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.

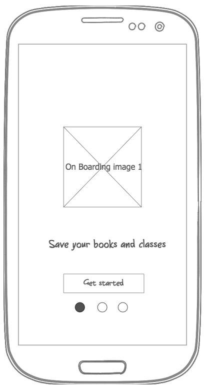
Features

- 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

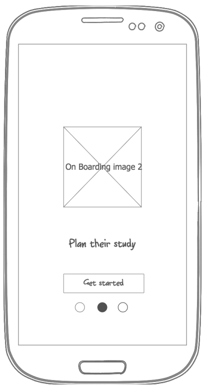
Onboarding Screens

Screen 1



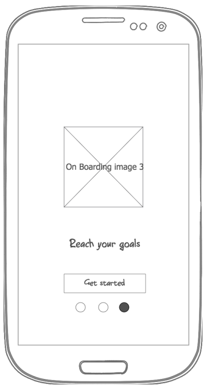
1st functionality : saving books

Screen 2



2nd : plan study

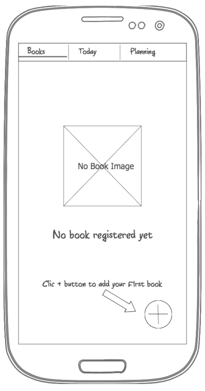
Screen 3



3rd : reach goals

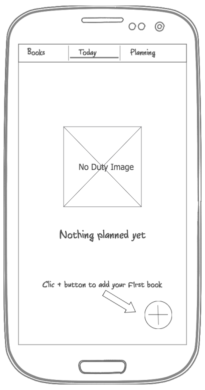
Screen 4 : Screen Books empty

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



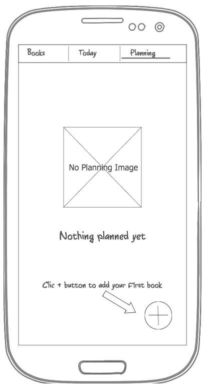
Screen « Books » empty : the user is invited to add a book by clicking the FAB.

Screen 5 : Screen Today empty



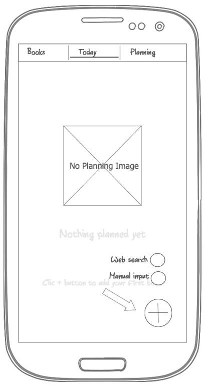
Screen « Today » empty : the user is invited to add a book by clicking the FAB.

Screen 6 : Screen Planning empty



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

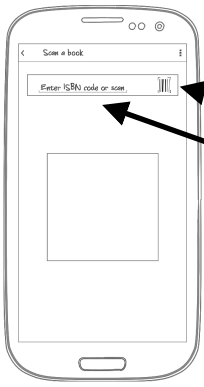
Screen 7 : FAB and sub-menus



Tab « Planning » empty : **the user clicks the FAB** => 2 options to add a book :

- Web search : opens a new window to search the book by ISBN, title, author.. Or scan the ISBN code (=> **Screen 8 and 9**)
- Manual Input : the user input all data manually (=> **Screen 10**)

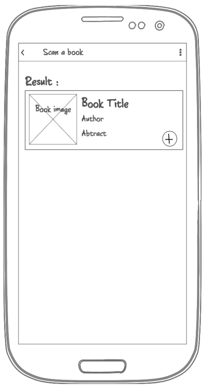
Screen 8 : ISBN search



Screen ISBN, when the user chose FAB ISBN

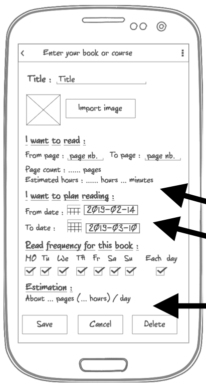
- This icon lets the user scan the ISBN code
- This input lets the user fill the ISBN code manually

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 « Book Detail» (**Screen 10**).

