object should hold
Author	Define the properties each author object should hold
User	Define the properties each user object should hold
Note	Define the properties each note object should hold
NoteList	Implementation of the structure which holds the list of the notes ordered and divided by highlights and notes and implements the different queries
Databases connection	
BookLDH	Creates the connection and manages with the local database
LoadBooks	Sends HTTP the queries to the PHP webserver for the lists of books with a given condition, which will then run the query on the server database, and then translate the JSON response to return a list of book objects
LoadAuthors	Sends HTTP the queries to the PHP webserver for the lists of authors with a given condition, which will then run the query on the server database, and then translate the JSON response to return a list of author objects
JSONParser	Implements the methods to send the JSON requests
Various classes for specific backend queries that will be written as private classes inside the activity classes where they will be used	
Storage	
UserFileManager	Singleton class (one object) that manages the connection of the app with the text file where the details of the user currently logged in are stored.
Adapters	
Various adapters to display and control the different list and grid views in the library	
Layout specific classes	
Various classes that defines particular elements of the design	
Other classes	
EmailValidator	Checks that an email typed in the Login or Registration activity is valid before this is sent to the server.

Other research

eBook file format

I considered two main possibilities for the format of the eBook file that will be stored in the database and then displayed in the app:

- ePub: the official standard of the International Digital Publishing Forum (IDPF) since 2007 and the most used globally. It is XML-based and the file is an unencrypted zipped archive with interrelated resources. The ePub container usually contains:
 - Many content documents, usually based on HTML5
 - One navigation document
 - One .opf package document listing all publication resources
 - Some style sheets usually CSS
 - PLS Documents
 - media overlay documents
- customised text file with special mark-ups to indicate for example the beginning of new chapters. These will allow stopping the page at the end of a chapter and displaying the title of the chapter on the next page in a larger font and different colour.

I conducted some experiments paginating an ePub file into a phone, this was made possible by reading the navigation document and displaying the content documents one at the time in an android WebView. However, this kind of implementation would limit significantly the amount of customisation the user would be able to do at the layout of the page as for each of these changes a JavaScript code injection would be required slowing the navigation down and requiring an incredible amount of code.

Therefore I decided to use a customised text file which would allow users to have a greater control over the layout of the page, limiting however their possibility to directly import books.

File Security Methods

As part of the data collected is personal I will install a certain level of security on my system. Firstly each user when registering will have to specify a certain password following these principles:

- Passwords must be between 6 and 20 characters long
- Passwords will not be displayed whenever the user is typing them
- Files with password will be encrypted

Password will be encrypted using a one way strong hash function. To check passwords when the user tries to login, the plain password is hashed and then compared with the encrypted password in the database, if they are the same the access is granted. This process is executed in the backend server for security purpose.

DESIGN

UI – User Interface Design

The app is aimed at individuals who are competent with smartphones and have sufficient experience to use them on a daily basis. However, the individuals the program is aimed for are not experts. As a result, the graphical user interface (GUI) will reflect this and offer an easy to navigate and use face, with little confusion as to what it does.

Register activity



Figure 1 – Register activity design

Figure 1 shows the design of the register activity where the user will be directed if he says he doesn't have an account.

The user will be required to insert an email, password (twice) and genres (single choice from a dropdown).

Default choice for the language will be English.

A link will allow a user that is already registered to move to the login activity.

Login activity



Figure 2 – Login activity design

Figure 2 shows the design of the Login Activity which is the activity the program will open by default when the app is started. Before creating the view the program will verify if it has active user token (a user had previously logged in on this device) and in case will pass directly to the home activity for the found user.

The activity will show the app name with a related background and on the bottom edit boxes where the user can insert his data (email and password) in order to log in. Moreover, a link on the text “Register now” will enable the user to move to the register activity to create an account.

When the user logs in, the app will synchronise all the book he has downloaded on other devices in the current device.

Home activity

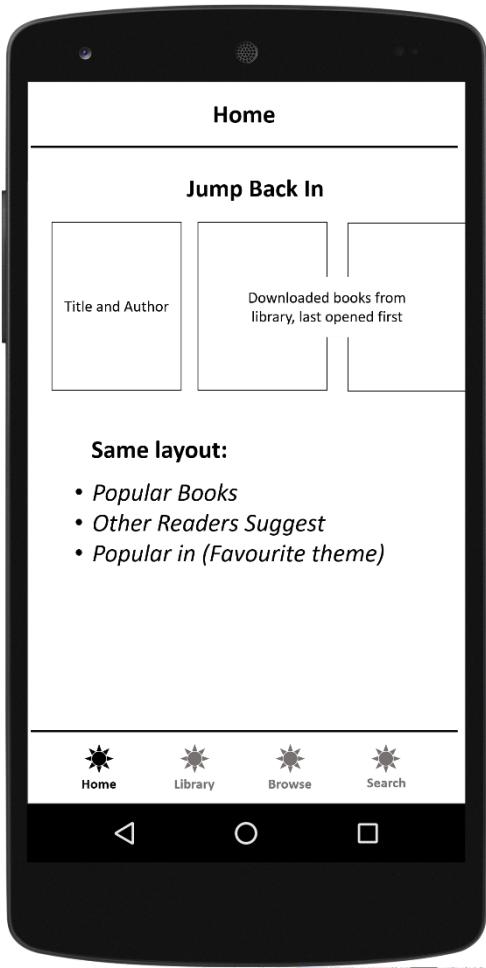


Figure 3 – Home activity design

Figure 3 shows the design of the Home Activity where the user will be directed once he is logged in.

The bottom bar allows to navigate between all the different sections of the library: Home, Library, Browse and Search.

The activity will display many horizontal lists displaying the cover of each book and allowing horizontal scroll. Some of the list can be expanded leading to a grid activity with the same list.

The lists in the Home activity will be:

- Jump Back In: downloaded books from library, last opened first
- Popular Books: most read books by other users
- Other Readers Suggest: books with best reviews from other users
- Popular In (Favourite Theme): best rated books in the user's favourite theme

If a book is selected from the Jump Back In list the ReadingActivity of that book is opened, if a book is selected from another list the correspondent BookActivity is opened.

Library activity



Figure 4 – Library activity design

Figure 4 shows the design of the Library activity which contains the grid of all the books the user read on the device (which will therefore be downloaded) ordered initially by the date when he last read it.

The user can change the order with a tap on the icon in the top-right corner. Figure 5 shows the dialog that will be opened when the user selects this option.

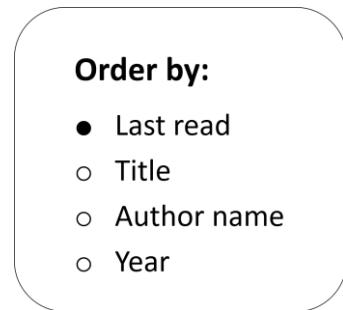


Figure 5 – Order selection dialog

Books in this activity and throughout the whole app are represented by a coloured rectangle with their title and author.

In order to make the design more various the background colour of the book will vary based on the bookid of the book. In the icon for books that the user is reading also the progress will be shown.

Browse activity

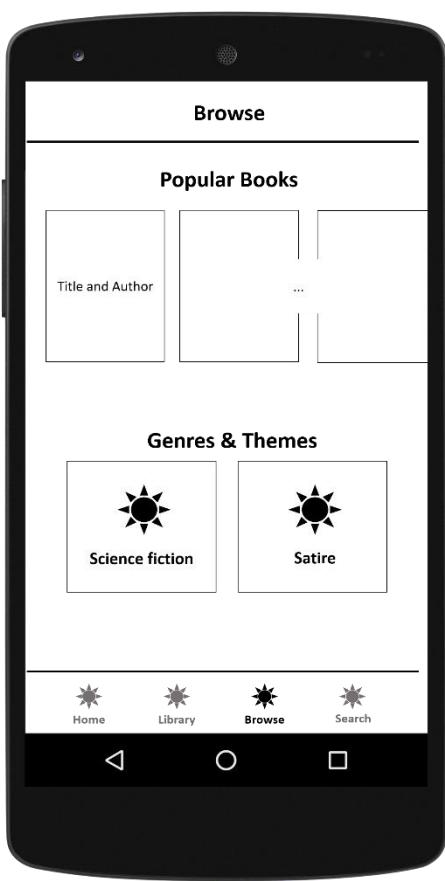


Figure 6 – Browse activity design

Figure 6 shows the design of the Browse activity from which the user can navigate between the popular books and the genres. If the user taps on a book in the popular list for him the book activity for that specific book will be opened. Instead, if the user selects one of the genres the Genre activity of the correct theme will be opened.

All genres are:

"Adventure", "Romance", "Science Fiction", "Fantasy", "Mystery", "Gothic Novel", "Tragedy", "Historical Fiction", "Bildungsroman", "Drama", "Comedy", "Politics and Economics", "Modernism", "Novel of Manners".

Their correspondent icons are:



Genre activity

Figure 7 shows the design of the Genre activity. In this activity the books of the genre the user selected are displayed in a grid view. In the top left corner a backward navigation button let the user go back to the browse activity. Initially rating, the order of the books can be changed using the button on the top right corner. Figure 8 shows the dialog that will be opened when the user selects this option.



Figure 8 – Order selection dialog



Figure 7 – Genre activity design

Search activity



Figure 9 – Search activity design

Figure 9 shows the design of the Search activity from which the user can search for a particular book or author by their name. The user will be able to input text in the textbox at the top of the screen. Once he types something, the books with their title containing the input and authors with a name that contains the input will be displayed.

If the user taps on one of the books a corresponding book activity will be opened, if the user selects one of the authors an author activity will be displayed.

Book activity



Figure 10 – Book activity design

Figure 10 shows the design of the Book activity which contains details of the book the user has selected. On the top are displayed general information such as title, author and year of publication of the book, which are followed by stars showing the average rating of the other users (approximated to the nearest 0.5). This is followed by the genre and the description of the book and finally a list of the most recent reviews of the book.

The 'Start reading' button on the bottom of the screen leads to the reading activity of the book. If the user taps on the Author name he will be directed to the correct Author activity. The 'X' on the top right of the screen will let the user return to his previous activity.

The "Start reading" button will be changed to "Continue reading" if the user has already started reading the book.

Author activity

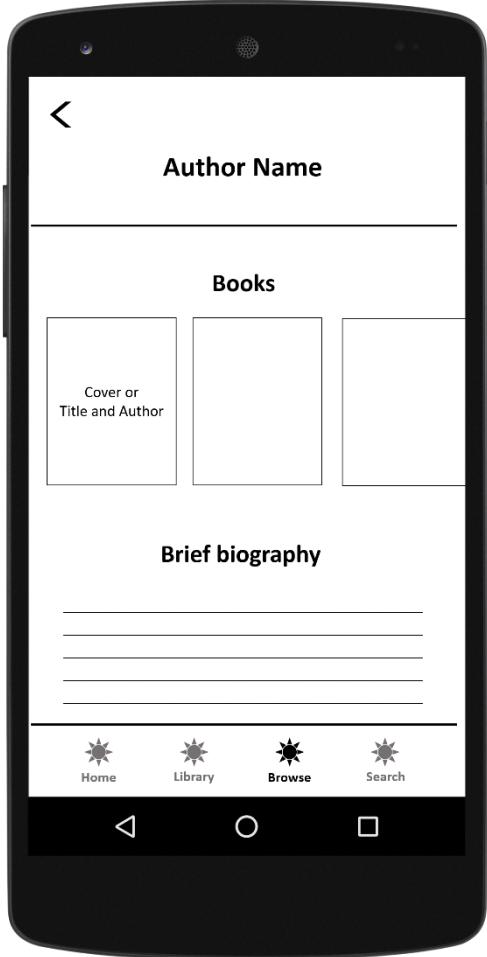


Figure 11 – Author activity design

Figure 11 shows the design of the Author activity which display the main information about a certain writer. Author name, books and a brief biography are shown in the page.

Tapping on one of the books in the list the correct book activity will be opened. With the small ‘<’ button on the top left corner the user can navigate back to the previous activity.

Reading activity

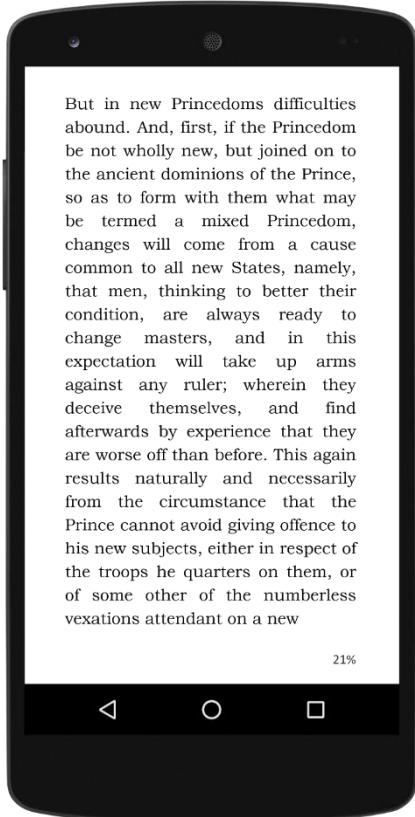


Figure 1 – Normal design

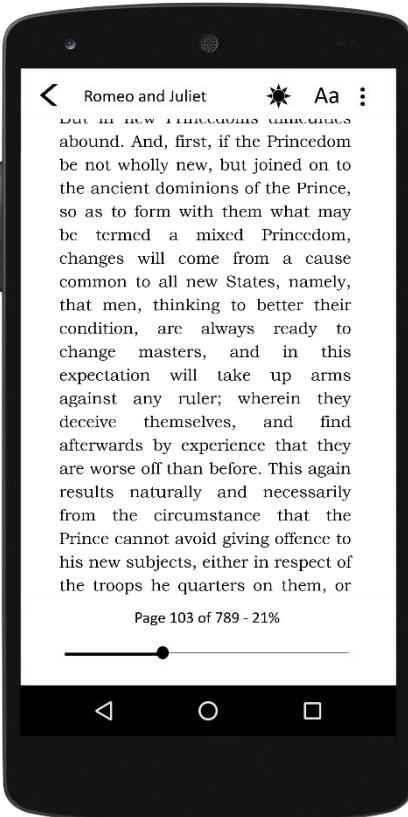


Figure 14 – Control design

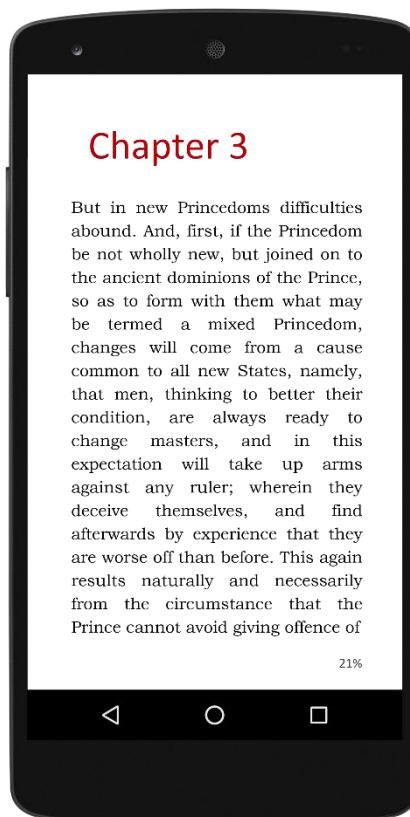


Figure 15 – Chapter design

These are the design for the reading activity where the user will actually be able to read eBooks.

Figure 13 shows the normal design the user uses to read. The text fits nicely in the page leaving some margin on left, top and right and fills the page with text until the page is about to overflow. On the bottom right the page the activity displays the percentage progress in the book. The user can move between pages just by touching the right side (next page) and left side (previous page). The user will be able to select text and choose to highlight or copy these words, add a note in the current position or translate them in another language (android built in functionality). The current position of the user in the book is always saved when the user leaves the activity and retrieved the following time he opens it.

Figure 14 shows the control design that the user can access by touching the centre of the screen from the normal design. On the screen a bottom bar will appear with which the user will be able to quickly navigate across the book. In addition, the top action bar which was originally hidden will appear on the top. On the action bar the user will see the book name and buttons to:

- Navigate back to the home activity
- Add a note in the current position

- Font control design access
- Menu window with the following options:
 - Notebook
 - Add a bookmark in the current position
 - About the book which will open the book activity of the current book
 - Help & Feedback
 - Settings

Figure 15 shows the design the activity will show the text whenever it recognises a new chapter is beginning. The app will leave blank the rest of the page after the end of a chapter and then begin the following page with the chapter title in a larger font at the top.

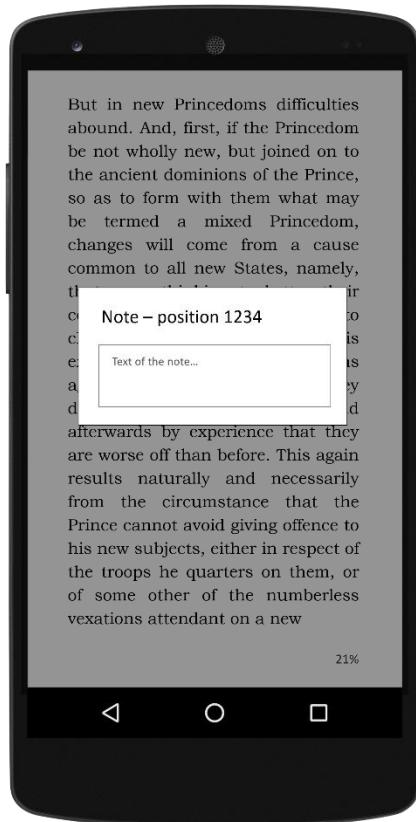


Figure 16 – New Note design

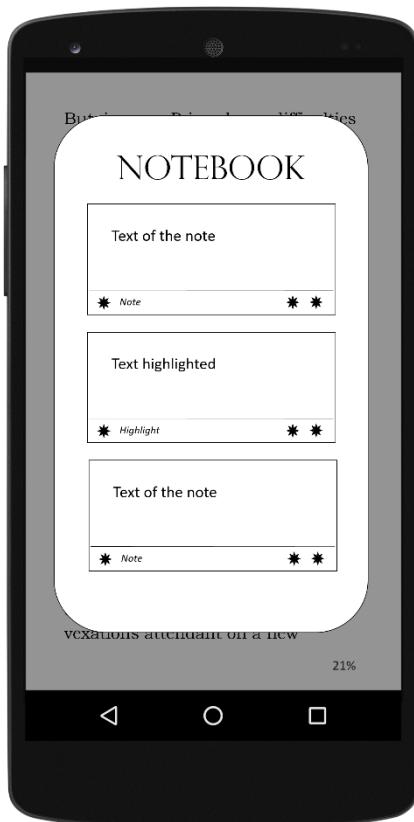


Figure 17 – Notebook design

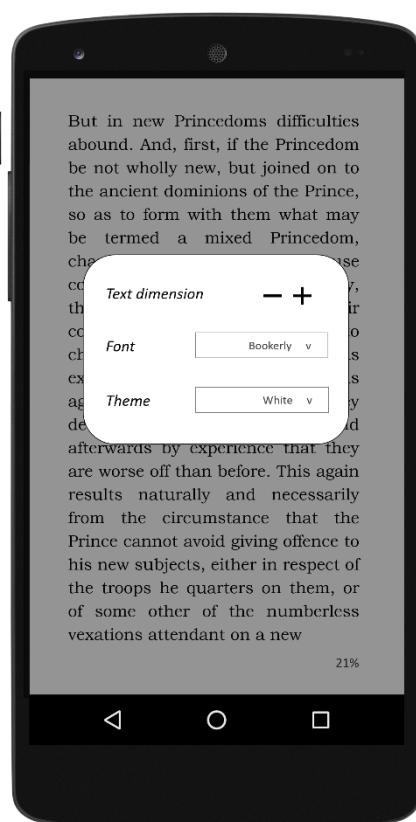


Figure 18 – Font Control design

Figure 16 shows the dialog that is opened whenever the user wants to add a new note (either from text or action bar). As the user touches the save button with the note text box not empty the app stores the note text along with its position.

