How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: Pilpin

The Register

(not sure if this is how it's called in english, I'm french and was thinking of calling it "Faire l'appel" which I think means "do the register" or "do the roll call", anyway, I might keep the french name if it's allowed)

Description

Tired of doing the register on paper every day and for multiple classes ? Use The Register to keep it all on your phone and export it to google drive whenever needed !

With the register you can create multiple classes with as many students as needed, you can even take note of when students left the class. If you need to communicate this information to someone else, you can easily export the data to google drive as a csv file.

Intended User

Teachers and other school workers who have to make sure and write down if kids are present or absent.

Why this idea - My mother takes care of children at school but after the classes end, she's supposed to do the register at the beginning of her "shift" and she has to write down when each kid leaves, she then has to send this data once a month to the school in a spreadsheet.

Features

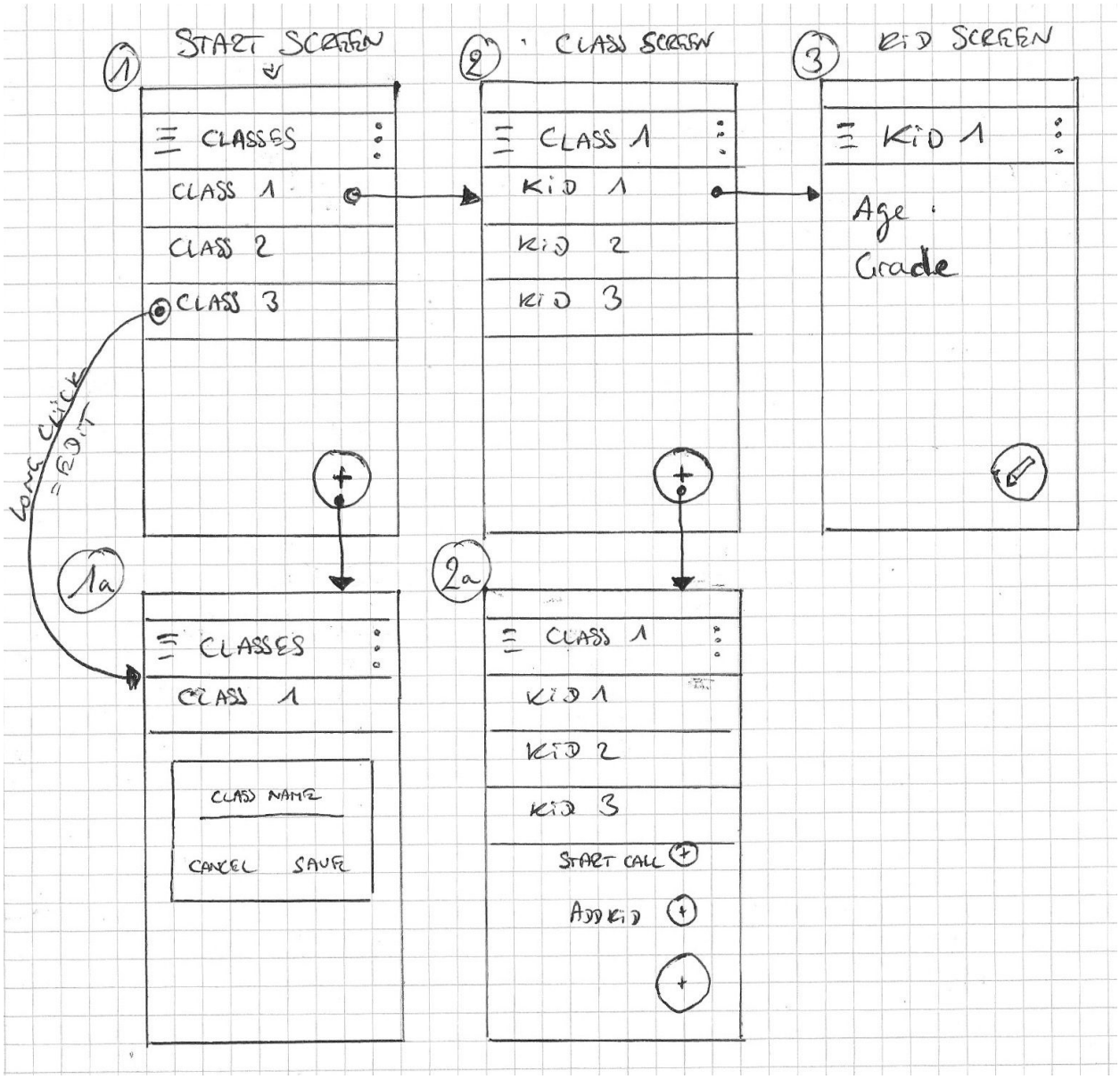
The app allows you to:

- Create kids (name, first name, age, grade)
- Create classes or "groups of kids"
- Note if kids are present or absent
- If needed, note when they left (date and time)
- Export the informations in google drive as a csv file