Screen 10 : Book detail


Enter your book or course
Title : Title
Import image
I want to read :
From page : page nb. To page : page nb.
Page count : ...... pages
Estimated hours : ...... hours ... minutes
I want to plan reading :
From date : 2019-02-14
To date : 2019-03-10
Read frequency for this book :
MO Tu We TH Fr Sa Su Each day
Estimation :
About ... pages (... hours) / day
Save
Cancel
Delete

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

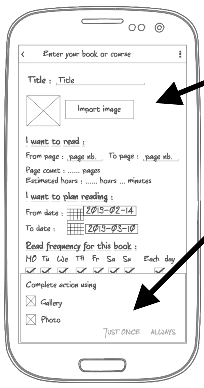
Number of pages a day is calculated according to the user input.

Button *Save* saves in DB and goes to **Screen 12**, list of books.

Button Delete deletes from DB (after conformation) and goes to **Screen 12**, list of books.

Button Cancels cancels modifications (after confirmation) and goes to **Screen 12**, list of books.

Screen 11 : Book detail : load image


Enter your book or course
Title : Title
Import image
I want to read :
From page : page nb. To page : page nb.
Page count : ...... pages
Estimated hours : ...... hours ... minutes
I want to plan reading :
From date : 2019-02-14
To date : 2019-03-10
Read frequency for this book :
MO Tu We Th Fr Sa Su Each day
Complete action using
Gallery
Photo
JUST ONCE ALWAYS

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.

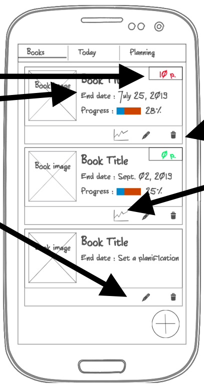
Screen 12 : Books

Screen Books : The list of books registered by the user

Number of pages to read today

Deadline

Modify the book (**screen 10**)



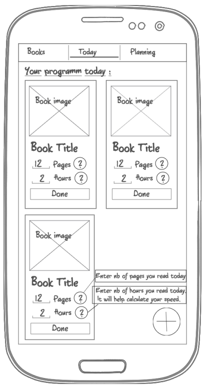
Progression indicator.

Delete the book from DB (after confirm)

See progression charts (**Screen 16**)

FAB Add (+) : add a new book.

Screen 13 : Today


Books
Today
Planning
Your programm today :
Book image
Book Title
12 Pages
2 Hours
Done
Enter nb of pages you read today
Enter nb of hours you read today. It will help calculate your speed.

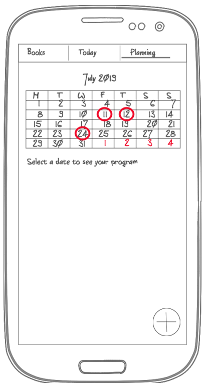
Screen Today : the list of readings planed for the actual day.

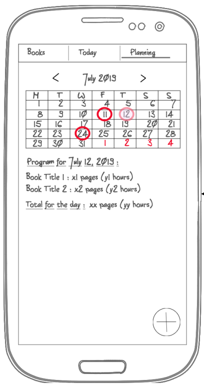
The user can modify the number of pages and hours planned : he enters his real readings, more or less than planned. If he inputs his time of reading the work time estimation can then be personalized.

FAB Add (+) : add a new book.

Screen 14 and 15 : Monthly Planning

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

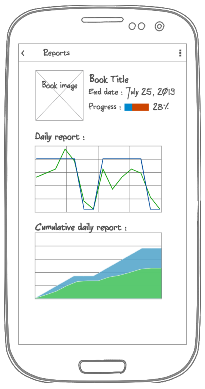



Books
Today
Planning
<
July 2019
>
M T W F T S S
Program for July 12, 2019 :
Book Title 1 : x1 pages (y1 hours)
Book Title 2 : x2 pages (y2 hours)
Total for the day : xx pages (yy hours)

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 16 : Reports



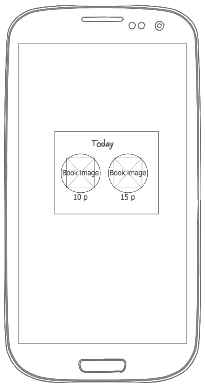
Presents 2 report charts for the book chosen.