Figure 17 shows the notebook design that will be opened when the user chooses this option from the menu. This has a list of all the notes and highlights the user made throughout the current book ordered by position. The item layout will have different colours for notes and highlights. If the user

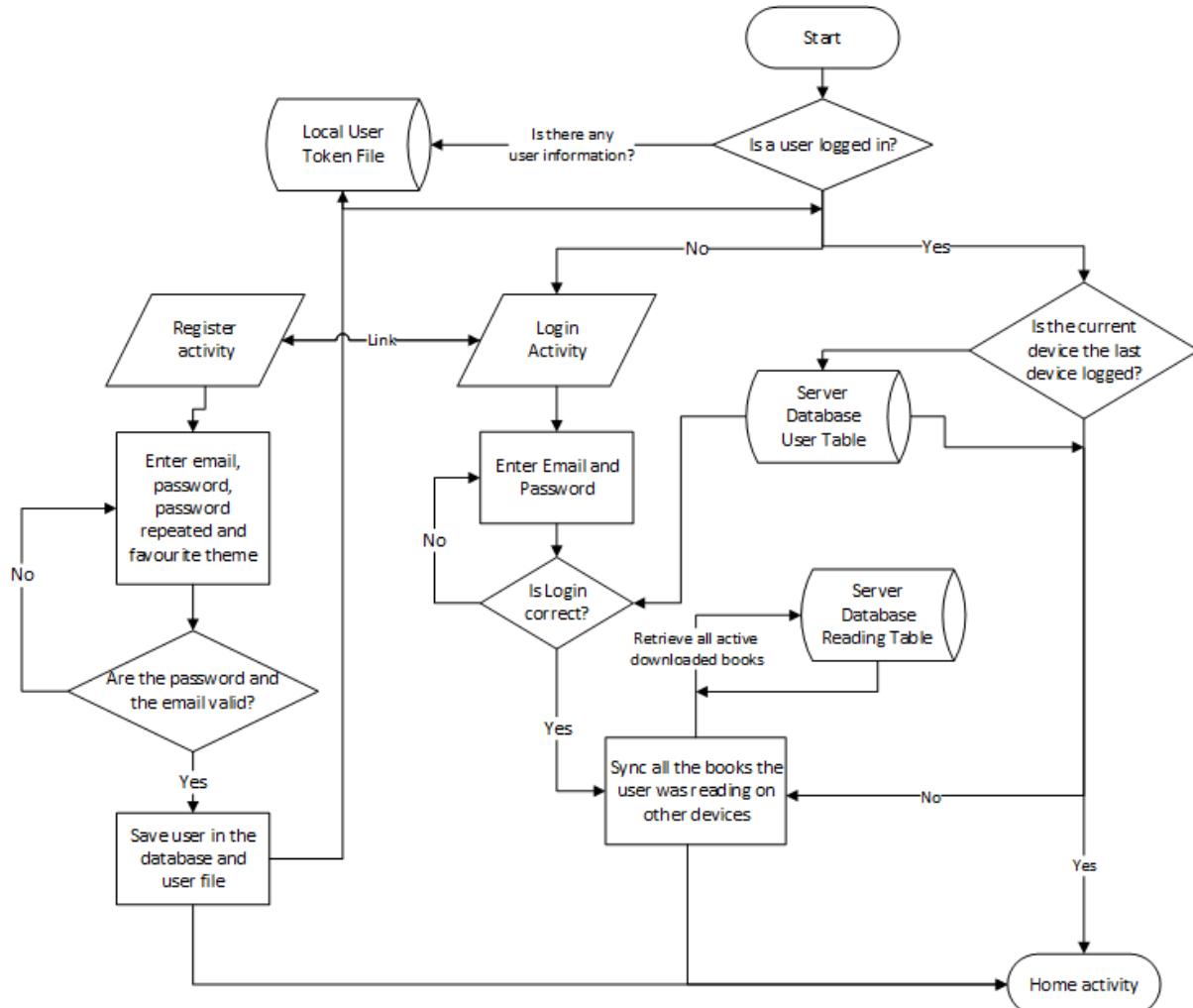
taps on the note the activity will move to the position of that note. Moreover, each item also has a button to copy the content and one to delete the note.

Figure 18 shows the font control where the user can modify the text dimension, the font (between a list) and the theme (white, black and sepia). Once the user makes a change the layout of the activity will refresh and the preferences will be saved, therefore once the user will open a reading activity for any book on the same device it will reload the same font preferences.

After the last page the user will be given a chance to give a rating to the book.

System Flow Diagrams

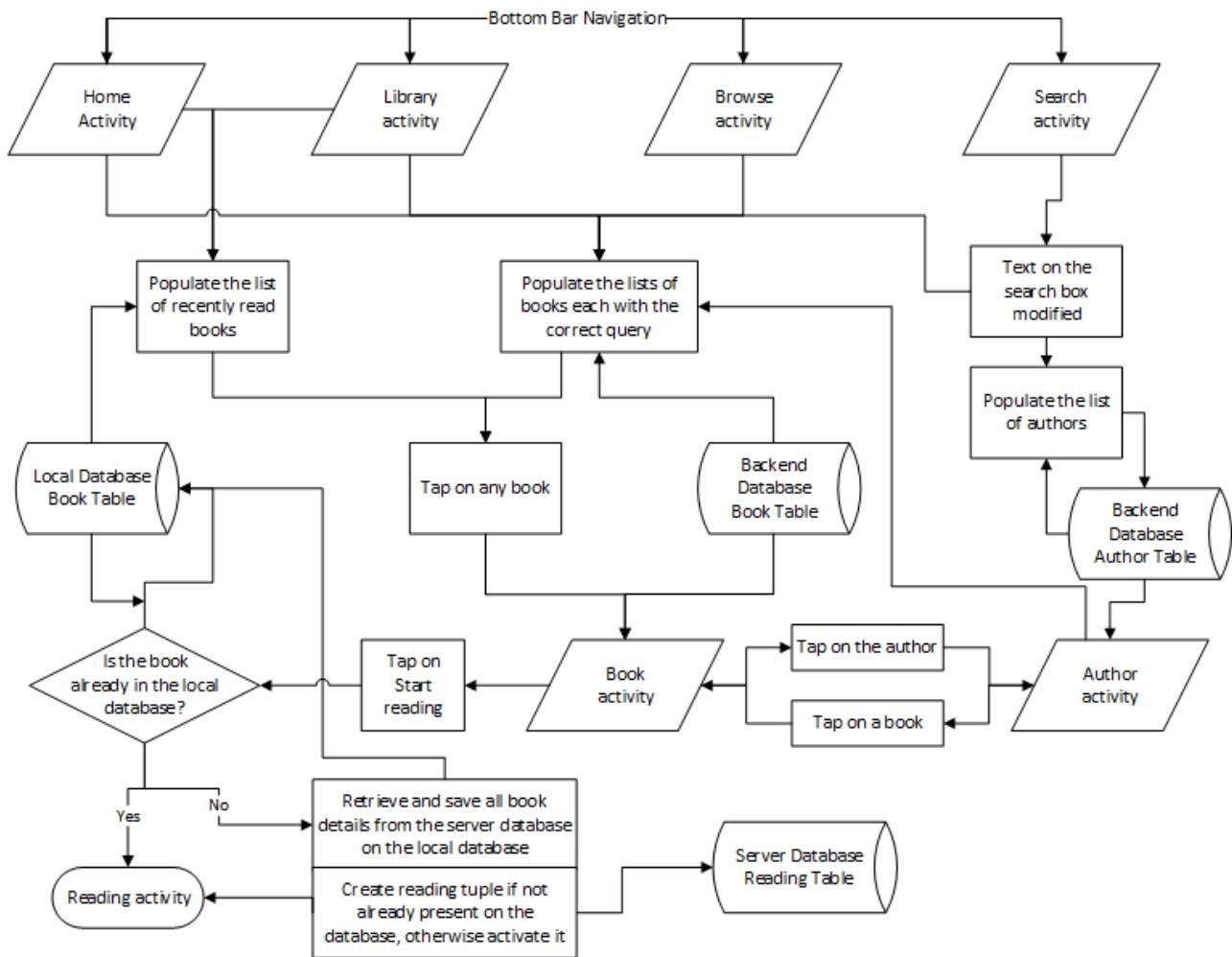
Register and Login process



The flow diagram above represents the process through which the user undergoes when he has to register, login or simply reopens the app on a device which is already logged in:

- to register a user has to enter valid email, password (twice) and favourite theme, then the details are saved both in the server user table and the local user file. The user is then directed to the HomeActivity.
- to log in the user has to enter correct email and password, these are checked in the server against the user table and if correct the details are saved in the local user file. Then the process of synchronization of all active books of the user starts. After this the user is directed to the HomeActivity.
- if the user is already logged in the app verifies in the server this is the last device used. If it is the user is directed straight away to the HomeActivity, otherwise first the synchronisation is executed and then the user is directed.

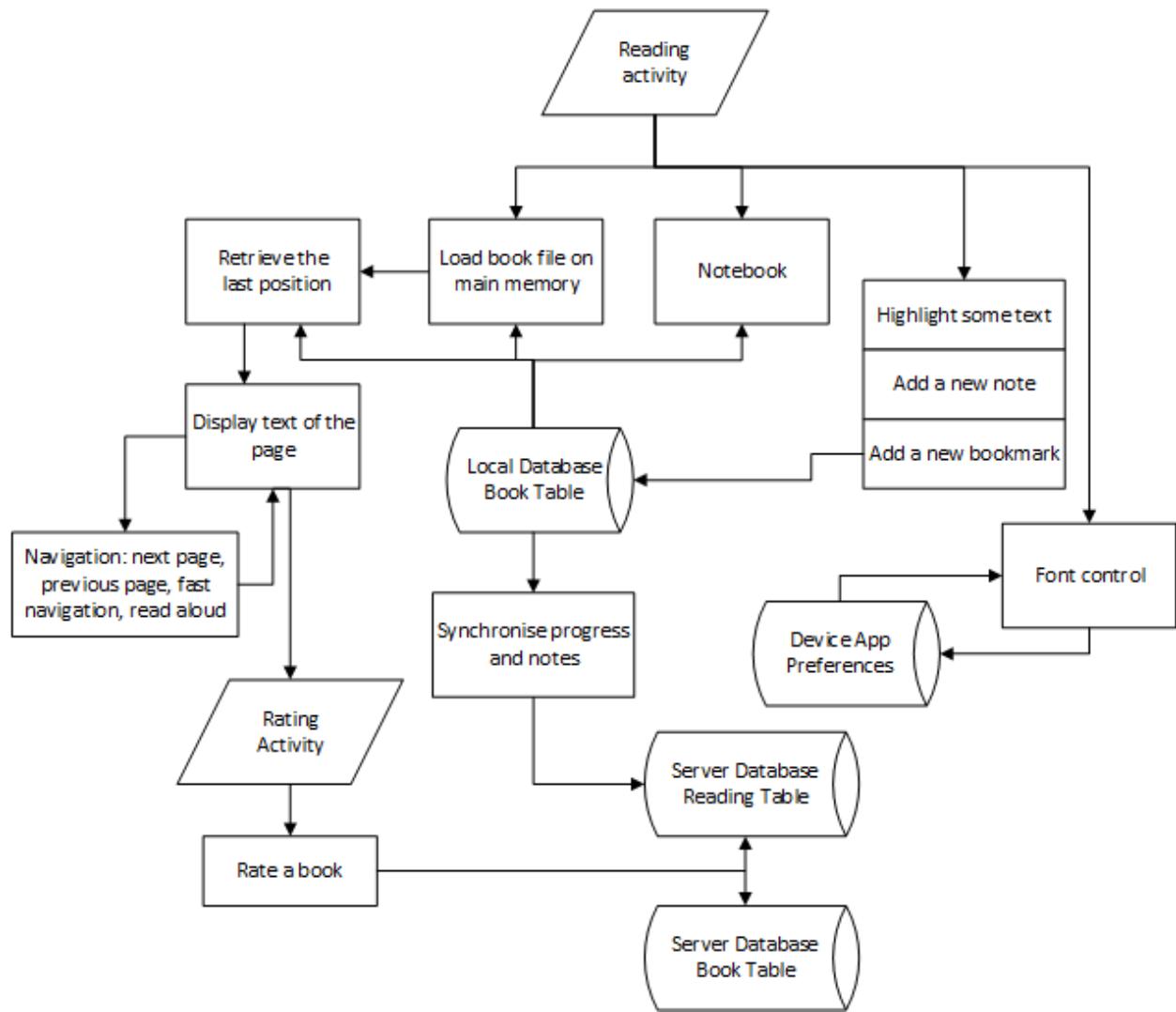
Library



The flow diagram above represents the process of navigation in library. The library can be divided in two main parts:

- the activities showing lists of books (Home, Library, Browse, Search, Genre, Author): these retrieve lists of books from the local database (in the case of the recently read books of Home and Library) or the server database (all the other cases) and show them in their layouts.
- the BookActivity where a user is conducted when it taps on a book from the lists. It gets the details of a book from the local database, if the book is present there, or the server database otherwise. When the user taps on the button "Start Reading" if the book is not present in the local database all its details are downloaded from the server and saved and a tuple of a new reading is created in the Reading table (or activated if already present).

Reading



The diagram above represents the process of the different functions that can be used in the ReadingActivity. The file of the book and progress are found in the local database. Progress, notes, highlights and bookmarks when changed are saved in the local database and then synchronised in the server reading table. Instead the preferences of the user on the font are saved in the device app preferences.

After the last page of the book the RatingActivity is opened and if the user rates a book, the server database book and reading tables are updated.

Classes Inheritance

The following table shows the inheritance between the classes. I decided to show them as a table rather than a classic inheritance diagram for a better clarity.

Subclass name	Inheritance
AuthorActivity	
BookActivity	
BrowseActivity	
GenresActivity	
Home Activity	
LibraryActivity	
SearchActivity	<i>extend AppCompatActivity</i>
LoginActivity	
RegistrationActivity	
HelpAndFeedbackActivity	
SuggestionActivity	
RatingActivity	
ReadingActivity	
ExpandableHeightGridView	<i>extends GridView</i>
ExpandableHeightListView	<i>extends ListView</i>
BookTableGridAdapter	
GenresTableGridAdapter	
SearchAuthorListAdapter	<i>extend BaseAdapter</i>
SearchBookListAdapter	
NoteListAdapter	
SquareLayout	<i>extends RelativeLayout</i>
BooksListOpenHelper	<i>extends SQLiteOpenHelper</i>

LoadAuthors	
LoadBooks	
GetBookDetails	
GetBookFile	
DeactivateBook	
GetAuthorDetails	
Login	<i>extend AsyncTask<String, String, String></i>
CheckDevice	
Synchronization	
RegisterUser	
UpdateNotes	
AddRating	
UpdateProgress	
SendSuggestion	
BookHorizGridAdapter	<i>extends RecyclerViewAdapter</i>
BookTableGridCursorAdapter	<i>extends CursorAdapter</i>
Note	<i>implements Comparable</i>
TextSelectionCallback	<i>Implements ActionMode.Callback</i>

Classes description

The following table explains briefly the goal that each class will have in the program.

