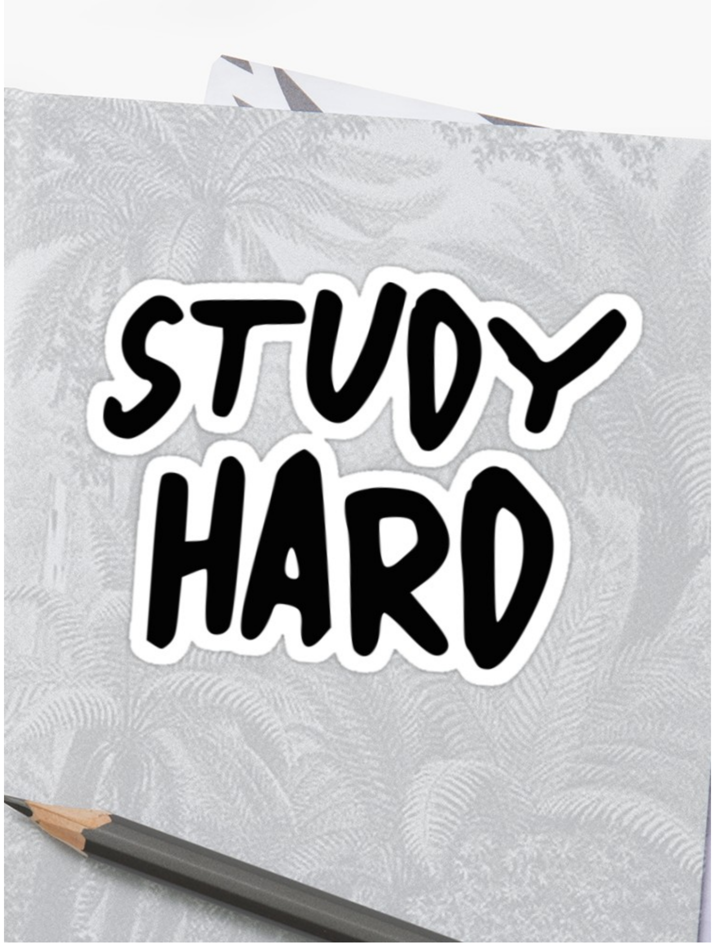
CHAPTER EIGHT : POWER FOCUS PROJECT OVERVIEW



I. INTRODUCTION

With all the knowledge from the previous chapters, from this chapter on, we are going to build our first application! If you still have any doubt about the contents of the previous chapters, we will demonstrate them in a different context. The best way to learn iOS development is to try things out yourself. You will make much more progress and gain experience building your own application from ground zero. Now it's time to fire up your Xcode and get started!

II. FUNCTION OVERVIEW

The app that we want to build is a simple time management app. It loops a work-break cycle to remind you how much time left for work and break. More specifically, the timer alternates between work and break, when the countdown goes to zero, it notifies you to change your activity and starts the timer of the next activity.

Here is a list of the main features' descriptions:

- Countdown during each activity (work, break, long break)
- Notify user at the end of an activity and go to next activity
- Go to long break after several cycles
- Allow users to modify the activity length
- Allow users to modify the notification (sound, vibration, banners)
- Allow users to modify long break frequency
- Display the number of completed cycles

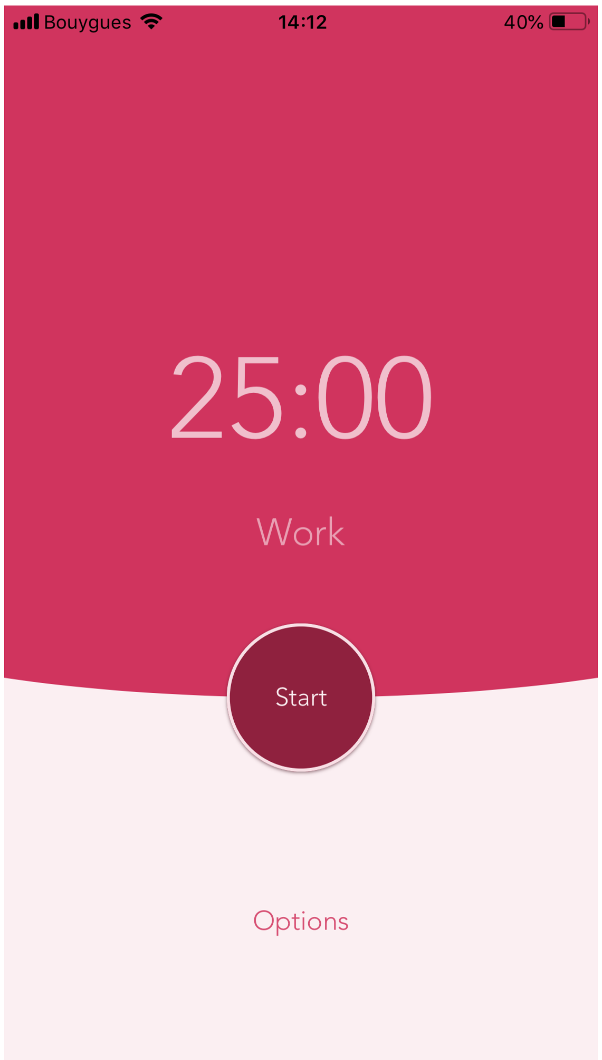
In our case, our application is not that big and relatively easy to define and specify. This is a simple version of a preliminary requirement and specification document. For bigger applications, this can be rather complex and hard to define. It contains many use cases and many complex sequences. But whatever the size of the application is, its purpose is always to give development team an overview of the application. If you already have another app idea in mind and you want to start your own app, this first step is highly recommended! If you start with a clear and well designed goal in mind, your development process will be much more pleasant and it will be very unlikely to get stuck somewhere (because of the bad design). If you want to save some time in the long run, this is one of the most important step you can't skip.