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.

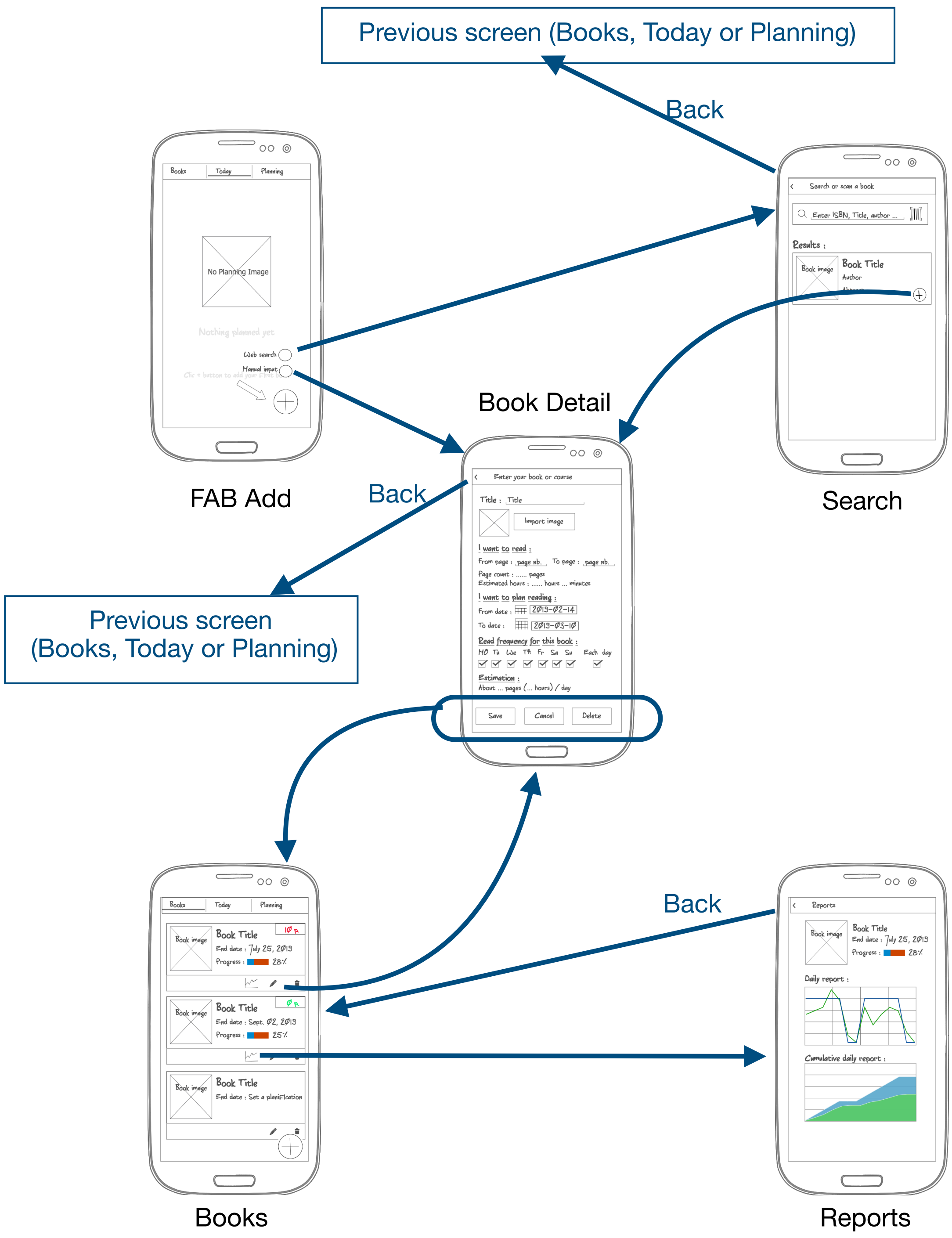
Back returns to the list of Books (**Screen 12**)

Screen 17 : 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.