Class name	Inheritance
AuthorActivity	<i>each of the following classes will render the layout of the corresponding activity, populate lists and details by making queries to the databases and control the user interaction</i>
BookActivity	
BrowseActivity	
GenresActivity	
Home Activity	
LibraryActivity	
SearchActivity	
LoginActivity	
RegistrationActivity	
HelpAndFeedbackActivity	
SuggestionActivity	
RatingActivity	
ReadingActivity	
ExpandableHeightGridView	<i>Used for the grid of themes in the Browse activity, a normal gridview cannot be used since it is inside a ScrollView (two vertically scrollable items cannot be one inside the other).</i>
ExpandableHeightListView	<i>Used for the list of books and authors in the Search activity, normal listviews cannot be used since they are inside a ScrollView (two vertically scrollable items cannot be one inside the other).</i>
BookTableGridAdapter	<i>Adapter to manage and display the grid view used for books in the Genres activity</i>
GenresTableGridAdapter	<i>Adapter used to manage and display the grid of genres in the Genres activity</i>
SearchAuthorListAdapter	<i>Adapter to manage and display the list view used for searched authors in the Search activity</i>
SearchBookListAdapter	<i>Adapter to manage and display the list view used for searched books in the Search activity</i>
NoteListAdapter	<i>Adapter to manage and display the list of notes of the Notebook</i>

SquareLayout	<i>Dynamic layout definition in order to make the items in the genres grid squared</i>
BooksListOpenHelper	<i>Private class of ... to which is therefore connected by a composition aggregation.</i>
LoadAuthors	<i>Background Async Task to load authors by making HTTP Request</i>
LoadBooks	<i>Background Async Task to load books by making HTTP Request</i>
GetBookDetails	<i>Private class of the BookActivity class to which is therefore connected by a composition aggregation. Background Async Task to get book details</i>
GetBookFile	<i>Private class of the BookActivity class to which is therefore connected by a composition aggregation. Background Async Task to get the book file, save the book in the local database and save a reading entry in the server database.</i>
DeactivateBook	<i>Private class of the BookTableGridCursorAdapter class to which is therefore connected by a composition aggregation. Background Async Task to deactivate a book.</i>
GetAuthorDetails	<i>Private class of the AuthorActivity class to which is therefore connected by a composition aggregation. Background Async Task to get complete author details</i>
Login	<i>Private class of the LoginActivity class to which is therefore connected by a composition aggregation. Background Async Task to login.</i>
CheckDevice	<i>Private class of the LoginActivity class to which is therefore connected by a composition aggregation. Background Async Task to check lastdevice of a user.</i>
Synchronization	<i>Private class of the LoginActivity class to which is therefore connected by a composition aggregation. Background Async Task to synchronize the books of a user.</i>
RegisterUser	<i>Private class of the RegisterActivity class to which is therefore connected by a composition aggregation. Background Async Task to register user.</i>
UpdateNotes	<i>Private class of the NoteList class to which is therefore connected by a composition aggregation. Background Async Task to update notes in the server database.</i>
AddRating	<i>Private class of the RatingActivity class to which is therefore connected by a composition aggregation. Background Async Task to set rating in the server database</i>

UpdateProgress	<i>Private class of the ReadingActivity class to which is therefore connected by a composition aggregation. Background Async Task to update progress in the server database.</i>
SendSuggestion	<i>Private class of the SuggestionActivity to which is therefore connected by a composition aggregation. Background Async Task to send suggestion to the server database.</i>
BookHorizGridAdapter	<i>Adapter to manage and display the horizontal scroll list view used for books in the Home activity and Browse activity</i>
BookTableGridCursorAdapter	<i>Adapter to manage and display the grid view used for books in the Library activity</i>
Note	<i>Note object</i>
TextSelectionCallback	<i>Private class of the ReadingActivity class to which is therefore connected by a composition aggregation. Sets up the menu opened when some text is selected.</i>
Author	<i>Author object</i>
Book	<i>Book object</i>
Constants	<i>Various public constants used in the program</i>
BottomBarManager	<i>Sets up the actions of the bottomBar in any activity that calls setBar</i>
User	<i>User object</i>
BookLDH	<i>BooksLDH objects are used to create and manage connection with the local database</i>
JSONParser	<i>Manages the parsing of the json responses arriving from the server database</i>
EmailValidator	<i>Checks that an email typed in the Login or Registration activity is valid before this is sent to the server.</i>
UserFileManager	<i>Singleton class to read and save the user on the data text file</i>
NoteList	<i>Class that defines a List of Notes</i>

Local Database Structure

The local database will store data of the books the user has downloaded. It will be formed by a single table Book with the structure as shown in the diagram and SQL statement.

Table creation SQL

```
CREATE TABLE Book
(BookId INT PRIMARY KEY,
Name VARCHAR(50),
AuthorId INT,
AuthorName VARCHAR(30),
Year INT,
Description TEXT,
File LONTEXT,
Genre VARCHAR(20),
Language VARCHAR(20),
Rating DECIMAL(4,3),
Reviews INT,
Progress INT,
Notes LONTEXT,
LastOpened INT,
Length INT );
```

Book	
	BookId
Name	
AuthorId	
Year	
Description	
Genre	
Language	
File	
Rating	
Reviews	
LastOpened	
Length	

Compared to the structure proposed in the analysis two fields have been added:

- *LastOpened* represents the last time in milliseconds when the user opened the book on the current device
- *Length* represents the length of the file field and is used to quickly calculate the percentage progress of each book in the library without having to load all the book (file field) in memory

SQL Queries

- *Downloaded books*

```
SELECT BookId, Name, Authorname, Progress, Length FROM Book
ORDER BY LastOpened
```

- *Book details*

```
SELECT BookId, Name, AuthorId, AuthorName, Year,
Description, Genre, Language, Rating, Reviews
FROM book WHERE bookid = $bookid
```

- *Book file*

```
SELECT BookId, Name, AuthorId, AuthorName, Year,
File, Progress, Notes
FROM Book WHERE BookId = $bookid
```

- *Update last opened*

```
UPDATE Book SET LastOpened = Now(time in ms)
FROM Book WHERE BookId = $bookid
```

- *Update progress*

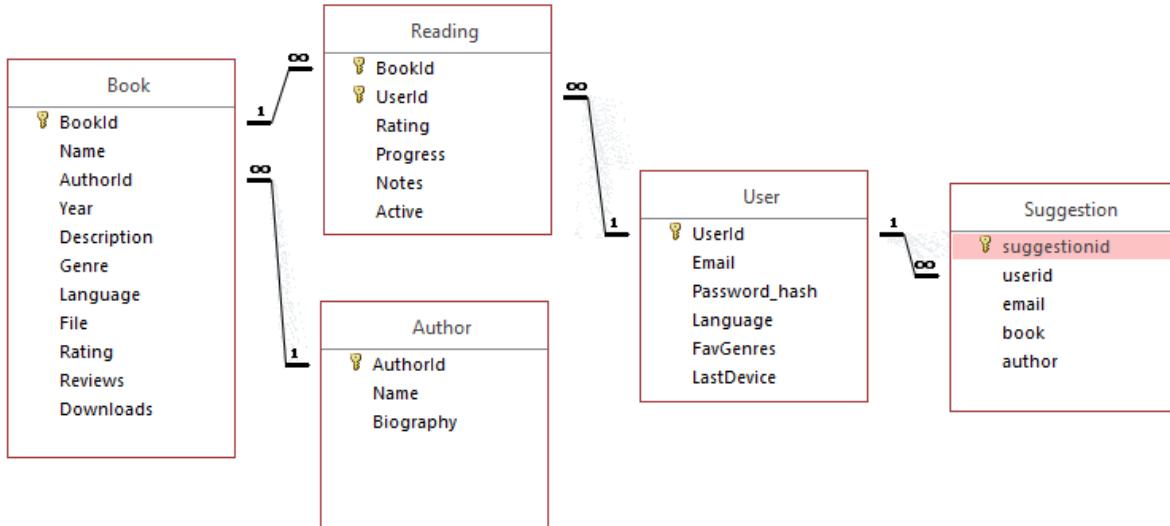
```
UPDATE Book SET Progress = $progress
FROM Book WHERE BookId = $bookid
```

- *Delete book*

```
DELETE FROM Book
WHERE BookId = $bookid
```

Server database structure

Final entity relationship diagram



In comparison with the initial entity relationship diagram in the analysis some changes have been made:

- Author table stores information about the authors of the book, for each author an AuthorId is assigned and Name and Biography are stored. No changes with the initial diagram.
- Book table stores information about each book, details of the title, author, year, genre, language and description. Compared with the initial diagram the following changes have been made:
 - The longtext field ‘file’ is added, which will therefore be stored directly inside the database (longtext fields are actually not stored directly inside the table so their long length does not affect the speed to access it).
 - The database has been slightly denormalised by adding the field authorname, this is to avoid continuous API calls to the backend for the actual name corresponding to that authorid since books are always displayed with their author’s name.
 - A second denormalisation is the insertion of the fields rating, reviews and downloads, these allow to quick order of data in the continuous calls that are made to the table (otherwise the entire reading table should be queried every book call).
- User table stores for each user email, hash of the password, preferred language, favourite genre and the number of the last device with which he opened the app. No changes with the initial diagram.

- Reading table stores the information of each book each user reads (identified by userid and bookid) regarding rating, progress and notes:
 - The progress and notes fields allow to sync the current position and the notes of the user in each book so that they can be retrieved when the user continue reading the book on another device.
 - The active field is set to 1 if the user has the book currently in its library, 2 if he removed it
- Suggestion table added in the new version of the diagram. In the app the users will be able to suggest the insertion of books that are not available yet, these suggestions are saved in this table. Userid corresponds with the id of the user that requested it, email will be set to the email of the user if he wants to be notified or will be left empty otherwise, book and author are the title and author name of the book the user is proposing to add.

Tables creation SQL

Once created a MySQL instance and database (hosted at the server: db714815721.db.1and1.com), I connected to it using the phpMyAdmin and run the following commands to create the tables:

Author table

```
CREATE TABLE author
(authorid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(30),
biography TEXT);
```

Book table

```
CREATE TABLE book
(bookid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(50),
authorid INT,
authorname VARCHAR(30),
year INT,
description TEXT,
file LONGTEXT,
genre VARCHAR(20),
language VARCHAR(20),
rating DECIMAL(4,3),
```

```
reviews INT,
FOREIGN KEY (authorid) REFERENCES author ( authorid ));
```

User table

```
CREATE TABLE user
(userid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
email VARCHAR(50),
password_hash VARCHAR(20),
language VARCHAR(20),
favgenres VARCHAR(60));
```

Reading table

```
CREATE TABLE reading (
bookid INT,
userid INT,
startdate Date,
rating INT,
PRIMARY KEY (bookid, userid),
FOREIGN KEY (bookid) REFERENCES book (bookid),
FOREIGN KEY (userid) REFERENCES user (userid));
```

Suggestion table

```
CREATE TABLE suggestion (
suggestionid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
userid INT,
email VARCHAR(255),
book VARCHAR(255),
author VARCHAR(255),
FOREIGN KEY (userid) REFERENCES user(userid));
```

Books retrieval SQL query

The following are the queries whose details will be passed from the android app to the PHP webserver, the PHP server will run it on the MySQL database and return the results to the app.

Home activity

- *Popular books*

```
SELECT bookid, name, authorname FROM book
WHERE file IS NOT NULL
ORDER BY downloads*(RAND()+2) DESC LIMIT 30
```

- *Best reviewed books*¹

```
SELECT bookid, name, authorname FROM book
WHERE file IS NOT NULL
ORDER BY rating*LN(reviews+1)*(RAND()+2) DESC LIMIT 30
```

- *Books of favourite genre*

```
SELECT bookid, name, authorname FROM book
WHERE genre = '$favourite_genre_user'
AND file IS NOT NULL
ORDER BY rating*LN(reviews+1)*(RAND()+2) DESC LIMIT 20
```

Genres activity

- *Books of genre selected*

```
SELECT bookid, name, authorname FROM book
WHERE genre = '$genre_selected'
AND file IS NOT NULL
ORDER BY rating*LN(reviews+1)*(RAND()+2) DESC LIMIT 100 or
ORDER BY downloads*(RAND()+2) DESC LIMIT 100 or
ORDER BY name LIMIT 100 or
```

¹ When ordering the books by rating, I thought I should also consider the number of reviews (a book with 10 reviews with average 4.5 should be listed before one with just one of 5). Therefore I will use the formula rating * ln(reviews +1), in order to give more importance to the rating but still consider the reviews, the +1 is to avoid problems with the logarithm of 0.

```
ORDER BY authorname LIMIT 100
or
ORDER BY year DESC LIMIT 100
```

Search activity

- *Books searched (containing certain text in title, the query is called every time the text in the search text box changes)*

```
SELECT bookid, name, authorname FROM book
WHERE name LIKE '%$text_searched%'
AND file IS NOT NULL
ORDER BY rating*LN(reviews+1) DESC LIMIT 20
```

Book activity

- *Book details*

```
SELECT bookid, name, authorid, authorname, year,
description, genre, language, rating, reviews
FROM book WHERE bookid = $bookid
```

- *Book file*

```
SELECT file FROM book WHERE bookid = $bookid
```

Author activity

- *Author books*

```
SELECT bookid, name, authorname FROM book
WHERE authorid = $authorid
AND file IS NOT NULL
ORDER BY rating*LN(reviews+1)*(RAND()+2) DESC LIMIT 20
```

Authors retrieval SQL query

Author activity

- *Author details*

```
SELECT authorid, name, biography
FROM author WHERE authorid = $authorid
```

Search activity

- *Authors searched*

```
SELECT authorid, name FROM author
WHERE name LIKE '%$text_searched%'
ORDER name LIMIT 20
```

Reading update SQL query

- *Start reading a book*

```
INSERT INTO reading(bookid, userid)
VALUES('$bookid', '$userid')
```

- *Activate/deactivate reading*

```
UPDATE reading SET active = 1/2
WHERE bookid = $bookid AND userid = $userid
```

- *Update notes*

```
UPDATE reading SET notes = '$notes'
WHERE bookid = $bookid AND userid = $userid
```

- *Update progress*

```
UPDATE reading SET progress = $progress
WHERE bookid = $bookid AND userid = $userid
```

- *Update rating*

if the user had already given a rating to the current book

```
$ratingall = ($ratingall * $reviewsall + $difference) / $reviewsall;
```

```
otherwise
```

```
$ratingall = ($ratingall * $reviewsall + $rating)/($reviewsall + 1);

$reviewsall = $reviewsall + 1;

UPDATE book SET rating = $ratingall, reviews = $reviewsall
WHERE bookid = $bookid

UPDATE reading SET rating = $rating
WHERE bookid = $bookid AND userid = $userid
```

User control SQL query

Registration activity

- *User insertion*

```
INSERT INTO user(email, password_hash, language, favgenres)
VALUES('$email', '$password_hash', '$language', '$favgenres')
```

Login activity

- *Password retrieval (on the webserver on which it will be compared to the one inserted)*

```
SELECT userid, password_hash, language, favgenres
FROM user WHERE email = '$email'
```

- *Update last device*

```
UPDATE user SET lastdevice = $lastdevice
WHERE userid = $userid
```

- *Synchronise all the books*

```
SELECT book.bookid, name, authorid, authorname, year,
description, file, genre, language, book.rating, reviews,
progress, notes FROM book, reading WHERE userid = $userid
AND book.bookid = reading.bookid AND active = 1
```

Webserver structure

Android does not allow a direct connection between the app and the backend database, therefore this must be achieved through a webserver² (written in PHP in my case). The app will send a Http request using a Json parser to the web location where the specific part of the server is stored, the server will query the database (MySQL) and return the results to the app.

Moreover, android does not allow internet connection with servers to be done on the main thread, therefore the connection has to be done through a class extending AsyncTask which will do the query in the background and return a value when finished. The file field is queried only when the user starts reading a book and once all the book is retrieved, all the details are then saved in the local database.

The following will be the PHP files stored on the server, each of them (apart from *db_config.php* and *db_connect.php* which interface the connection for all the others) is designed for a different type of query:

- *query_books.php*: allows the app to populates the various lists of books through queries of just bookid, name and authorname fields.
- *get_book_details.php*: queries all the fields of a particular book record (apart from the file field) when the app is populating the details of a book in the BookActivity (unless the book is already saved in the local database)
- *get_book_file.php*: the file of the book with a specified id from a specified character and for a specified length
- *query_authors.php*: allows the app to populate the list of authors in the SearchActivity through queries of just authorid and name.
- *get_author_details.php*: queries all the fields of a particular author record when the app is populating the details of a author in the AuthorActivity (apart from the list of books that are retrieved through *query_books.php*)
- *register_user.php*: creates a user with the details passed and an hash of the password it produces
- *login.php*: checks that the details are correct (by hashing the password and comparing it with the one on the database) and returning a boolean. If the details are correct it also updates the last device field.
- *check_device.php*: check if the number that the app has saved with the login details is the lastdevice i.e. if the device the user is using was the last one to access the app.

² Part of structure of the webserver and asynchronous calls and small parts of their code was taken from this article:
<http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql>

- *get_books_user.php*: queries all the books that the user had actively downloaded on other devices and return them so that they can be synced in the current device
- *add_rating.php*: inserts a rating into the database both updating fields in the reading and book table
- *create_reading.php*: adds a reading either by setting it as active if it already exist or by creating a new tuple.
- *deactivate_book.php*: deactivates a reading
- *update_notes.php*: updates the notes field of the correspondent reading
- *update_progress.php*: updates the progress field of the correspondent reading

Security

As described in the Analysis the password is hashed before being stored on the server database. The algorithm I chose to hash is crypt_blowfish. It is supported in the php password_hash function and creates automatically a salt which increases the security. The function returns a 60 character string which will then be stored in the database. To check passwords when the user tries to login, the plain password is hashed and then compared with the encrypted password in the database, if they are the same the access is granted.

Notes

In the app, notes, highlights and bookmarks will be stored in the same field with the following structure:

```
<<-->>note1<<-->>note2<<-->>note3<<-->>note4<<-->>
```

where note = pos<->text<->type which corresponds to:

	<i>Highlight</i>	<i>Note</i>	<i>Bookmark</i>
pos	position in the file of the beginning of the highlight	position in the file where the note was inserted	position in the file where the bookmark is added
text	text highlighted	note text	void
type	'H'	'N'	'B'

While the user is reading the correspondent book the notes will be stored in an ad hoc data structure NoteList. This keeps two list of Note elements: one of all the notes and one with just the highlights. Both of the lists are kept ordered. This structure will have to implement the most common methods for lists (add, get, remove, size) and two extra methods:

- update: transforms the note list in a string and updates the string in the local and server database
- highlights in interval (begin, end): which returns all the highlights that are even partially in the interval. Since this is called very frequently (every time the user changes page) and the number of highlights in a book can be quite high a binary search is necessary. Pseudo code for this function is on the side.

```
highlightsInInterval(begin, end)

*Binary search on first*
b = 0
e = last highlight
while (b<=e)
    current = (b+e)/2
    if end_of_current >= begin
        e = current - 1
    else
        b = current + 1

first = e+1

*Binary search on last*
b = 0
e = last highlight
while (b<=e)
    current = (b+e)/2
    if begin_of_current <= end
        b = current + 1
    else
        e = current-1

last = b-1

return elements in the list of
highlights between first and last
```

Books file

Since most of the books that can be found on the internet are in the ePub format while the app is using text files with special mark-ups (<<| |chapter| |>>) for the beginning of new chapters, I intend to create a Java program that converts ePub to text file adding a the mark-up in the correct places.

The program will have to:

- read the contents structure of the ePub file
- for each of the content file extract the text from html
- check at the beginning of each line for keywords such as “Chapter”, “Book” or “Part”
- if they contain it add the markup at the beginning of the line
- add each line to a text file and save it

Test plan

I then designed the test I would conduct on the app once completed. Tests are also colour coded based on their type: **Typical** **Boundary** **Erroneous**

RegistrationActivity

Test N°	Description & Expected result	Inputs
1.1	User can create an account using valid email, password and selecting a favourite genre.	Email: " email@crgs.co.uk " Password: "Password" Fav. Genre: "Romance"
1.2	User should be able to access to the login activity with a link on the bottom of the page.	Tap on the login link on the bottom of the page.
1.3	For security reasons the app should not show the password while in the edit box	Type "Pass" in the password box
1.4	User types an email not valid. The program should return an appropriate error message and do not register the user.	Emails tried: "" (empty), "bookipedia", "email@com"
1.5	User types a password with length outside 6-20. The program should return an appropriate error message and do not register the user.	Passwords tried: "Pass", "aaaaaaaaaaaaaaaaaaaa"
1.6	User types two passwords that do not match (case sensitive). The program should return an appropriate error message and do not register the user.	Passwords tried: "Password" and "password"
1.7	User does not select a favourite genre. The program should return an appropriate error message and do not register the user.	No favourite genre selected when trying to register.
1.8	User types a password with length just inside 6-20. The program should register the user.	Try password: "aaaaaaaaaaaaaaaaaaaa" (20)
1.9	User tries to register without a working internet connection. The program should return an appropriate error message.	Internet connection removed.