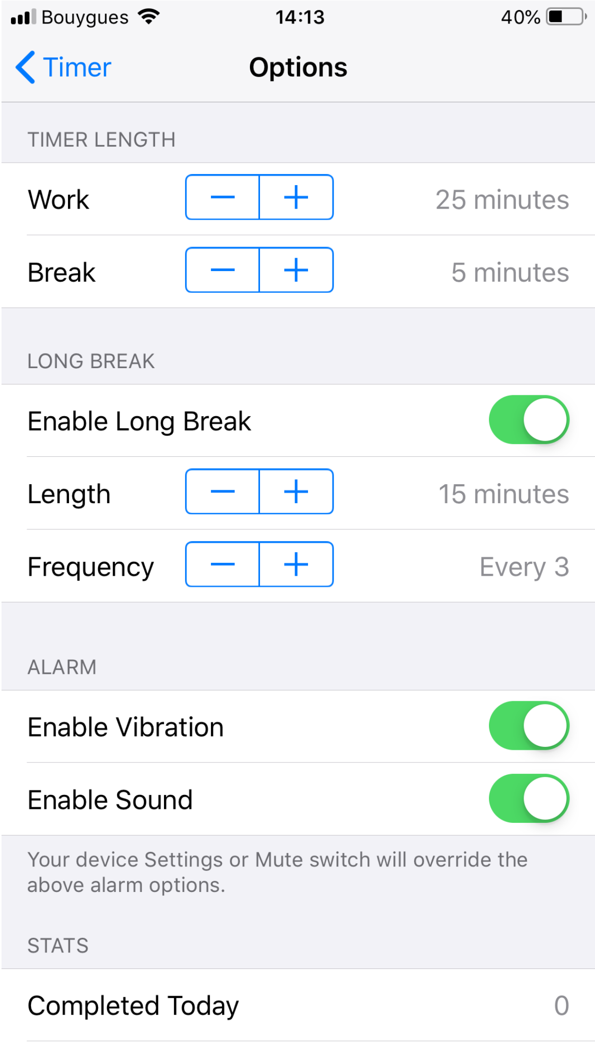
This is not an exhaustive list of all the features of our app, but it describes the basic functionalities of the app. Just like any software development process, this list can evolve and grow during the development.

III. INTERFACE AND PROTOTYPING

Prototyping is a great way to design the first version of the app's UI. There are many awesome prototyping apps on the market, and here I recommend Adobe XD for prototyping. The main idea about prototyping is to get an overview of the UI fast instead of spending a lot of time working on storyboards directly. Adobe XD gives you many convenient "fake" UI components to build an UI without really building an UI. If you have more than a dozen screens in your app, this can save you a large amount of time.

Here are the prototypes of this app:





TIMER LENGTH

Work − + 25 minutes

Break − + 5 minutes

LONG BREAK

Enable Long Break ●

Length − + 15 minutes

Frequency − + Every 3

ALARM

Enable Vibration ●

Enable Sound ●

Your device Settings or Mute switch will override the above alarm options.