Screens Kinematic :



Key Considerations

How will the app handle data persistence?

The app will use Room with LiveData and ViewModel.

Description of edge or corner cases in the UX.

- Test 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 Book Detail -> return to tab Books OR screen search
  - 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 Screen Today inputs : pages read and time are numbers.
- Manual ISBN input is algorithmically tested before search.
- Error message when an ISBN scan or input is not found.

Description of libraries we'll be using and why

- **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.

Description how we will implement Google Play 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

Task 1: Project Skeleton, first UIs

- Create a project with libraries required.
- App theme extends AppCompat
- Create skeleton UIs for screens Books, Today and Planning (3 layouts and 3 fragments) with AppBar and Toolbar.
- Test swipe and selection between 3 Tabs
- Create FAB and its sub-menus
- Test BAB and sub-menus
- Create a skeleton for UI Book Detail : Field Title
- Test navigation

Task 2: Implement Database

- Implement Entities skeletons, DAOs and Database for Room
- Test save, modify, delete a book with UI Book Detail
- For save, limit number of books to 10
- Implement setting : average reading time

Task 3: Setup LiveData and ViewModel

- Add libraries required and setup LiveData and View model
- Test save, modify, delete a book with UI Book Detail with LiveData
- Finalize UI Book Detail and Book Entity with all fields required
- Test date input : begin and end date >= than today

Task 4: Finalize UI « Books »

- Create RecyclerView, and item-layout for the list of Books
- Respect RTL specification
- Test UI Books shows relevant data

Task 5: Implement Google Books Service search

- Create the UI for search ISBN, create a Fragment for results
- Respect RTL specification in UI
- Use AsyncTask for requests
- Add libraries required and implement service classes
- Add a test for Internet connection
- Manual ISBN input is algorithmically tested before search
- Test requests to the service
- Error message when an ISBN scan or input is not found.

Task 6: Implement Google Mobile Vision Service

- Add libraries required for Google Mobile Vision Service
- Implement service classes
- Create Fragment for code scanner
- Finalize UI Scan ISBN with Fragment
- Test search by ISBN input and ISBN scan
- Test « Add book » button in a book result item opens screen Book Detail with relevant data

Task 7: Finalize UI « Today »

- Create all fiels for UI Today
- Test input data is correctly saved in DB
- When a task-book is done, it’s inactivated
- Number of read pages and time of reading must be numbers.
- Respect RTL specification
- Test back exists the app

Task 8: Finalize UI « Planning »

- Create all fiels for UI Planning
- Respect RTL specification
- Test output data is correctly uploaded from DB
- Test back exists the app

Task 9: Implement UI « Reports »

- Create all fiels for UI Reports
- Respect RTL specification
- Test output data is correctly uploaded from DB
- Test back returns in screen « Books »

Task 10: Implement Widget

- Create the widget
- The widget updates once a day
- When user clicks on the widget, screen « Today » is opened

Task 11: Implement « load image » in UI « Detail »

- Send an intent for selecting image
- Save image in db
- Resize size of loaded images

Task 12: Test last corner cases

Test 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 Book Detail -> return to tab Books OR screen search
- activated when in Screen Charts -> return to tab Books

Screen rotation keeps all similar.

Task 13: Implement Material Design Transition

- Create an element transition between screen Books and Book detail

Task 14: Implement Accessibility

- Describe contents
- D-Pad navigation

Task 15: Implement Notifications

- Each day, send a notification with the work to do

Task 16: Implement install release

- Signing, Keystore