1.10	User tries to register with an email already used. The program should return an appropriate error message.	Email: " email@crgs.co.uk " Password: "Password1" Fav. Genre: "Adventure"
------	--	---

LoginActivity

Test N°	Description & Expected result	Inputs
2.1	If a user was previously logged in and the account was not opened on other devices in the meanwhile, the app should load directly the home activity.	Starting the program for after having already logged in
2.2	If a user was previously logged in and the account was opened on other devices in the meanwhile, the app should sync the books the user is reading and load the home activity. On the server a new lastdevice number should be saved.	Starting the program after having already logged in on that device and then logged in another adding a book.
2.3	If the user details are correct the app moves to the HomeActivity. On the server a new lastdevice number should be saved and on the device books the user started on other devices should be downloaded with correct details.	Email: email@crgs.co.uk Password: "Password"
2.4	If the user details are not correct the app does not move to the HomeActivity and return appropriate message. For security reasons the app should not show the password while in the edit box.	Email: email@crgs.co.uk Password: "Wrongp"
2.5	User should be able to access to the RegistrationActivity with a link on the bottom of the page.	Tap on the login link on the bottom of the page.
2.6	User types a password that do not match just because of case sensitivity with the one he registered with. The program should return an appropriate error message and do not log in the user.	Password tried: "password" when the correct was "Password"
2.7	User is already logged in and opens the app without a working internet connection. The program should show an appropriate message, log in the user and open the LibraryActivity.	Starting the program after having already logged in on that device without an internet connection.

2.8	User tries to log in without a working internet connection. The program should return an appropriate error message.	Starting the app with no user logged in and no internet connection.
-----	---	---

HomeActivity

Test N°	Description & Expected result	Inputs
3.1	The program should show in the “Jump Back In” list the books downloaded on the device in order of last time opened.	Open the app (already logged in) then open a book that is not first
3.2	If there are no books downloaded on the device the activity should not show the “Jump Back In” list	Register a new user and wait for the activity to be opened
3.3	The program should show in the “Popular In ...” text box the favourite theme of the user and populate the three different lists.	Log in with a user whose favourite genre is “Romance”
3.4	If the user taps on a book from “Jump Back In”, the corresponding ReadingActivity should be opened.	Tap on “A Christmas Carol” from “Jump Back In”
3.4	If the user taps on a book from any of the other three lists, the corresponding BookActivity should be opened.	Tap on “Pride and Prejudice” from “Popular Books”
3.6	If the user taps on the icons of other activities in the bottom bar the app should open the corresponding activity.	Tap on “Library”
3.7	User has no working internet connection. The program should return an appropriate error message and move to the LibraryActivity.	Internet connection removed.

LibraryActivity

Test N°	Description & Expected result	Inputs
4.1	Once opened the activity should show the grid the books downloaded on the device in order of last time opened. On each book the app should show the progress of the user in the book.	Open the activity, open a book and go to 50% of it and then go back to the first activity
4.2	If there are no books downloaded on the device the activity should show an appropriate message instead of the grid	Register a new user and open the activity
4.3	If the order button is tapped the activity should show the dialog.	Tap on the order button
4.4	If “Title” option is selected from the order dialog the books are ordered based on the title alphabetical name.	Tap on the order button and select Title
4.5	The option chosen for ordering should be saved and remembered when the user reopens the app.	Close and reopen the app after having selected Title
4.6	If the user taps on a book, the corresponding ReadingActivity should be opened.	Tap on “Ulysses”
4.7	User has no working internet connection. The activity should be opened anyway and the list displayed.	App opened without internet connection.

BrowseActivity

Test N°	Description & Expected result	Inputs
5.1	The program should show in the “Popular Books” list the books from the server in order of downloads and the list of genres in a grid view.	Open the activity
5.2	If the user taps on a book, the corresponding BookActivity should be opened.	Tap on “Pride and Prejudice” from “Popular Books”
5.3	If the user taps on a genre, the corresponding GenreActivity should be opened.	Tap on “Romance”

5.4	If the user taps on the menu on the top right, the menu should be opened and, selecting the correct choice from the menu the HelpAndFeedbackActivity should be opened.	Tap on the menu and then on "Help & Feedback"
5.5	User has no working internet connection. The program should return an appropriate error message and move to the LibraryActivity.	Open the BrowseActivity from LibraryActivity without connection.

GenreActivity

Test N°	Description & Expected result	Inputs
6.1	Once opened the activity should show the grid the books of the correct genre in order of rating.	Open the activity selecting "Romance"
6.2	If the order button is tapped the activity should show the dialog.	Tap on the order button
6.3	If "Title" option is selected from the order dialog the books are ordered based on the title alphabetical name.	Tap on the order button and select Title
6.4	The option chosen for ordering should be saved and remembered when the user reopens the app.	Close and reopen the app after having selected Title

SearchActivity

Test N°	Description & Expected result	Inputs
7.1	The program should show when the search box is empty a message to suggest the user to search something	Open the activity
7.2	If the search box contains something the app should search and display the books and authors with names containing the searched characters.	Search 'ch'
7.3	If there are no results for either books or authors, the program should remove that part from the layout	Search "Jane A"
7.4	If there are no results for book and no results for authors, the app should display an appropriate error message and propose the user to suggest a book. If the user taps on the message the app should open the SuggestionActivity.	Search "xxxx"
7.5	If the user taps on a book, the corresponding BookActivity should be opened.	Tap on "Pride and Prejudice"
7.6	If the user taps on an author, the corresponding AuthorActivity should be opened.	Tap on "Jane Austen"
7.7	User inserts a SQL query into the search box, the app should return the same error of test 7.4 and no changes should be made to the database.	Search "DROP TABLE suggestion"

BookActivity

Test N°	Description & Expected result	Inputs
8.1	The program should display the title, author name, year, rating, genre and description of the selected book	Open "Pride and Prejudice" from Home Activity
8.2	If the user taps on the "Continue reading" button, the corresponding ReadingActivity should be opened.	Tap on "Continue reading" from "Emma"

8.3	If the book is not present on the local database, as soon as the user taps on “Start reading” button, the app retrieves the file field of the book and save everything on the local database before opening the ReadingActivity. The number of downloads should be incremented by 1 and a tuple in the reading table should also be created.	Tap on “Start reading” from “War and Peace”
8.4	If the user taps on the author, the corresponding AuthorActivity should be opened.	Tap on “Jane Austen” from “Emma”
8.5	If the user taps on the X icon on the top right the app should go back to the previous activity.	Tap on X icon
8.6	If the book has been downloaded before the app should retrieve the details from the local database and show “Continue reading” button even without an internet connection	Open “Emma” from Library activity without an internet connection

AuthorActivity

Test N°	Description & Expected result	Inputs
9.1	The program should display the name, list of books and biography of the selected author	Open “Jane Austen” from Search Activity
9.2	If the user taps on a book, the corresponding BookActivity should be opened.	Tap on “Pride and Prejudice”

HelpAndFeedbackActivity

Test N°	Description & Expected result	Inputs
10.1	The program should display the list of options from which the user can choose.	Open the activity from BrowseActivity
10.2	If the user taps on the “Log Out” option, the app should show a dialog to confirm the choice and then, if the user confirms log him out.	Tap on “Log Out” from the list and the “Log Out” from the dialog

10.3	If the user taps on the “Suggest a book” option, the app should open the SuggestionActivity.	Tap on “Suggest a book” from the list
10.4	If the user taps on the “Feedback” option, the system should open a menu to choose an external app to send an email.	Tap on “Feedback” from the list and then “Inbox”

SuggestionActivity

Test N°	Description & Expected result	Inputs
11.1	The program should display the two input boxes for book and author names, a checkbox for the update and a send icon.	Open the activity from BrowseActivity
11.2	If the user adds the details of a book and presses the send button the app should move to the home activity and display a success message. On the server database a correspondent tuple should be saved.	Input: “Prince”, “Machiavelli” and press the send button.
11.3	If the user unchecks the update checkbox the app should not save on the server database the user email.	Input: “The Little Prince”, uncheck the update box and press the send button.
11.4	If the user leaves the book field blank and tries to send the suggestion, the app should display an appropriate error message and not save the suggestion.	Tap on the send button with the book field blank
11.5	If the user tries to send a suggestion without internet connection, the app should display an appropriate error message.	Input: “The Little Prince” and press the send button without internet connection.

ReadingActivity

Test N°	Description & Expected result	Inputs
12.1	The first time the user opens a book the app should open the first page of the book showing the chapter layout with title. The text should fill and fit in the page.	Open “Pride and Prejudice” for the first time
12.2	The user should be able to navigate in the book by tapping on the right (next page) or left (previous page).	Tap on right of the page and then on the left.
12.3	If the user taps on the centre of the screen the activity should show the control design. Tapping again on the centre the control design should disappear.	Tap on the centre of the screen and then tap again.
12.4	If user slides on the quick navigation bar of the control design, the app should quickly move to different parts of the book. When the control design is off the activity should always show the progress in the book.	Tap on the centre of the screen, then move to the middle, then tap in the centre.
12.5	If the user taps on the font control icon the app should open the font menu.	Tap on the font control icon
12.6	If the user changes the font the app should change the layout but still fit the text nicely in the screen. If the app is closed and reopened the choice of the user should be saved.	In the font menu: increase the font, change the font to Baskerville and colour to black.
12.7	If the user taps the note icon the app should open a dialog to allow the user to type a note in the current position. If the user taps “save” the note should be saved.	Tap on the note icon, type “Test note” and tap “Save”.
12.8	If the user taps on the menu icon, the app should show all the options.	Tap on the menu icon
12.9	If the user taps on “Add bookmark” from the menu, the app should save a bookmark in the current position and display an appropriate success message.	Tap on the menu icon, tap on “Add bookmark” and then open the notebook.

12.10	If the user presses on a specific part of the text, the app should open an option menu, if the user selects “highlight” the app should highlight the text and save the highlight.	Press on the words “Netherfield Park”, tap “highlight”.
12.11	If the user taps on “Notebook” from the menu, the app should open the notebook where all the notes, bookmarks and highlights of the current book are listed. If the user presses an item in the list the activity should move to the position in the book where it was taken.	Tap on “Notebook”, then tap on the first bookmark.
12.12	If the user taps on the delete icon in a note item the note should be deleted.	Tap on the delete icon of the first bookmark.
12.13	If the user taps on “Read Aloud” from the menu, the app should start reading the book aloud from the beginning of the current page and continue reading turning pages automatically. If the user taps on stop reading the app should stop immediately.	Tap on “Read Aloud”, let the app read two pages and then tap on “Stop Reading”
12.14	If the user taps on “About this book” from the menu, the app opens a BookActivity of the current book.	Tap on “About this book” from “Pride and Prejudice” ReadingActivity
12.15	If the user logs out or opens the app on another device when he reopens a book the last position and notes should be retrieved.	Leave the “Pride and Prejudice” at 16% with text highlighted on the same page, log out, then log in and reopen the book.
12.16	If the user tries to navigate after the last page of a book he should be directed to the RatingActivity of the same book.	Get at the end of “Pride and Prejudice” and press on the right of the screen.

RatingActivity

Test N°	Description & Expected result	Inputs
13.1	The program should display a rating bar and a list of books of the same author of the current book	Go after last page of "Pride and Prejudice" from ReadingActivity
13.2	If the user sets a rating, it should be saved in the database	Set the rating bar to 5 stars
13.3	If the user taps of the ← back navigation button the app should go back to the last page of the current book	Tap on ← back navigation button
13.4	If the user taps on a book from the list, the corresponding BookActivity should be opened.	Tap on "Emma"
13.5	If the user sets a rating on a book he had already rated, the rating on the server database should be correctly updated	Set the rating bar to 1 star

Epub to txt converter

Test 14: Convert the "Pride and Prejudice" book that can be downloaded from Project Gutenberg website at this link <http://bit.ly/2FAUCRP>.

TESTING

Once I finished the implementation of the project I have executed all the tests I had planned in the design to make sure the app works as intended. The tables below show some the tests I conducted, the test table for each activity is followed by evidence of the correspondent test. Tests are also colour coded based on their type: **Typical** **Boundary** **Erroneous**

RegistrationActivity

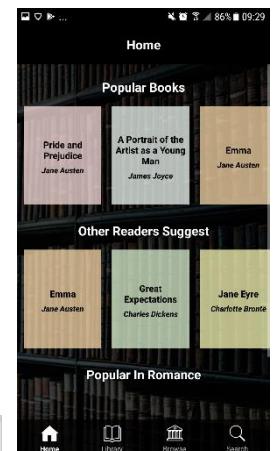
Test N°	Description & Expected result	Inputs	Result
1.1	User can create an account using valid email, password and selecting a favourite genre.	Email: " email@crgs.co.uk " Password: "Password" Fav. Genre: "Romance"	✓
1.2	User should be able to access to the login activity with a link on the bottom of the page.	Tap on the login link on the bottom of the page.	✓
1.3	For security reasons the app should not show the password while in the edit box	Type "Pass" in the password box	✓
1.4	User types an email not valid. The program should return an appropriate error message and do not register the user.	Emails tried: "" (empty), "bookipedia", "email@com"	✓
1.5	User types a password with length outside 6-20. The program should return an appropriate error message and do not register the user.	Passwords tried: "Pass", "pppppppppppppppppp"	✓
1.6	User types two passwords that do not match (case sentive). The program should return an appropriate error message and do not register the user.	Passwords tried: "Password" and "password"	✓
1.7	User does not select a favourite genre. The program should return an appropriate error message and do not register the user.	No favourite genre selected when trying to register.	✓

1.8	User types a password with length just inside 6-20. The program should register the user.	Try password: “pppppppppp pppppppppp” (20)	✓
1.9	User tries to register without a working internet connection. The program should return an appropriate error message.	Internet connection removed.	✓
1.10	User tries to register with an email already used. The program should return an appropriate error message.	Email: “ email@crgs.co.uk ” Password: “Password1” Fav. Genre: “Adventure”	✓

1.1.



Once the user taps on the register button the app open a progress dialog for less than a second and then the Home Activity is opened. On the server database the entry is created correctly.



18 | email@crgs.co.uk

\$2y\$10\$000UPReHDal4sH3/SlaYpekSaDTvzz01KL4uoliwdVu...

English

Romance

52299

1.2.



Once the user taps on the login activity link the app opens the login activity.

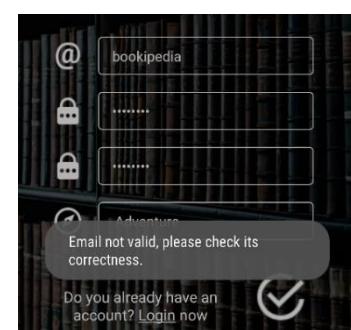
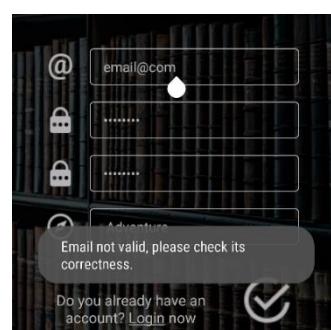


1.3.



As the user types the password, it is not shown, but only dots are used for characters.

1.4.



Once the user taps on the register button with an invalid email the app prints a toast message "Email not valid, please check its correctness." and do not register the user.

1.5.



Once the user taps on the register button with an invalid password the app prints a toast message “Password not valid, it needs to be between 6 and 20 characters.” and do not register the user.

1.6.



Once the user taps on the register button with two passwords that do not match the app prints a toast message “Passwords do not match, please check them.” and do not register the user.

1.7.



Once the user taps on the register button without selecting a genre the app prints a toast message “Please choose a favorite genre.” and do not register the user.

1.8.



Once the user taps on the register button with a password of 20 characters a, the app works normally: opens a progress dialog for less than a second and then the Home Activity is opened. On the server database the entry is created correctly.



1.9.



Once the user taps on the register button without an internet connection the app prints a toast message “Internet connection not available. Please check your connection.” and do not register the user.

1.10.



If the user taps on the register button without an internet connection the app prints a toast message “Internet connection not available. Please check your connection.” and do not register the user.

LoginActivity

Test N°	Description & Expected result	Inputs	Result
2.1	If a user was previously logged in and the account was not opened on other devices in the meanwhile, the app should load directly the home activity.	Starting the program for after having already logged in	✓
2.2	If a user was previously logged in and the account was opened on other devices in the meanwhile, the app should sync the books the user is reading and load the home activity. On the server a new lastdevice number should be saved.	Starting the program after having already logged in on that device and then logged in another adding a book.	✓
2.3	If the user details are correct the app moves to the HomeActivity. On the server a new lastdevice number should be saved and on the device books the user started on other devices should be downloaded with correct details.	Email: email@crgs.co.uk Password: "Password"	✓
2.4	If the user details are not correct the app does not move to the HomeActivity and return appropriate message. For security reasons the app should not show the password while in the edit box.	Email: email@crgs.co.uk Password: "Wrongp"	✓
2.5	User should be able to access to the RegistrationActivity with a link on the bottom of the page.	Tap on the login link on the bottom of the page.	✓
2.6	User types a password that do not match just because of case sensitivity with the one he registered with. The program should return an appropriate error message and do not log in the user.	Password tried: "password" when the correct was "Password"	✓
2.7	User is already logged in and opens the app without a working internet connection. The program should show an appropriate message, log in the user and open the LibraryActivity.	Starting the program after having already logged in on that device without an internet connection.	✓
2.8	User tries to log in without a working internet connection. The program should return an appropriate error message.	Starting the app with no user logged in and no internet connection.	✓

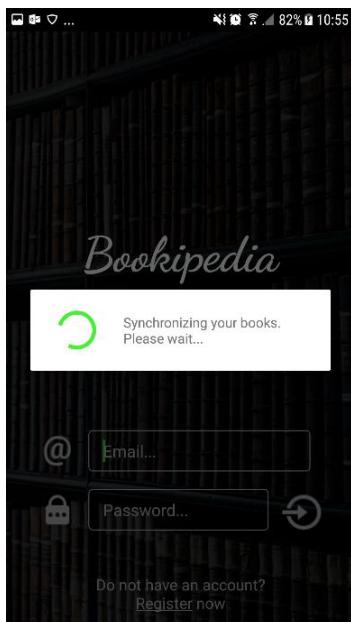
2.1.



