GitHub Username: Vanessaguillemain

Book Study Planner

Description

You have several books or courses to read for different dates before exams, and you want to go step by step and finish in time with no stress?
Book Study Planner will generate a study plan for each book you need to read, based on your deadlines and days of reading. Every day you will know how many pages of each book you have to read to reach your goals. If one day you miss the scheduled reading, Book Study Planner will automatically recalculate for you a new reading schedule.
Graphical job tracking will motivate you and help you keep the track.
This app is designed for books, but you can also follow your course reviews.

Intended User

Students, active people, readers, they all can plan their lessons or books before deadlines.

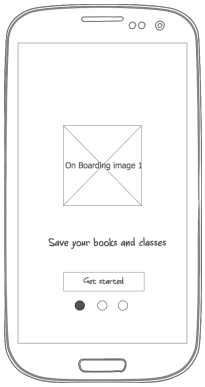
Features

- Saves the books or lessons you want to plan
- Saves your daily effective reads
- Book search by ISBN scan or web research
- Loads pictures to personalize your items
- Calculates a reading plan
- Recalculates the plan each day, depending on your progression
- shows graphical progress
- A widget presents your readings for the day

User Interface Mocks

Onboarding Screens

Screen 1

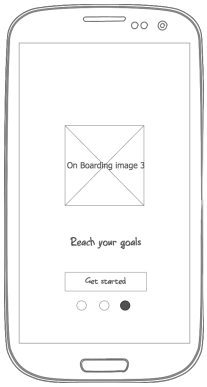


1st functionality : saving books

Screen 2

2nd : plan study

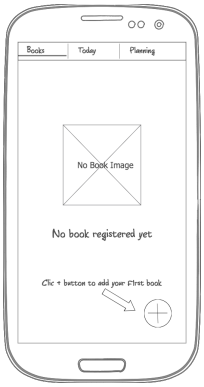
Screen 3



3rd : reach goals

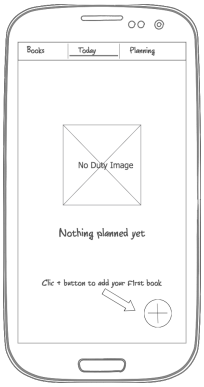
Screen 4 : Books empty

The app has 3 major Tabs : Books : for books managing, Today : the plan of the day and Planning, the global plan.



Screen « Books » empty : the user is invited to add a book by clicking the FAB.

Screen 5 : Today empty



Screen « Today » empty : the user is invited to add a book by clicking the FAB.

Screen 6 : Planning empty

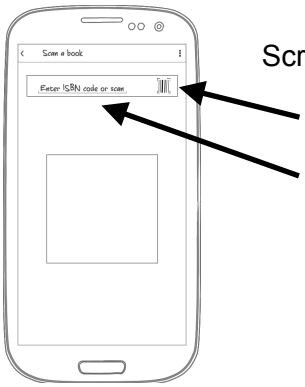
Screen « Planning » empty : the user is invited to add a book by clicking the FAB.

Screen 7 : FAB sub-menus

Tab « Planning » empty : **the user clicks the FAB** => 3 options to add a book :

- Input : the user input all data manually (=> **Screen 11**)
- Web search : opens a new window to search the book by title (=> **Screen 10**)
- ISBN : opens a new window to enter manually an ISBN code or to scan it (=> **Screens 8 and 9**)

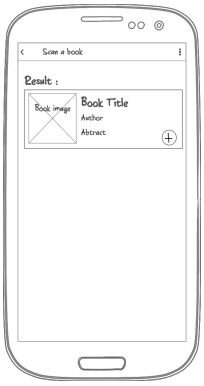
Screen 8 : ISBN search



Screen ISBN, when the user chose FAB ISBN

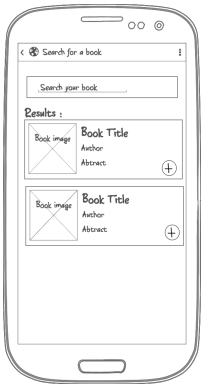
- This icon lets the user scan the ISBN code
- This input lets the user fill the ISBN code manually

Screen 9 : ISBN search



After the ISBN identification in screen 8, a book is proposed (or a message « not found »). The user has to clic the Add (+) button to enter the Screen « Create/ Modify book » (**Screen 11**).

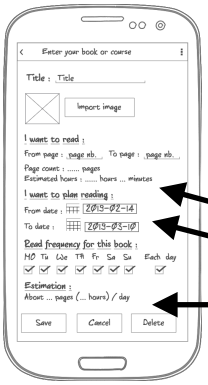
Screen 10 : Web search



Screen Web search, when the user chose FAB Web search. After the search a list of result is proposed (or a message « not found »). The input will be the book title.

The user has to clic the Add (+) button to enter the Screen « Create/Modify book » (**Screen 11**).

Screen 11 : Book detail



Screen « Create/Modify Book », when the user chooses FAB Input, or when the users clicks on the Add Button after an ISBN or Web search.

The user inputs *From* and *To* pages.

Page count is calculated, *Estimated hours* too.

Date pickers are opened to choose start and end date

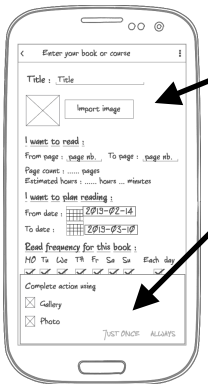
Number of pages a day is calculated according to the user input.

Button *Save* saves in DB and goes to **Screen 13**, list of books.

Button Delete deletes from DB (after conformation) and goes to **Screen 13**, list of books.

Button Cancels cancels modifications (after confirmation) and goes to **Screen 13**, list of books.