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screens



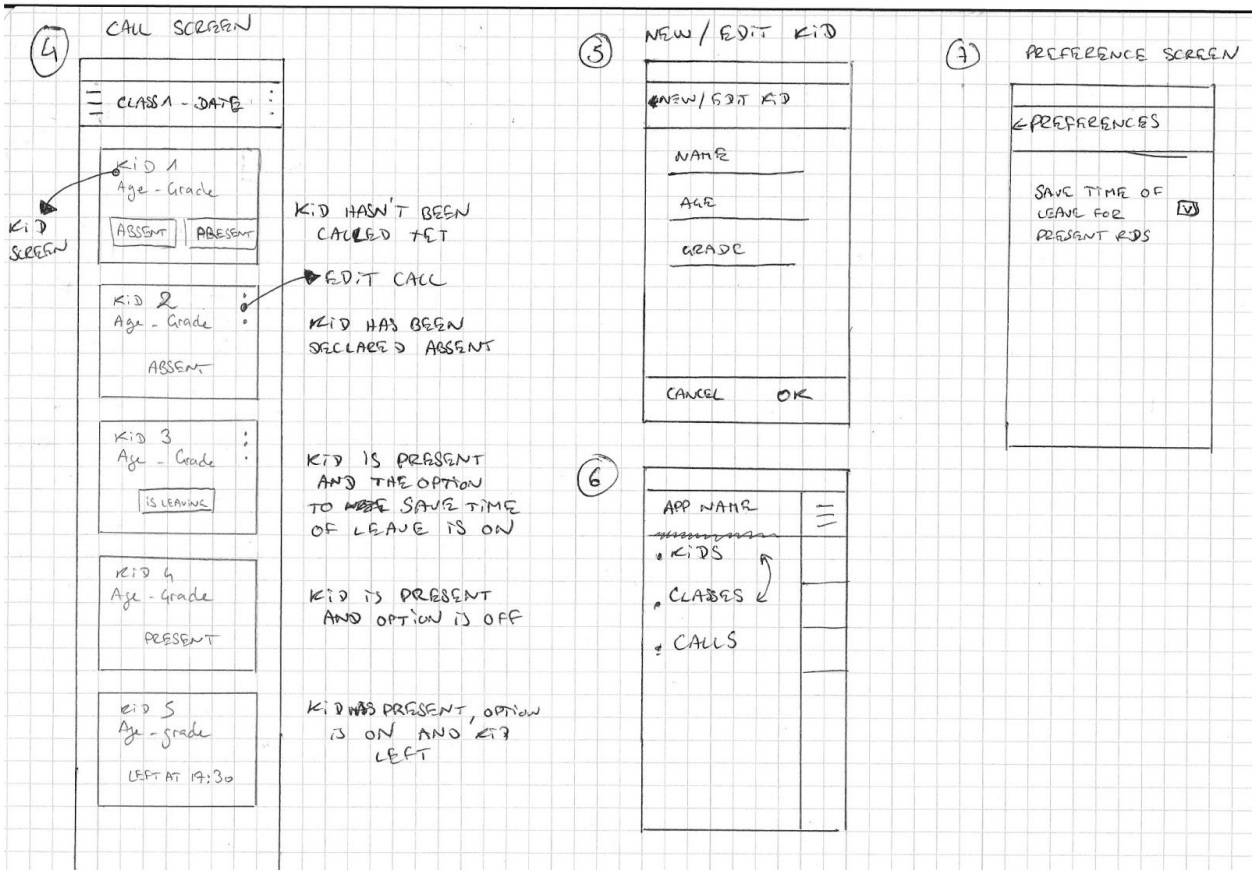
Screen 1 - Start Screen - Classes List Screen
Displays the differents classes the teacher may have to “do the register” for.

Screen 1a - New Class dialog
Allows the user to create a new class.

Screen 2 - Class Screen
Displays the kids who are in the class. From there you can either add a kid to the class or start to do the register by clicking on the “+” action button, you can also remove a kid from a class by long clicking on the kid’s name.

Screen 2a - Add Kid and “Start Call’ or “Start Register” action buttons
Considering perhaps moving Add kid to the overflow menu because it’s not an action that’s often used throughout the year (only at the beginning)

Screen 3 - Kid Screen
Displays the informations we have on each kid.



Screen 4 - “Call Screen” / “Register Screen”
Displays a (past or present) call, basically a list of kids displayed in cards containing the info on each kid and at first two button “Present” and “Absent”, if the user turned on the option to record the time at which every kid left, after clicking present, a new button replaces the two. The user can edit each “kid’s call” by pressing the three dots on the card.

Screen 3 - New/Edit Kid
Basic form.

Screen 6 - Side menu
Allows the user to jump between the list of kids, the list of classes and the list of calls

Screen 7 - Preference Screen
To turn on or off the “record time of leave” option.

Screen 8 - Calls list (not drawn, extremely similar to screen 1)
Every call is named as “class called - date of call”

Key Considerations

How will your app handle data persistence?

The app will store everything on the device using a database and a content provider.

Describe any corner cases in the UX.

Different button interactions described in the mockup section. Back button always brings the user back to previous screen.

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

Butter Knife
I’ve never used this library before but it looks like it’s going to make binding view and changing their properties easier than before, the less time I spend on this, the more I can spend on some other key areas of the app.

Describe how you will implement Google Play Services.

Google Drive - used to export the data in a spreadsheet, as a csv at first, as google drive allows users to turn csvs in sheets, is export is done automatically everyday.
Firebase Report Crashes - curious as to how it works and how much data I can gather after a crash.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally 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:

- Start new project
- Add libraries to the project
- Configure libraries if necessary
- Configure app signing

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for ClassesListActivity
- Build UI for KidsListActivity
- Build UI KidDetailsActivity
- Build UI for KidEditActivity
- Build UI for CallsListActivity
- Build UI for CallActivity
- Build UI for call cards
- Build UI for Preferences
- Build UI for side menu

Task 3: Content Provider

- Create the database
- Create the data provider

Task 4: Activities & Fragments

- Write each activity or fragment code to display data from the provider

- Add code to add and remove data from the provider and bind it to the proper UI elements

Task 5: Google dependencies

- Create the asynctask allowing to write files on user's google drive
- Add calls to the asynctask where necessary in the activities/fragments
- Add Firebase Crash Report calls.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"