The user has previously logged in and the account was not opened on other devices in the meanwhile.

Once the user opens the app, the home activity is loaded directly.

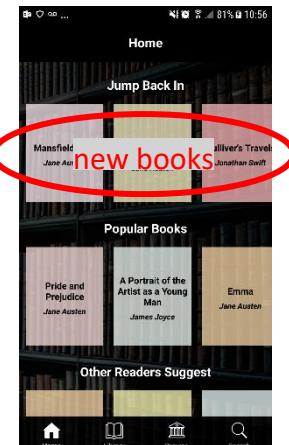
2.2.



The user has previously logged in and the account was opened on another device in the meanwhile where some books were downloaded.

The app syncs the books the user is reading (during the process the app shows a progress dialog and it takes about a second) and loads the home activity. On the server a new lastdevice number is saved.

18	email@crags.co.uk	\$2y\$10\$000U	English	Romance	62
----	-------------------	----------------	---------	---------	----



2.3.



Once the login button is clicked the app moves to the HomeActivity. On the server a new lastdevice number should be saved and on the device books the user started on other devices are synced downloaded with correct details.



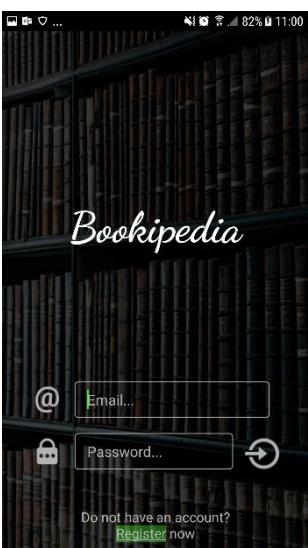
2.4.



Once the user taps on the login button with an wrong password the app prints a toast message “Email or password not correct. Please check them.” and do not log in the user.

For security reasons the app does not show the password while in the edit box

2.5.



Once the user taps on the register activity link the app opens the register activity.

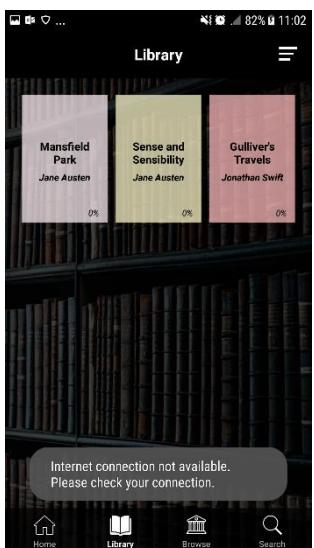


2.6.



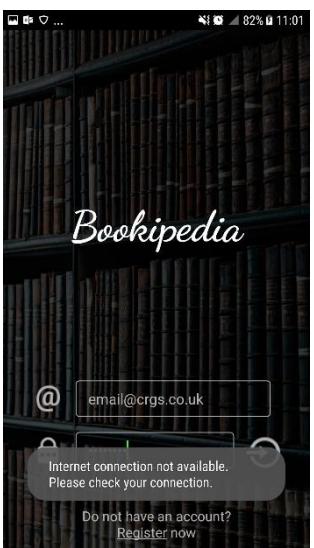
The user types a password that do not match just because of case sensitivity with the one he registered with. The app prints a toast message “Email or password not correct. Please check them.” and do not log in the user.

2.7.



User is already logged in and opens the app without a working internet connection. The app prints a toast message “Internet connection not available. Please check your connection.” and open the LibraryActivity.

2.8.

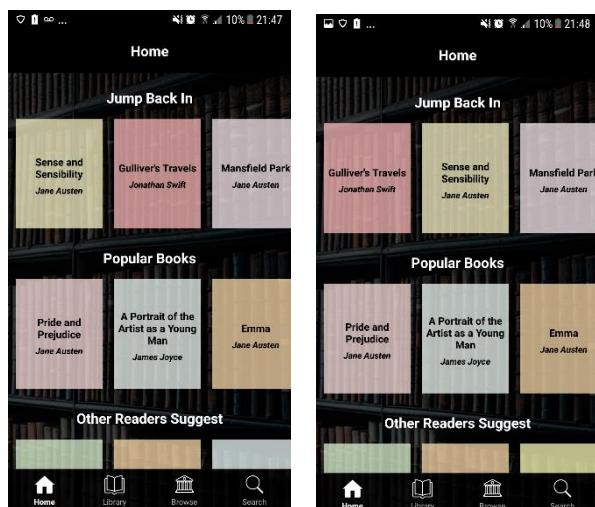


User tries to log in without a working internet connection. The app prints a toast message “Internet connection not available. Please check your connection.” and does not log in the user.

HomeActivity

Test N°	Description & Expected result	Inputs	Result
3.1	The program should show in the “Jump Back In” list the books downloaded on the device in order of last time opened.	Open the app (already logged in) then open a book that is not first	✓
3.2	If there are no books downloaded on the device the activity should not show the “Jump Back In” list	Register a new user and wait for the activity to be opened	✓
3.3	The program should show in the “Popular In ...” text box the favourite theme of the user and populate the three different lists.	Log in with a user whose favourite genre is “Romance”	✓
3.4	If the user taps on a book from “Jump Back In”, the corresponding ReadingActivity should be opened.	Tap on “A Christmas Carol” from “Jump Back In”	✓
3.4	If the user taps on a book from any of the other three lists, the corresponding BookActivity should be opened.	Tap on “Pride and Prejudice” from “Popular Books”	✓
3.6	If the user taps on the icons of other activities in the bottom bar the app should open the corresponding activity.	Tap on “Library”	✓
3.7	User has no working internet connection. The program should return an appropriate error message and move to the LibraryActivity.	Internet connection removed.	✓

3.1.



The app has just been opened and in the “Jump Back In” list appear the three books downloaded on device “Sense and Sensibility”, “Gulliver’s Travels” and “Mansfield Park” in this order. “Gulliver’s Travels” is opened and when the user goes back to the Home Activity it is first in the list.

3.2.



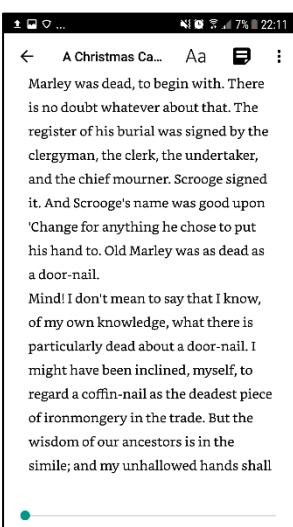
A new user is registered, so when the HomeActivity is opened there are no books downloaded on the device and therefore the activity does not show the “Jump Back In” list

3.3.



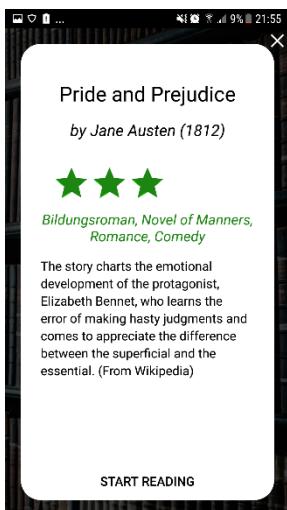
The app correctly shows “Popular In Romance”, and populates the three lists respectively with books with high downloads, high rating and review and romance books.

3.4.



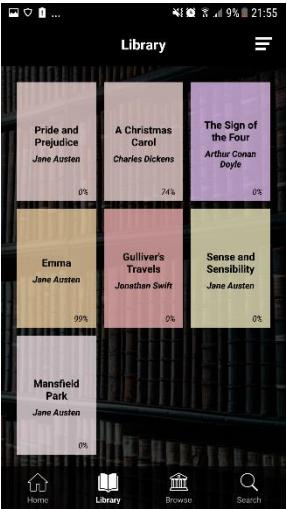
Once the user taps on “A Christmas Carol” from “Jump Back In”, the corresponding ReadingActivity is opened.

3.5.



If the user taps on “Pride and Prejudice” from “Popular Books”, the corresponding BookActivity is opened.

3.6.



The user taps on the icons of the Library activity in the bottom bar, the app opens the Library activity.

3.7.



Remove internet connection and open the home activity. The app prints an error toast message “Internet connection not available. Please check your connection.” and move to the LibraryActivity.

LibraryActivity

Test N°	Description & Expected result	Inputs	Result
4.1	Once opened the activity should show the grid the books downloaded on the device in order of last time opened. On each book the app should show the progress of the user in the book.	Open the activity, open a book and go to 50% of it and then go back to the first activity	✓
4.2	If there are no books downloaded on the device the activity should show an appropriate message instead of the grid	Register a new user and open the activity	✓
4.3	If the order button is tapped the activity should show the dialog.	Tap on the order button	✓
4.4	If “Title” option is selected from the order dialog the books are ordered based on the title alphabetical name.	Tap on the order button and select Title	✓
4.5	The option chosen for ordering should be saved and remembered when the user reopens the app.	Close and reopen the app after having selected Title	✓
4.6	If the user taps on a book, the corresponding ReadingActivity should be opened.	Tap on “Ulysses”	✓
4.7	User has no working internet connection. The activity should be opened anyway and the list displayed.	App opened without internet connection.	✓



Once opened the activity shows the grid the books downloaded with a percentage on the bottom right of the box (the progress in the book). The book ‘Ulysses’ is first.

‘A Christmas Carol’ is opened and the scrolled until the progress is 50%. Then, going back to the first activity, this book is showed first with a correct 50 % progress.

4.2.



A new user is registered, so when the LibraryActivity is opened there are no books downloaded on the device and therefore the activity show the message “No book downloaded yet. Start reading a book to download it and it will appear in this section.”

4.3.



The order button on the top-right is tapped the activity shows the order dialog with title “Sort by” and options: “Last opened”, “Title”, “Author name” and “Year”.

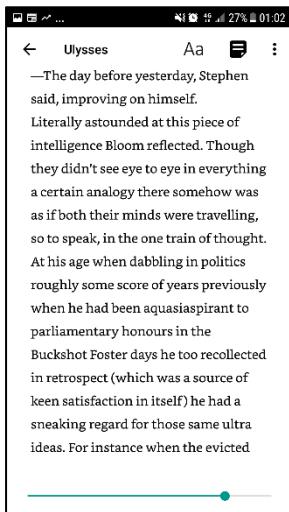
4.4.



The “Title” option is selected from the order dialog the books are correctly ordered alphabetically based on the title.

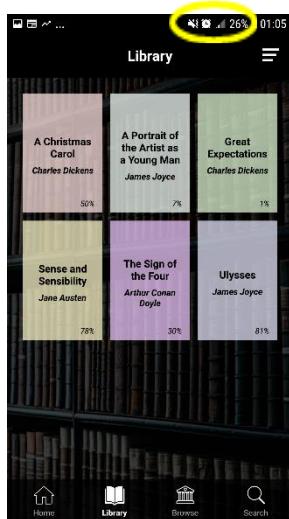
4.5. If the app is closed and reopened the choice selected is still “Title” and the app correctly shows books ordered alphabetically.

4.6.



The book 'Ulysses' is selected and the corresponding ReadingActivity is be opened.

4.7.

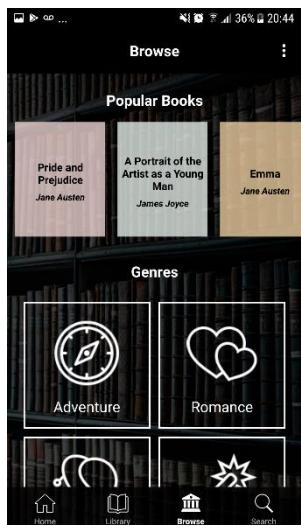


Without internet connection the activity is be opened anyway and the list displayed.

BrowseActivity

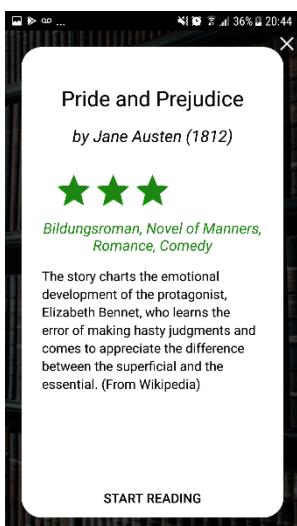
Test N°	Description & Expected result	Inputs	Result
5.1	The program should show in the “Popular Books” list the books from the server in order of downloads and the list of genres in a grid view.	Open the activity	✓
5.2	If the user taps on a book, the corresponding BookActivity should be opened.	Tap on “Pride and Prejudice” from “Popular Books”	✓
5.3	If the user taps on a genre, the corresponding GenreActivity should be opened.	Tap on “Romance”	✓
5.4	If the user taps on the menu on the top right, the menu should be opened and, selecting the correct choice from the menu the HelpAndFeedbackActivity should be opened.	Tap on the menu and then on “Help & Feedback”	✓
5.5	User has no working internet connection. The program should return an appropriate error message and move to the LibraryActivity.	Open the BrowseActivity from LibraryActivity without connection.	✓

5.1.



Once opened the activity shows in the “Popular Books” list the books from the server in order of downloads and the list of genres in a grid view.

5.2.



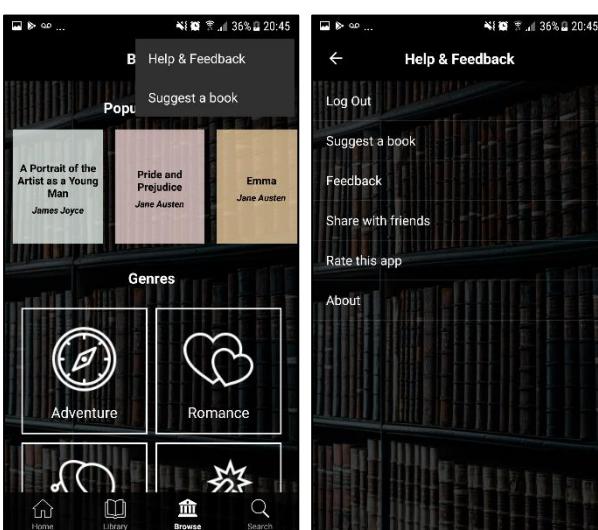
Tapping on “Pride and Prejudice” from “Popular Books” the correct BookActivity is opened.

5.3.



Tapping on “Romance”, the correct GenreActivity is opened.

5.4.



Tap on the menu on the top right, the menu is opened showing correctly “Help & Feedback” and “Suggest a book”.

From the menu the “Help & Feedback” option is selected and the HelpAndFeedbackActivity is opened.

5.5.



The BrowseActivity is opened from the LibraryActivity without connection. The app prints an error toast message “Internet connection not available. Please check your connection.” and move to the LibraryActivity.

GenreActivity

Test N°	Description & Expected result	Inputs	Result
6.1	Once opened the activity should show the grid the books of the correct genre in order of rating.	Open the activity selecting "Romance"	✓
6.2	If the order button is tapped the activity should show the dialog.	Tap on the order button	✓
6.3	If "Title" option is selected from the order dialog the books are ordered based on the title alphabetical name.	Tap on the order button and select Title	✓
6.4	The option chosen for ordering should be saved and remembered when the user reopens the app.	Close and reopen the app after having selected Title	✓

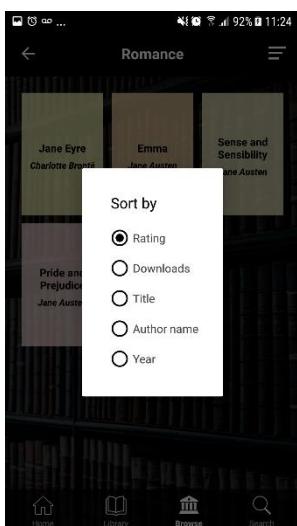
6.1.



Once opened for the first time the activity populates the grid the books of the correct genre in order of rating.

bookid	name	authorid	authorname	year	description	file	genre	language	rating	reviews
1	Mansfield Park	3	Jane Austen	1814	Mansfield Park is the third published novel by Jan...	<< chapter >>CHAPTER I About thirty years ago Mi...	Bildungsroman, Novel of Manners, Romance, Comedy	English	3.000	1
3	Pride and Prejudice	3	Jane Austen	1812	The story charts the emotional development of the ...	<< chapter >>Chapter 1 It is a truth universally...	Bildungsroman, Novel of Manners, Romance, Comedy	English	3.000	1
4	Emma	3	Jane Austen	1815	Emma, by Jane Austen, is a novel about youthful hu...	<< chapter >> VOLUME I << chapter >> CHAPTER I...	Bildungsroman, Novel of Manners, Romance, Comedy	English	3.500	2
2	Sense and Sensibility	3	Jane Austen	1811	Sense and Sensibility tells the story of the Dashw...	<< chapter >>CHAPTER 1 The family of Dashwood ha...	Bildungsroman, Novel of Manners, Romance	English	3.500	1
12	Jane Eyre	4	Charlotte Brontë	1847	Primarily of the Bildungsroman genre, Jane Eyre fo...	<< chapter >>CHAPTER I There was no possibility ...	Bildungsroman, Romance, Gothic Novel, Social Criti...	English	4.000	2

6.2.



Order button is tapped the activity shows the order dialog with options “Rating”, “Downloads”, “Title”, “Author name” and “Year”.

6.3.



The title option is selected and the books are now ordered alphabetically based on their title.

6.4. After closing and reopening the app the activity still remembers the ordering option selected

SearchActivity

Test N°	Description & Expected result	Inputs	Result
7.1	The program should show when the search box is empty a message to suggest the user to search something	Open the activity	✓
7.2	If the search box contains something the app should search and display the books and authors with names containing the searched characters.	Search 'ch'	✓
7.3	If there are no results for either books or authors, the program should remove that part from the layout	Search "Jane A"	✓
7.4	If there are no results for book and no results for authors, the app should display an appropriate error message and propose the user to suggest a book. If the user taps on the message the app should open the SuggestionActivity.	Search "xxxx"	✓
7.5	If the user taps on a book, the corresponding BookActivity should be opened.	Tap on "Pride and Prejudice"	✓
7.6	If the user taps on an author, the corresponding AuthorActivity should be opened.	Tap on "Jane Austen"	✓
7.7	User inserts a SQL query into the search box, the app should return the same error of test 7.4 and no changes should be made to the database.	Search "DROP TABLE suggestion"	✓

7.1.



Once the activity is opened it shows the message "Search Bookipedia. Find your favourite books and authors".

7.2.