STATS

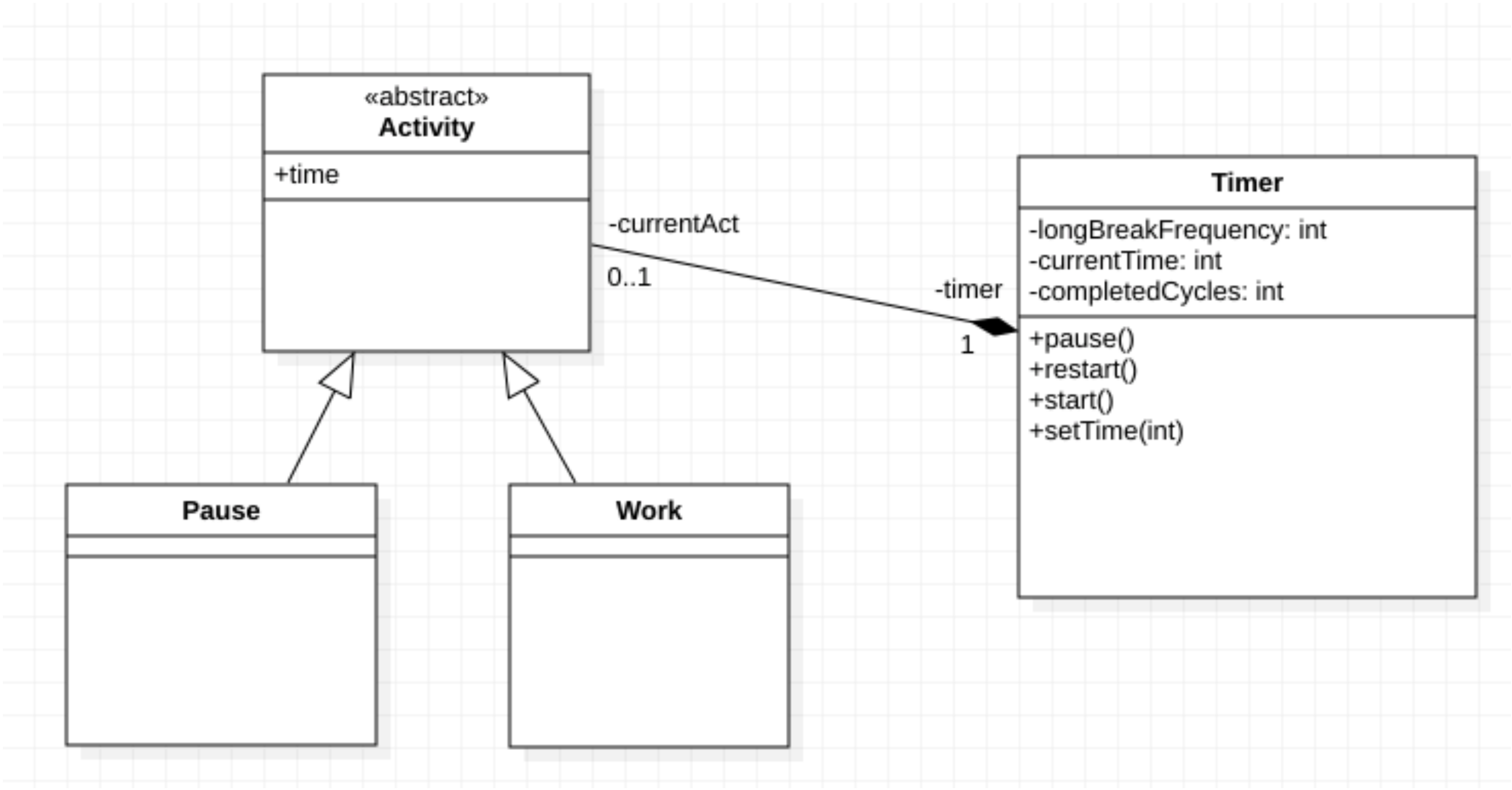
Completed Today 0

The main screen contains the countdown, the activity's name (work or break), and the buttons to start, pause, and restart. By clicking options, users can access a table view to set up options and view info.

IV. DATA STRUCTURE

The next step after having a prototype is to think about the data structure of the app. If you are not familiar with the term "data structure", you can ask yourself this question: What am I going to present in the GUI? This is also the "M" (model) of the MVC architecture. In large applications, the model can be huge and very complex. For instance, a social media app has to store and organize all the accounts, posts, lists of friends and groups… This part is usually the "core" of your application and it always should be UI independent, which means you can migrate this into a completely different platform without changing a single line of code in your model.

In this class diagram, we have two classes: Pause and Work that extends an abstract class Activity. An activity has a integer time and belongs to zero or one Timer. A Timer has zero or one current action of type Activity (which can be a Pause or a Work). A Timer object also has to know the long break frequency, and the completed cycles to determine the next activity.



This data structure design is not unique. You can think of your own data structure design with different classes, methods, and attributes. The difference between a good and a bad design is the difficulty of the implementation and the efficiency. Maybe some of the attributes and associations don't make sense to you right now or you want to do it otherwise. I highly encourage you to follow your own design instead of mine. Your design should always cover all the use cases and all the features, but how exactly? It is totally up to you. It requires some experience and it might take some time to come up with a great design. But I can assure you, it worths every effort! The last thing you want to do is to change your design in the middle of the implementation, which leads to inconsistency and potential bugs.

SUMMARY

In this chapter, we have worked on these topics:

- overview of the Power Focus's main features
- design the app's GUI
- create a prototype of the app
- model the data structure
- create a class diagram

If any of these is unclear to you, please make sure to go back and read the related part or parts before moving on the the next chapter.