Screen 12 : Book detail : load image



Button *Import image* is present if no image was found (via ISBN service)

When the user clicks *Import Image*, an intent is sent allowing the user to choose a photo.

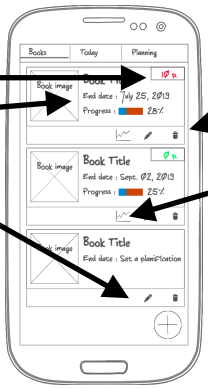
Screen 13 : Books

Screen Books : The list of books registered by the user

Number of pages to read today

Deadline

Modify the book (**screen 11**)



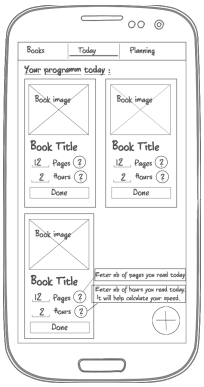
Progression indicator.

Delete the book from DB (after confirm)

See progression charts (**Screen 17**)

FAB Add (+) : add a new book.

Screen 14 : Today



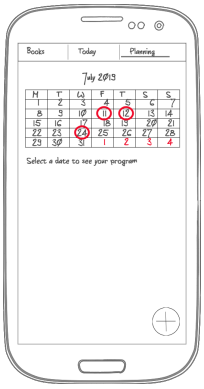
Screen Today : the list of readings planed for the actual day.

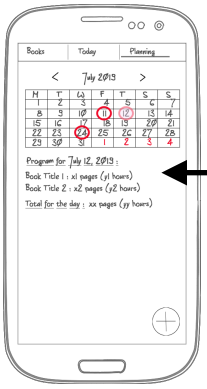
The user can modify the number of pages and hours planned : he enters his real readings, more or less than planned. If he inputs his time of reading the work time estimation can then be personalized.

FAB Add (+) : add a new book.

Screen 15 and 16 : Monthly Planning

Screen Planning : Each day with a reading planned is circled.

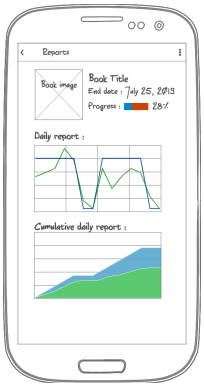




When the user selects a date in calendar, the app shows below the work planned for the day

FAB Add (+) : add a new book.

Screen 17 : charts



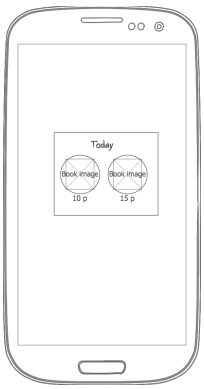
Presents 2 report charts for the book chosen.

The first, day by day, in one color we have the number of pages planned, and in another color the number of pages really read.

The second one, is a cumulative chart : after x days, the total number of pages that should be read, and the total number effectively read.

Back returns to the list of Books (**Screen 13**)

Screen 18 : Widget



Presents the work that is planned for the current day : an image of the book in a round, with under the number of pages.

Key Considerations

How will your app handle data persistence?

The app will use Room with LiveData and ViewModel.

Describe any edge or corner cases in the UX.

Physical Back button :
- activated when in tab Books, Today, Planning => exit the app
- activated when in Screen Scan a book -> return to the initial tab (can be Books, Today or Planning)
- activated when in Screen Web Search -> return to the initial tab (can be Books, Today or Planning)
- activated when in Screen Book Detail -> return to tab Books
- activated when in Screen Charts -> return to tab Books

Test Internet connection when Search Web and Scan ISBN is required.
Limit number of books to 10 (expanded in a paid version later)
Resize size of loaded images.
Test date input : begin and end date must be greater or equal than today.
Screen rotation keeps all similar.
Test in input Today : number of read pages and time of reading must be numbers.
Manual ISBN input is algorithmically tested before search.

Error message when an ISBN scan or input is not found.
Web search :
- Limit list of results
- Message if nothing was found

Describe any libraries you'll be using and share your reasoning for including them.

- **Picasso** for loading and caching images.
- **AppCompat**, for back-compatibility.
- **Design Support library**, for FAB and transitions, material design.
- **Butterknife** to inject views in components.
- **Gson** for json parsing.
- **Room** for managing data base.
- **LiveData** for optimizing data base access.
- **Okhttp3** for faster and optimized requests.

Describe how you will implement Google Play Services or other external services.

Google Books API.
It will be used to get book information in screens Web Search and ISBN.

For the Web search, if the user wants the title « android for dummies », we will send a request like :
https://www.googleapis.com/books/v1/volumes?q=android+for+dummies

For the ISBN search, if the user wants the ISBN 9782412015735, we will send a request like :
https://www.googleapis.com/books/v1/volumes?q=+isbn:9782412015735

Google Mobile Vision : Barcode API.
We will use the Barcode API in Screen Scan ISBN.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

- Configure libraries
- Something else

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for something else

- Develop activity and layouts for the 3 Main tabs : Books, Today and Planning
- Develop Entities, DAOs and Database form Room
-

Task 3: Your Next Task

Describe the next task. For example, “Implement Google Play Services,” or “Handle Error Cases,” or “Create Build Variant.”

Describe the next task. List the subtasks. For example:

- Create layout
- Something else

Task 4: Your Next Task

Describe the next task. List the subtasks. For example:

- Create layout
- Something else

Task 5: Your Next Task

Describe the next task. List the subtasks. For example:

- Create layout
- Something else

Add as many tasks as you need to complete your app.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]

    - Make sure the PDF is named “**Capstone_Stage1.pdf**”

- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”

- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”