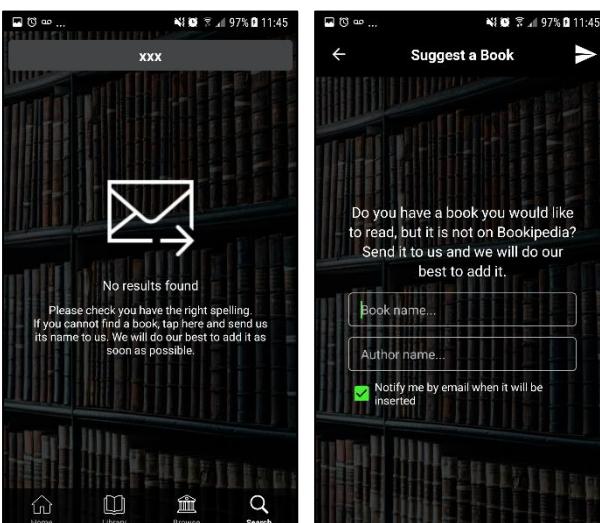
"ch" is searched in the search box and the app correctly displays the books and authors with names containing "ch".

7.3.



"Jane A" is searched and since there are no books containing those characters in the name the books layout is removed from the layout of the page and only the authors list is displayed.

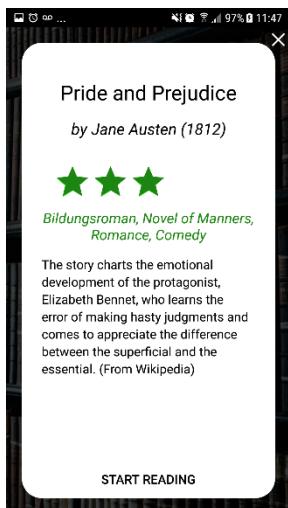
7.4.



"xxx" is searched, since there are no results for books and no results for authors, the app shows the message: "No results found. Please check you have the right spelling. If you cannot find a book, tap here and send us its name. We will do our best to add it as soon as possible".

Tapping the message the SuggestionActivity is opened.

7.5.



If the user taps on “Pride and Prejudice”, the corresponding BookActivity is opened.

7.6.



If the user taps on “Jane Austen”, the corresponding AuthorActivity is opened.

7.7.

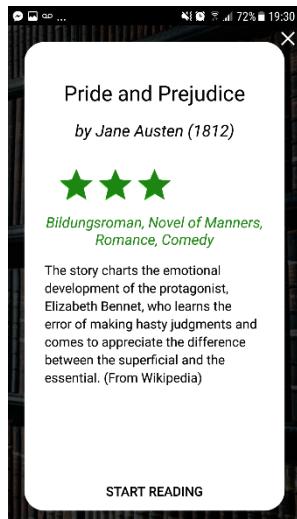


Searching “DROP TABLE suggestion” in the search box, the app shows the error message: “No results found. Please check you have the right spelling. If you cannot find a book, tap here and send us its name to us. We will do our best to add it as soon as possible” and no changes are made to the database.

BookActivity

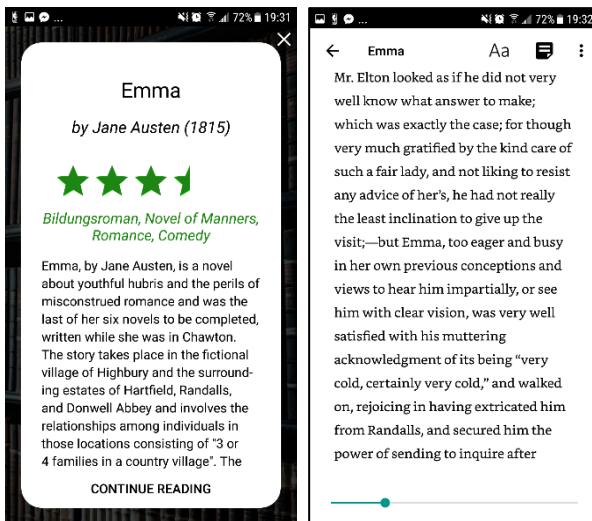
Test N°	Description & Expected result	Inputs	Result
8.1	The program should display the title, author name, year, rating, genre and description of the selected book	Open "Pride and Prejudice" from Home Activity	✓
8.2	If the user taps on the "Continue reading" button, the corresponding ReadingActivity should be opened.	Tap on "Continue reading" from "Emma"	✓
8.3	If the book is not present on the local database, as soon as the user taps on "Start reading" button, the app retrieves the file field of the book and save everything on the local database before opening the ReadingActivity. The number of downloads should be incremented by 1 and a tuple in the reading table should also be created.	Tap on "Start reading" from "War and Peace"	✓
8.4	If the user taps on the author, the corresponding AuthorActivity should be opened.	Tap on "Jane Austen" from "Emma"	✓
8.5	If the user taps on the X icon on the top right the app should go back to the previous activity.	Tap on X icon	✓
8.6	If the book has been downloaded before the app should retrieve the details from the local database and show "Continue reading" button even without an internet connection	Open "Emma" from Library activity without an internet connection	✓

8.1.



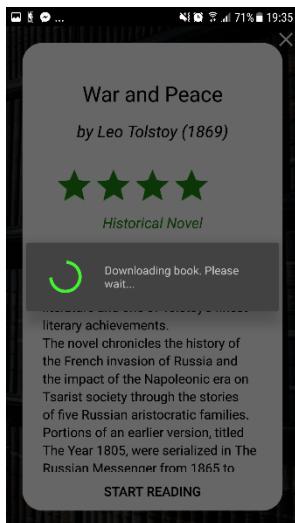
Opened "Pride and Prejudice" from Home Activity, the app displays its title, author name, year, rating, genre and description.

8.2.



Tapping on “Continue reading” from “Emma” the corresponding `ReadingActivity` is opened.

8.3.



Tapping on “Start reading” from “War and Peace”. The app shows a progress for few seconds with the message “Downloading book. Please wait...” and then opens the `ReadingActivity`. The number of downloads on the server database is incremented by 1 and a tuple in the reading table is created.

bookid	userid	rating	progress	notes	active
63	18	-1	0		1

8.4.



Tapping on “Jane Austen” from “Emma”, the corresponding `AuthorActivity` should be opened.

8.5.



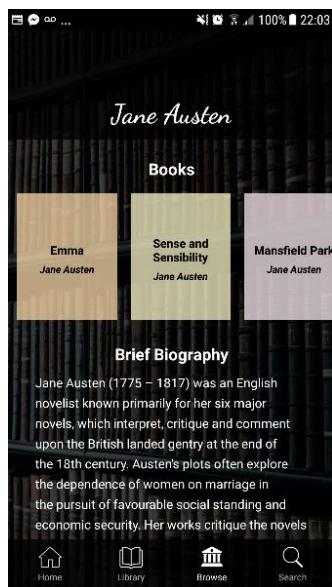
Tapping on X icon the app goes back to the previous activity (HomeActivity in this case).

- 8.6. Opening “Emma” from Library activity without an internet connection the app behave normally, details are shown in the BookActivity and the book is opened when the user taps on “Continue reading”.

AuthorActivity

Test N°	Description & Expected result	Inputs	Result
9.1	The program should display the name, list of books and biography of the selected author	Open “Jane Austen” from Search Activity	✓
9.2	If the user taps on a book, the corresponding BookActivity should be opened.	Tap on “Pride and Prejudice”	✓

9.1.



Opening “Jane Austen” from Search Activity, the app displays correctly the name, list of books and biography of the author.

9.2.

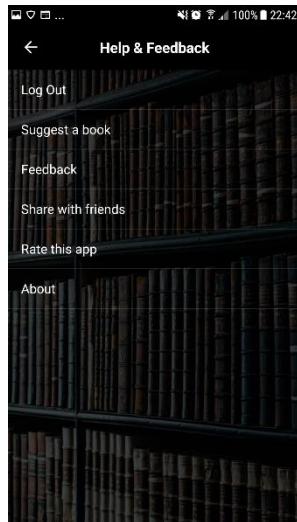


Tapping on “Pride and Prejudice”, the correct BookActivity is opened.

HelpAndFeedbackActivity

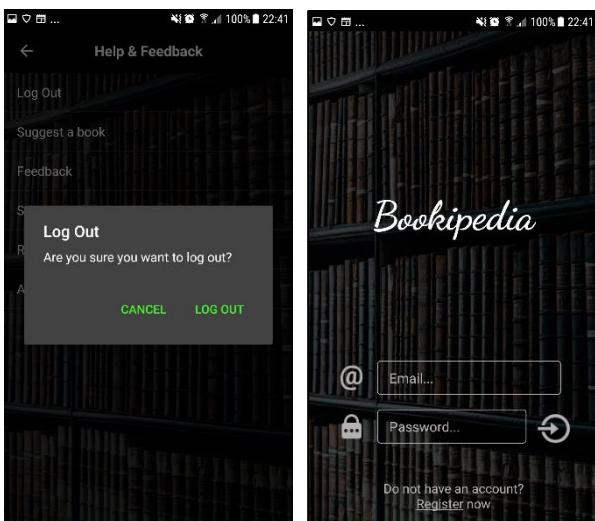
Test N°	Description & Expected result	Inputs	Result
10.1	The program should display the list of options from which the user can choose.	Open the activity from BrowseActivity	✓
10.2	If the user taps on the “Log Out” option, the app should show a dialog to confirm the choice and then, if the user confirms log him out.	Tap on “Log Out” from the list and the “Log Out” from the dialog	✓
10.3	If the user taps on the “Suggest a book” option, the app should open the SuggestionActivity.	Tap on “Suggest a book” from the list	✓
10.4	If the user taps on the “Feedback” option, the system should open a menu to choose an external app to send an email.	Tap on “Feedback” from the list and then “Inbox”	✓

10.1.



Opening the activity from BrowseActivity, the app displays the list of options from which the user can choose: "Log Out", "Suggest a book", "Feedback", "Share with friends", "Rate this app" and "About".

10.2.



Tapping on “Log Out” from the list, the app shows an alert dialog saying “Log Out. Are you sure you want to log out?” with options “Cancel” and “Log Out”.

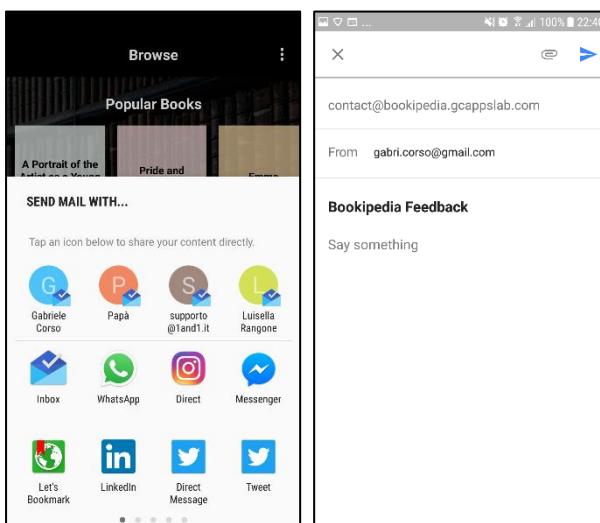
If “Log Out” option is selected the app moves to the LoginActivity and the user has to log in again (therefore all the user files are cancelled correctly).

10.3.



Tapping on “Suggest a book” from the list, the app correctly opens the SuggestionActivity.

10.4.

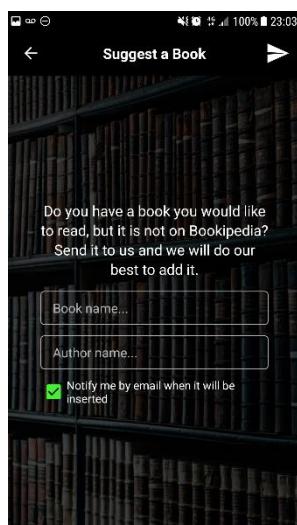


Tap on “Feedback” from the list, the system opens a menu to choose with which other client app the user wants to send an email. Selecting “Inbox” (or another client) the selected app is opened with an email already directed to the correct email address (contact@bookipedia.gcappslab.com) and subject “Bookipedia Feedback”

SuggestionActivity

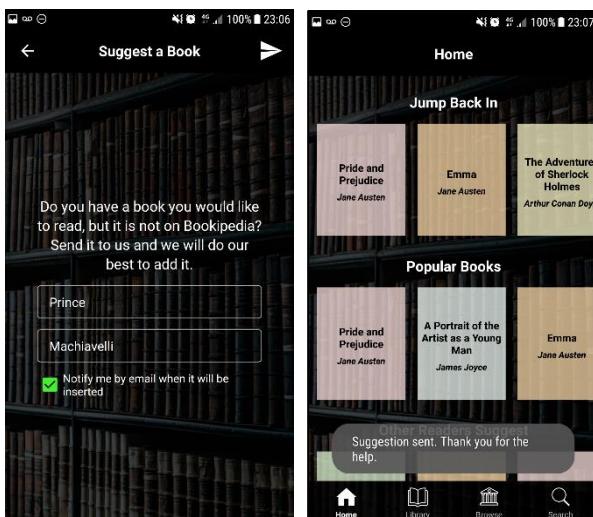
Test N°	Description & Expected result	Inputs	Result
11.1	The program should display the two input boxes for book and author names, a checkbox for the update and a send icon.	Open the activity from BrowseActivity	✓
11.2	If the user adds the details of a book and presses the send button the app should move to the home activity and display a success message. On the server database a correspondent tuple should be saved.	Input: "Prince", "Machiavelli" and press the send button.	✓
11.3	If the user unchecks the update checkbox the app should not save on the server database the user email.	Input: "The Little Prince", uncheck the update box and press the send button.	✓
11.4	If the user leaves the book field blank and tries to send the suggestion, the app should display an appropriate error message and not save the suggestion.	Tap on the send button with the book field blank	✓
11.5	If the user tries to send a suggestion without internet connection, the app should display an appropriate error message.	Input: "The Little Prince" and press the send button without internet connection.	✓

11.1.



Opening the activity from BrowseActivity, the app displays the two input boxes for book and author names, a checkbox for the update and a send icon.

11.2.



Input: “Prince”, “Machiavelli” and press the send button. The app correctly moves to the HomeActivity and displays a success message “Suggestion sent. Thank you for the help.” On the server database a correspondent tuple is saved.

suggestionid	userid	email	book	author
4	11	g@g.com	Prince	Machiavelli

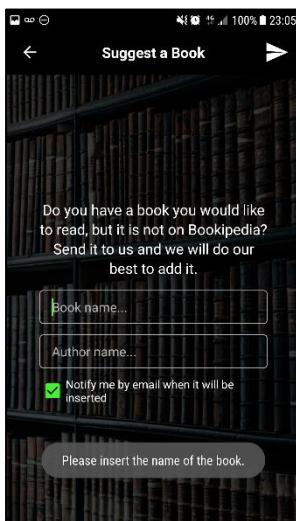
11.3.



Input: “The Little Prince”, uncheck the update box and press the send button. The app saves the tuple of the suggestion with the email field blank on the server database.

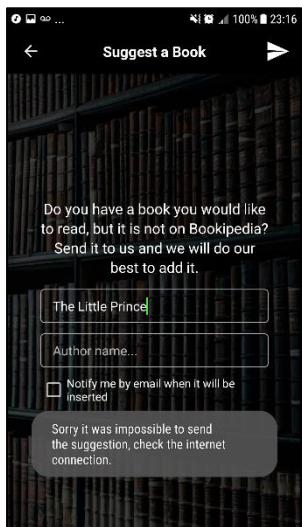
suggestionid	userid	email	book	author
3	11		The Little Prince	

11.4.



Tapping on the send button with the book field blank, the app displays the error message “Please insert the name of the book” and does not save the suggestion.

11.5.



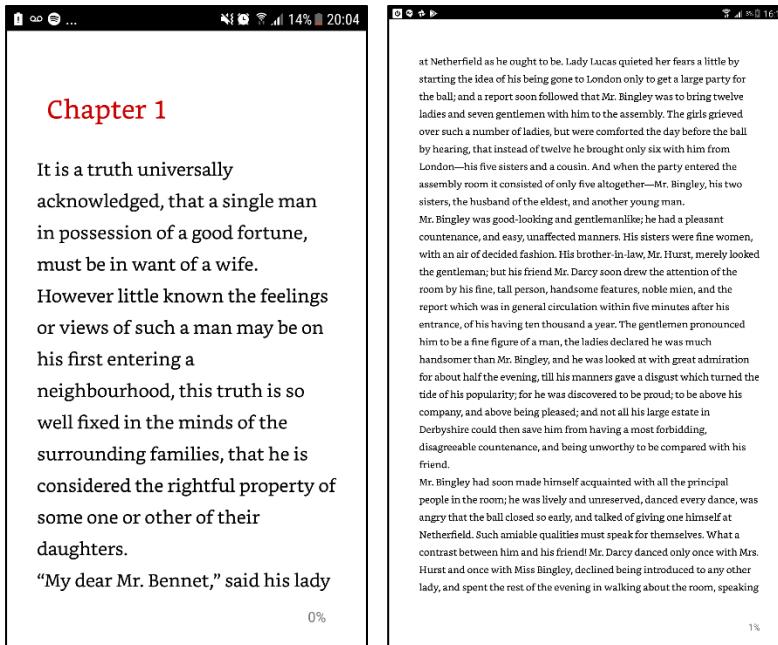
Input “The Little Prince” and press the send button without internet connection. The app displays the error message “Sorry it was impossible to send the suggestion, check the internet connection”.

ReadingActivity

Test N°	Description & Expected result	Inputs	Result
12.1	The first time the user opens a book the app should open the first page of the book showing the chapter layout with title. The text should fill and fit in the page.	Open "Pride and Prejudice" for the first time	✓
12.2	The user should be able to navigate in the book by tapping on the right (next page) or left (previous page).	Tap on right of the page and then on the left.	✓
12.3	If the user taps on the centre of the screen the activity should show the control design. Tapping again on the centre the control design should disappear.	Tap on the centre of the screen and then tap again.	✓
12.4	If user slides on the quick navigation bar of the control design, the app should quickly move to different parts of the book. When the control design is off the activity should always show the progress in the book.	Tap on the centre of the screen, then move to the middle, then tap in the centre.	✓
12.5	If the user taps on the font control icon the app should open the font menu.	Tap on the font control icon	✓
12.6	If the user changes the font the app should change the layout but still fit the text nicely in the screen. If the app is closed and reopened the choice of the user should be saved.	In the font menu: increase the font, change the font to Baskerville and colour to black.	✓
12.7	If the user taps the note icon the app should open a dialog to allow the user to type a note in the current position. If the user taps "save" the note should be saved.	Tap on the note icon, type "Test note" and tap "Save".	✓
12.8	If the user taps on the menu icon, the app should show all the options.	Tap on the menu icon	✓
12.9	If the user taps on "Add bookmark" from the menu, the app should save a bookmark in the current position and display an appropriate success message.	Tap on the menu icon, tap on "Add bookmark" and then open the notebook.	✓

12.10	If the user presses on a specific part of the text, the app should open an option menu, if the user selects “highlight” the app should highlight the text and save the highlight.	Press on the words “Netherfield Park”, tap “highlight”.	✓
12.11	If the user taps on “Notebook” from the menu, the app should open the notebook where all the notes, bookmarks and highlights of the current book are listed. If the user presses an item in the list the activity should move to the position in the book where it was taken.	Tap on “Notebook”, then tap on the first bookmark.	✓
12.12	If the user taps on the delete icon in a note item the note should be deleted.	Tap on the delete icon of the first bookmark.	✓
12.13	If the user taps on “Read Aloud” from the menu, the app should start reading the book aloud from the beginning of the current page and continue reading turning pages automatically. If the user taps on stop reading the app should stop immediately.	Tap on “Read Aloud”, let the app read two pages and then tap on “Stop Reading”	✓
12.14	If the user taps on “About this book” from the menu, the app opens a BookActivity of the current book.	Tap on “About this book” from “Pride and Prejudice” ReadingActivity	✓
12.15	If the user logs out or opens the app on another device when he reopens a book the last position and notes should be retrieved.	Leave the “Pride and Prejudice” at 16% with text highlighted on the same page, log out, then log in and reopen the book.	✓
12.16	If the user tries to navigate after the last page of a book he should be directed to the RatingActivity of the same book.	Get at the end of “Pride and Prejudice” and press on the right of the screen.	✓

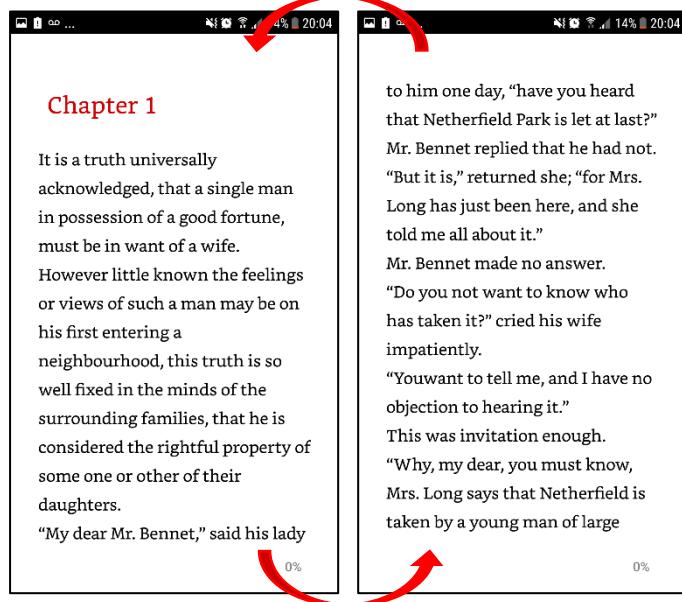
12.1.



Opening “Pride and Prejudice” for the first time the app shows the first page of the book. The text fills and fits in the page.

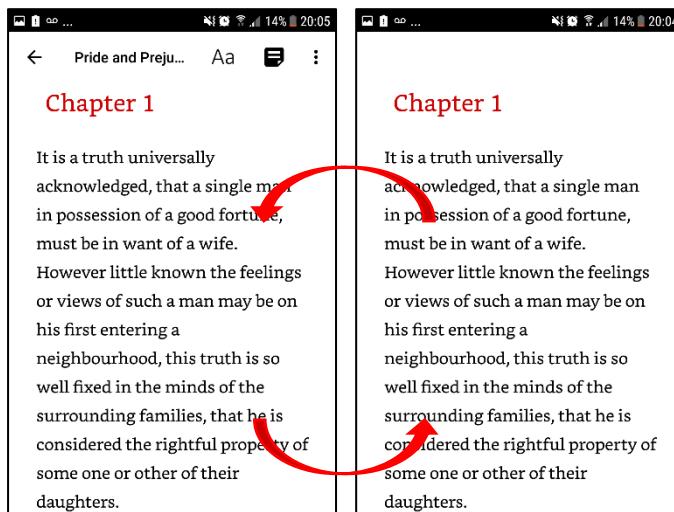
The second image is from a 10 inch tablet: as can be seen the text still fit dynamically in the page and it is not stretched .

12.2.

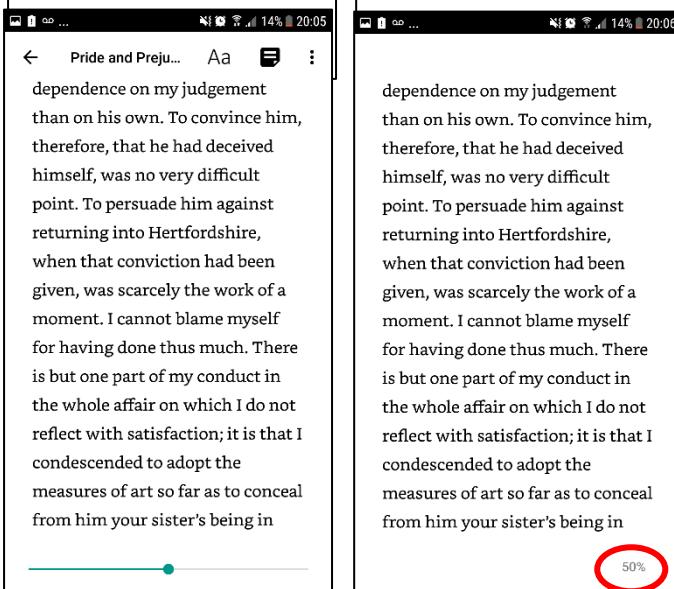


Tapping on the right (next page) or left (previous page) the user is able to navigate the book.

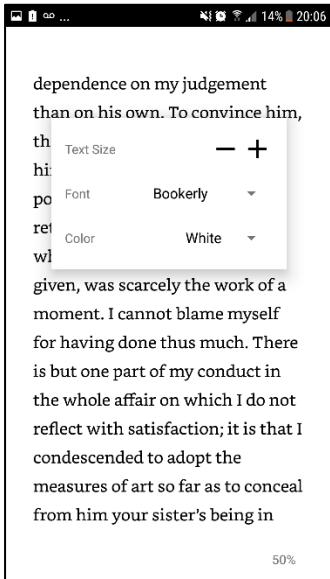
12.3.



12.4.



12.5.

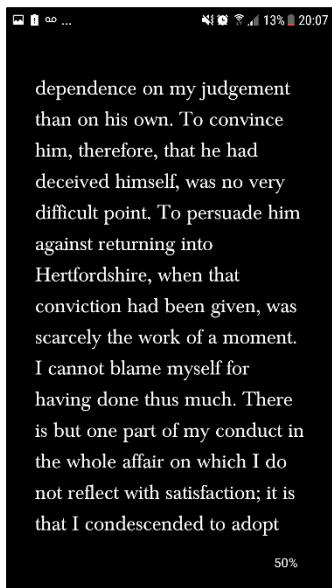


Tapping on the centre of the screen the activity shows the control design: a bottom bar for the quick navigation and a top bar with many other options and the title of the book. Tapping on the centre again the control design disappears.

Tapping on the centre of the screen and then sliding the bottom bar to the middle, the app quickly moves to the centre of the book. Tapping the centre of the screen, the control design disappears and the activity on the bottom right correctly shows the progress at 50%.

Tapping on the font control icon the app opens the font menu. This gives option to change text size, choose between seven different fonts and choose a background colour.

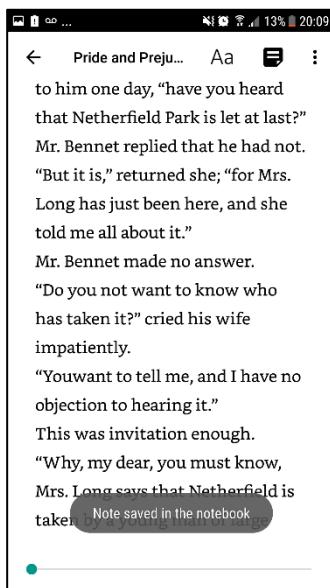
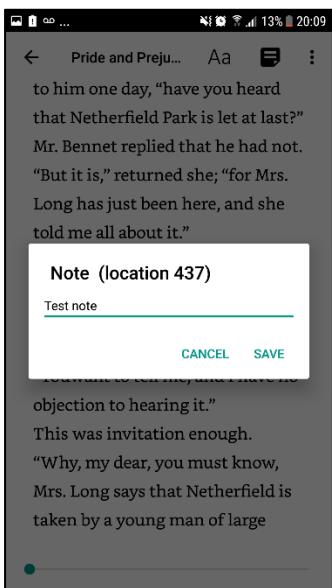
12.6.



In the font menu: increase the font, change the font to Baskerville and colour to black.

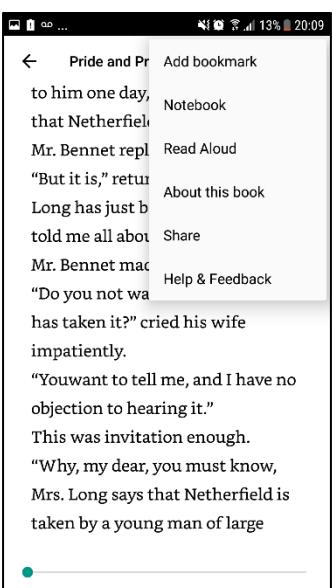
The app changes the layout in real time and the text still fit the text nicely in the screen. If the app is closed and reopened the layout is still the one chosen.

12.7.



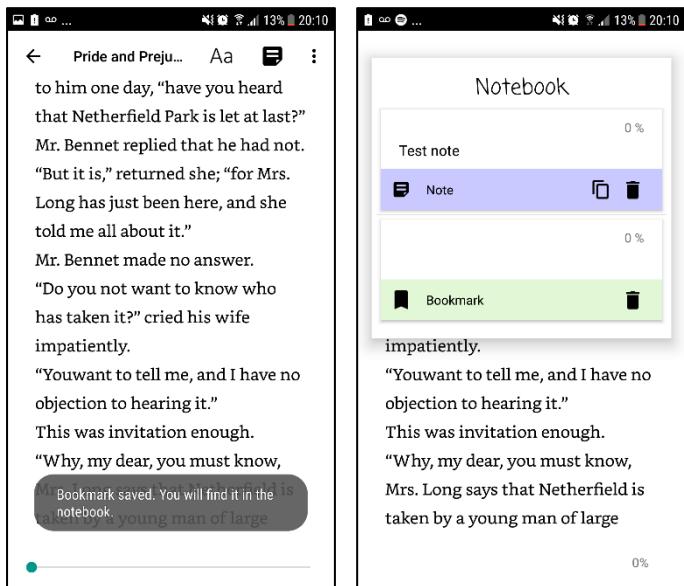
Tapping the note icon, the app opens a dialog with a text box. Typing "Test note" and tapping "Save", the app closes the dialog and prints the toast message "Note saved in the notebook".

12.8.

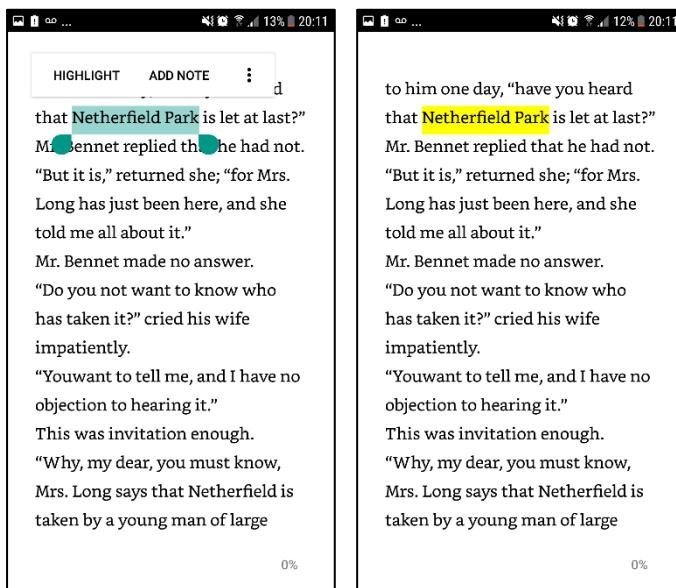


Tapping on the menu icon, the activity shows the options: "Add bookmark", "Notebook", "Read Aloud", "About this book", "Share" and "Help & Feedback".

12.9.



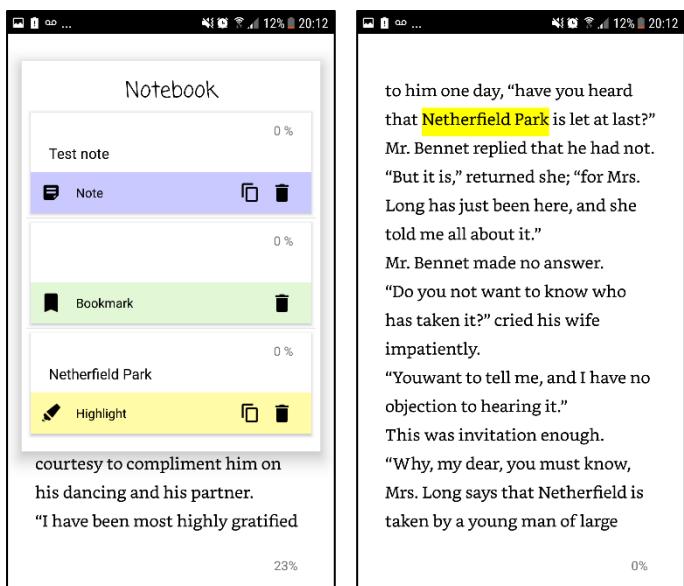
12.10.



Pressing on the words "Netherfield Park", the app shows a menu with options "Highlight", "Add note", "Copy", "Share", "Translate" and "Search Wikipedia".

Tapping then on "Highlight" the app highlights the text selected.

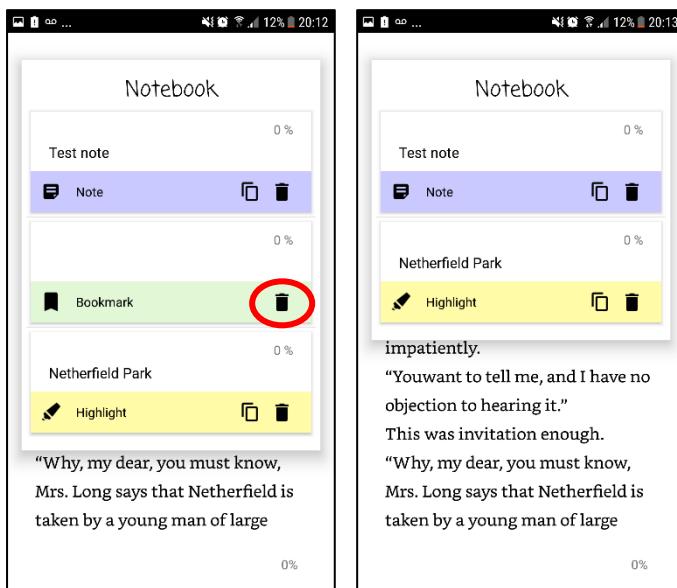
12.11.



Tapping on "Notebook" from the menu, the app opens the notebook where the three notes, bookmarks and highlights previously added in the book are listed.

Tapping on the first bookmark, the app moves to the point of the book where this was taken.

12.12.

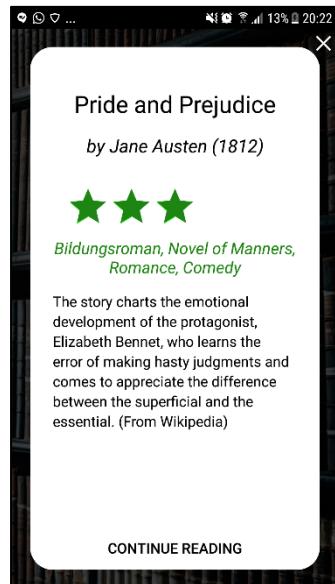


Tapping on the delete icon of the first bookmark, the bookmark is deleted and removed from the list.

12.13.

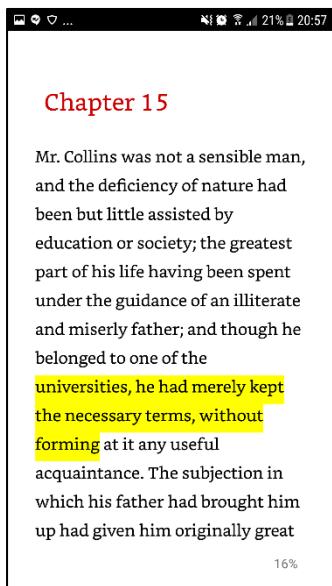
Tapping on “Read Aloud” from the menu, the app starts reading the book aloud from the beginning of the current page and continue reading turning pages automatically. After two pages tapping on “Stop reading” the app stops immediately. Video evidence of this test can be found at: <http://bit.ly/2FCH0kV>

12.14.



Tapping on “About this book” from the menu (still from “Pride and Prejudice” ReadingActivity), the app opens a BookActivity of the current book.

12.15.



Leaving “Pride and Prejudice” book at 16% with text highlighted on the same page, then logging out and logging in again, when the book is reopened the position is correct and the text is still highlighted.

bookid	userid	rating	progress	notes	active
3	11	-1	679634	437<->Test note<->N<<-->>474<->Netherfield Park<->...	1

12.16.



Getting at the end of “Pride and Prejudice” and pressing on the right of the screen, the app opens the RatingActivity of the same book. .

RatingActivity

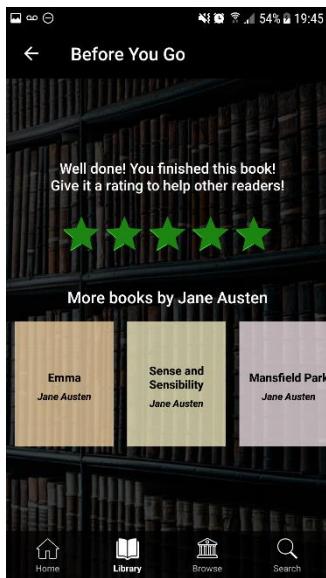
Test N°	Description & Expected result	Inputs	Result
13.1	The program should display a rating bar and a list of books of the same author of the current book	Go after last page of "Pride and Prejudice" from ReadingActivity	✓
13.2	If the user sets a rating, it should be saved in the database	Set the rating bar to 5 stars	✓
13.3	If the user taps of the ← back navigation button the app should go back to the last page of the current book	Tap on ← back navigation button	✓
13.4	If the user taps on a book from the list, the corresponding BookActivity should be opened.	Tap on "Emma"	✓
13.5	If the user sets a rating on a book he had already rated, the rating on the server database should be correctly updated	Set the rating bar to 1 star	✓

13.1.



Going after last page of "Pride and Prejudice" from ReadingActivity, the app opens the RatingActivity showing an empty rating bar and a list of books of Jane Austen.

13.2.



Setting the rating bar to 5 stars, the app sets the rating in the corresponding reading tuple on the database and adjusts the rating and reading fields in the book table.

*Book table**Before:*

bookid	name	author	authortname	year	description	file	genre	language	rating	reviews
3	Pride and Prejudice	3	Jane Austen	1812	The story charts the emotional development of the ...	<< chapter >>Chapter 1 It is a truth universally...	Bildungsroman, Novel of Manners, Romance, Comedy	English	3.000	1

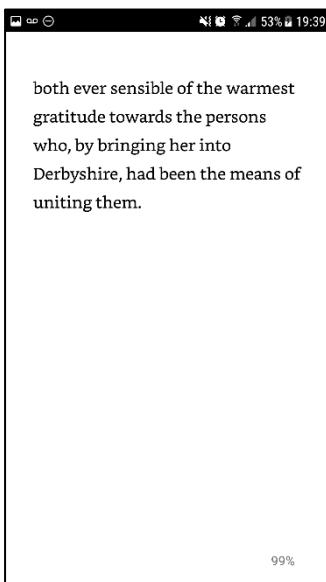
After:

bookid	name	author	authortname	year	description	file	genre	language	rating	reviews
3	Pride and Prejudice	3	Jane Austen	1812	The story charts the emotional development of the ...	<< chapter >>Chapter 1 It is a truth universally...	Bildungsroman, Novel of Manners, Romance, Comedy	English	4.000	2

Reading tuple after:

bookid	userid	rating	progress	notes	active
3	11	5	679634	437<->Test note<->N<<-->474<->Netherfield Park<->...	1

13.3.



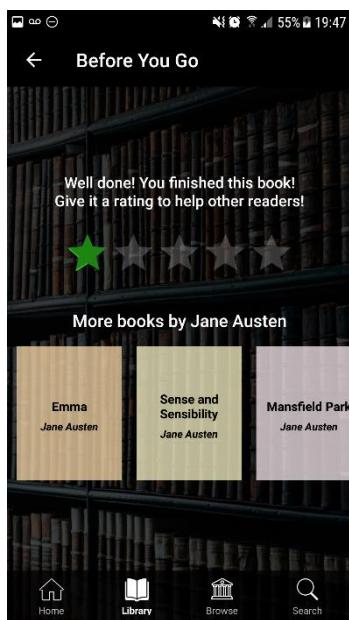
Tapping on ← back navigation button, the app goes back to the last page of the book "Pride and Prejudice".

13.4.



Tapping on “Emma” from the list, the corresponding BookActivity should be opened.

13.5.



Setting the rating bar from 5 to 1 star, the app sets the new rating in the corresponding reading tuple on the database and adjusts the rating and reading fields in the book table.

Book table after:

bookid	name	authorid	authorname	year	description	file	genre	language	rating	reviews
3	Pride and Prejudice	3	Jane Austen	1812	The story charts the emotional development of the ...	<< chapter >>Chapter 1 It is a truth universally...	Bildungsroman, Novel of Manners, Romance, Comedy	English	2.000	2

Epub to txt converter

If the Java program is run the program converts the HTML of each file of the book into text and put them together in a single text file adding chapter markups.

Test 14: This is an example of the first page of the first chapter from the “Pride and Prejudice” book that can be downloaded from Project Gutenberg website at this link <http://bit.ly/2FAUCRP>.

Epub

Chapter 1

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.

However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters.

“My dear Mr. Bennet,” said his lady to him one day, “have you heard that Netherfield Park is let at last?”

Mr. Bennet replied that he had not.

“But it is,” returned she; “for Mrs. Long has just been here, and she told me all about it.”

Mr. Bennet made no answer.

Converted file

<<||chapter||>>Chapter 1

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.

However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters.

“My dear Mr. Bennet,” said his lady to him one day, “have you heard that Netherfield Park is let at last?”

Mr. Bennet replied that he had not.

EVALUATION

Evaluation against specific objectives

After the end of the testing I started evaluating the app based on the specific objective I set in the analysis section:

Reading activity

12. *Read the customised .txt file and display it clearly in an activity, with a text dimension that is absolute and not relative to the size of the page (very large for tablets and too small for small smartphones) in order to make the books easily readable on any device.*

As can be seen in the test 12.1 the app makes the text fit nicely on devices of every dimension

13. *Memorise the last position that will be retrieved whenever a book is reopened.*

Every time the user turns a page the new position is saved in the local database first and then also in the server, so as text 12.15 shows the last position is always retrieved when a book is reopened.

14. *Navigation between pages with swipes or touches on left (previous page) and right (next page) sides of the screen and access to the control bars with a touch in the middle.*

As can be seen in tests 12.2 and 12.3 the app implements all this functionalities.

15. *Quick navigation through the book pages using a bottom bar that compares with the controls after a touch in the middle of the screen.*

This feature was implemented successfully and tested in test 12.4

16. *Display of the progress on a book (page).*

This is shown, as can be seen in test 12.4, by a percentage displayed at the bottom right of the page. I chose to show it as a percentage rather than a page as the concept of page is relative to the font type and dimension and device size. Since the user can change font and change device while reading the same book an indication of the page number would probably be insignificant to him.

17. Taking notes while reading, the note will be linked to the position of the book where it was taken.

As shown in test 12.7 the user can easily take a note in the current position of the book and, as shown in test 12.11, return to the position where the note was taken by tapping on the note in the Notebook.

18. Highlighting of text, which has to stay highlighted every time the user opens that book again. Users will also be able to retrieve all the highlighted pieces of text in a book and the notes in a summary page.

Test 12.10 shows how, when the user select some text, he is given the possibility to highlight it and if he does so the text is highlighted and remains so every time the page is reopened (test 12.15). The summary page is called Notebook, it is accessible by the menu (test 12.11) and contains all the notes, highlights and bookmarks of the user in the current book. Notes are synchronized so that the user can retrieve them when he read with different devices.

19. Saving bookmarks, where the user can easily come back

A bookmark can be easily inserted from the control menu (test 12.9) and accessed through the Notebook (test 12.11).

20. Font control through which you can change:

- a. Text dimension*
- b. Text font*
- c. Light / Dark / Sepia themes.*

As shown in test 12.7 the user can change all three option and the page is adjusted accordingly. The user choices are saved and the page is reloaded with the same options the following time.

21. For audio books (registration of books read out loud by a reader) while a storage of a registered voice would be really heavy in term of space and in particular to create in the first place, a function of the app that would automatically read (using some external API) the text would be a good compromise.

As shown by test 12.13 the app uses Google text-to-speech API to read text automatically aloud when the “Read Aloud” option is selected.

22. Possibility to rate a book once the user has finished reading it.

Tests 13.2 and 13.5 show that the user can rate (and change the rating) a book and the rating is saved in the server database.

Library

14. Registration with a new app account, giving email, password and favourite genre.

Test 1.1 shows how a user can register.

15. Log in with email and password.

As shows test 2.3 shows how a user can login.

16. Logged user registration to avoid having to log in every time.

The app memorizes the user data on a local file, when the app is reopened the user is not asked the email and password again (test 2.1).

17. Shelf section where the books read are displayed.

In the app it has been called LibraryActivity and, as shown by its tests, shows the list of the book downloaded (and therefore started) by the user.

18. Home section where the users are suggested books with algorithms that consider the following criteria:

- a. Most popular: based both on the number of downloads and on the ratings of other users.*
- b. Recommended for the user: based on favourites genres.*

In the app it has been called HomeActivity and displays lists of books based on the downloads ("Popular books"), the rating ("Other users suggest") and the favourite genre of the user ("Popular in ...").

19. Browse section where the user can navigate in the library according to genres.

The BrowseActivity allows users to choose between 14 different genres and if a genre is selected related books are displayed (tests 5.3 and 6.1).

20. Search section where a book or an author can be searched

As shown in tests 7.2 , in the SearchActivity the user can search for a book or author and open them.

21. Book overview: when a user selects a book from the library a window with info on the book is opened. This shows title, author name, year, rating, genre and description with a button to start the reading.

Tests 8.1 and 8.3 show how in the book activity shows all the details of a book and give the possibility to a user to start reading a book

22. Author page: displays author's name, books and brief biography.

The app implements this functionality: the AuthorActivity shows name, biography and a grid of books of the author ordered by users rating (test 9.1) .

23. Creation of a local database that will store the data of the books the user read or is reading on the device so no internet connection is needed to finish a book started or look back at a book finished.

The app keeps a local database always updated (check done when the user opens the app) with the list of books started by the user. These books can be accessed even offline through the LibraryActivity (test 4.7).

24. Choice for the user of the order in which books are displayed in some lists in the library

Both the Library and the Genre activities offer the user the possibility to choose the order (tests 4.3 and 6.2) in which the books are displayed (for example last opened, title, author name, year, downloads and rating).

25. Possibility to suggest a book that should be added.

The SuggestionActivity allows the user to suggest a book and, if he wants, be notified when the book is added (tests 11.2 and 11.3).

26. Synchronization between multiple devices so that the user can start reading on a certain device and continue on others and notes, highlights and progress are memorized.

As shown in test 2.2, books, progress and notes are always synchronised on the device when the user opens the app with a different device from the last time and on the server when the user modifies them on the device.

Backend

6. *Working backend with a database with books, authors and users information and all the actual eBook files.*

Tests of the app shows that the presence of data.

7. *Strong and secure encryption for the passwords.*

The passwords are encrypted on the webserver with the standard php hashing function which is therefore secure and then just the hashed password is stored.

8. *Actively respond to calls from the app for login, registration, query of eBooks, query of authors, download of eBooks*

Tests of the app shows that the whole webserver works.

9. *Creation of a program through which book can be converted from ePub to .txt with the correct mark-up*

As shown by test 14, the program successfully runs on computers and converts ePub files to text file adding chapter mark-ups.

10. *Insert about 100 eBooks with all the related information*

Thanks to the Java program for the conversion of e-books the process of insertion of a book takes less than 5 minutes. As to 23/03/18 there are 75 books on the database and books are still added every day so they will soon pass the 100 quota.

User Feedback and Discussion

I then published the app on the Google Play Store (official app store for Android) in order to make it available to everyone with an android device.



Bookipedia - Read Books for Free - Open Library

Gabriele Corso · Books & Reference

3 PEGI 3

! This app is compatible with all of your devices.

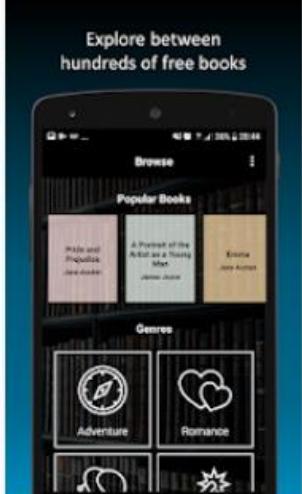
Installed



Read everywhere even offline



Join the first open mobile library



Explore between hundreds of free books

>

Bookipedia is an open mobile library. We believe that culture should be open and accessible to everyone, therefore we offer you a completely free app where you can read hundreds of books.

- Unlimited Reading – Choose from hundreds classics books we legally offer you.
- Read Books for Free – All books are completely free and without ads.
- Read Books Anywhere – Bookipedia allows you to download any book and access it offline. You can also read the same book across multiple devices with automatic synchronization.
- Read Books as you Like – Choose your favorite font, dimension, background color and orientation.

Enjoy reading!

I obtained feedbacks both from specific students in my school I mentioned in the Analysis section and general android users that reviewed my app in the first few day since its publication on the android app store.

Eesha Irfan

Feedback on Bookipedia



Eesha Irfan

Yesterday, 12:00

Gabriele Corso



Reply all | ▾

Dear Gabriele,

I have had the chance to download your app, use it and rate it. I think it is a very robust and user-friendly app. The layout and design are easy on the eyes and do not look like amateur's work. The app makes use of symbols universally used and so it is easy to understand their use such as the edit button for bookmarks and notes. The login screen is beautiful and inviting. Overall, the design has been very carefully and successfully implemented.

On the note of user friendliness, I like how when I created a bookmark or added a note to my book, it told me where to look to find them. This shows me that the program communicates with the user well. However, in terms of logging out, it is hard to find. You must go to the help and feedback section to log out, this is not fairly obvious.

I like the options I have while reading a book, caters for people of reduced sight and those with impaired hearing. For those who want to read at night, the sepia option for the page colour makes it easier to look at the screen and not strain the eyes. One suggestion I would say is that the voice of the reader is quite monotonous and some variety on the reader's voice would be nice. Moreover, having page numbers would be nice alongside the bookmark.

Overall though, I would say that this app is very well done and I already see many good reviews for it in the google play store. Very well done!

Kind Regards,
Eesha

Comments:

Eesha's remark on the log out are completely understandable, I was aware that I would not be easy to find in the Help and Feedback activity. A link to log out can be very easily added to the menu of the BrowseActivity with few lines of code, however logging out is not part of the ordinary use of the app. If a user was to log out and in every time, this would cost him time in inserting details and synchronization, therefore I am not sure that such action should be facilitated.

About the remark on page number as anticipated is a concept, that, while is universally used with paper books, with ebooks is relative to the font type and dimension and device size. Since the user

can change font and change device while reading the same book an indication of the page number would probably be insignificant to him. I explained this situation to Eesha afterwards and she also agreed that then the percentage would be better than an estimation of the screen sizes left that would change with the font.

Daniel Gibbons

[Bookipedia - feedback](#)



Daniel Gibbons
Wed 21/03, 09:06
Gabriele Corso

Reply all | v

The app seems quite useful, with a good range of books and a layout that is easy to navigate. The division of books into genres makes it particularly easy to find a book you might like, and the search function also seems quite functional.

I would recommend adding book cover icons to each book to improve the aesthetics of the app, and to make the books more recognisable. It would also be useful to have an in-built dictionary function to allow words to be looked up easily.

Comments:

Daniel's suggestion on an in-built dictionary is interesting and might be really useful particularly when reading classical books. A simple way to implement it is to use an external API (for example there is one from Oxford Dictionary) and give the user the opportunity to call the definition of a word when selecting it in the book.

The book cover's remark refers to a problem I had put myself in the Design section. The major issue with book covers is that while books that are added to the app are not subject to copyright, most of the covers would. Therefore, since creating all of them myself is unfeasible I am planning to give users the opportunity to upload covers they created themselves through a web platform.

Jeffrey Yuen

Bookipedia - app feedback



Jeffrey Yuen

Yesterday, 23:53

Gabriele Corso



Reply all



This app is actually nice. I can read many famous books in this app which I haven't read before. It includes more than 100 books so I can pick any one I like from a huge pool of books.

I also like the rating system and "what other people like" system. Sometimes it may be difficult for me to choose a new book for me to start reading it. These systems give me suggestions and help me to choose.

The only bad thing I found are few typos in some books and I hope they will be removed.

Jeffrey Yuen

Year 13

Comments:

Jeffrey's remark of few typos in the books is true and due to typos in the sources where I get the books from, however it would be infeasible for me to read and correct all the books before adding them. Therefore, as with covers, the idea is to integrate this in the web platform open to all users: here the users could correct/point out typos, which will be corrected in the database.

Nikini Kottegoda

Review of Bookipedia



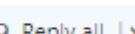
Nikini Kottegoda

Yesterday, 09:10

Gabriele Corso



Reply all



The app design is very intuitive and easy to use. I particularly like the fact you can search specific authors and it will give take you to a page of the author giving you some brief information as well as easy to access thumbnails to a range of books. Navigation between genres is also easy and done very well with an attractive interface. However, I would like to see the implementation of more filter abilities so that you can search through an author or genre by popularity or release date etc. I like how easy you can change the text colour and highlight particular parts of the book, which would be very useful when taking notes and studying. If I were to change anything with the interface provided for reading I would probably add the ability for a variety of scrubbing speeds with page numbers visible at the bottom of the screen.

Comments:

Nikini's remark of more filters is interesting and I had some ideas of new filters myself such as the last added to the database or most popular authors for each genre. To add these filters for example in the HomeActivity is easy as the layout and callback structure are already there, the only addition would be the specific SQL query to retrieve books corresponding to the specific filter.

The reviews from the app store were really good and in the first four days (from 21 to the 24 March) the app had 23 reviews with an average of 4.91 / 5.



Overall Evaluation

Given that the success of all the typical tests and the achievement of all the specific objectives I believe that the app was designed and implemented successfully. Moreover, the app has passed all the boundary and erroneous tests and, after a week on app store, it has had no crashes. This suggests that not the app is also robust and can cope with situations like the use without internet connection or an attempt of SQL injection.

I am also very satisfied with the design of the app which, as expressed also by the users in the feedbacks, is clean and easy to understand, but at the same time coloured and attractive. I am also happy with the structure and contents of the backend. All the requests are retrieved very quickly even when there are many people using it at the same time. The current number of books in the database is slightly lower than what I had set, but, thanks to the converter program, the time to insert a single book has been cut incredibly and therefore I am confident that the number of books will reach 100 and quickly pass it in the next few days.

Extension

Also thanks to the feedbacks I received, I have outlined what extension I am going to make in the next development cycle of this project. First, I will continue to add new books and authors to the database. These updates are synchronised directly in the user's app without the need of updates. Moreover, I will publish an update of the android app with the implementation of some extra functionalities starting from the English dictionary suggested by Daniel.

If using the Oxford dictionary API for example the retrieval of the definition of the word will look like any other backend callback. The https request will be headed to:

`https://od-api.oxforddictionaries.com:443/api/v1/entries/en/ + word`

With parameters the app id and key. The request will return a Json response containing all the details of the word, which will then be displayed in the app.

Another functionality I will implement is the choice of the preferred language and so the possibility to read books in different languages. Thanks to the way I designed the backend, inserting in the book and user table the attribute "language" (which for now are always set by default to "English"), this change will be very easy. I will just need to ask the user at the registration to choose his preferred language between the ones supported and then add the condition "*WHERE language = \$preferred_language*" in all the queries for books. These changes will be followed by the insertion of a good amount of books in each language provided and a translation of the whole all (e.g. textboxes and menus) in each of these languages.

Secondly, I am planning to make a version for iOS in order to support almost all mobile devices. This process is relatively easy as, while it is true that I will have to use another language (Swift), the overall structure of the program and the backend calls will be the same.

Finally, as previously anticipated, I am going to create a webserver where users can contribute to the library. In this platform it will be possible to upload new books as well as correct typos, update descriptions and upload covers of the books.