Multi-timer App

by Steven Hsieh for Wilson Fletcher

Hello Mark, Stephanie and team.

Thanks for the opportunity to meet both of you and get an understanding of the team and the great work you do.

Originally I had planned to use time over the weekend to complete this task due to my working schedule conflicting with other responsibilities.

However, I was excited after speaking to you both and decided that it was worth sending something over, even if it meant juggling a few tasks around.

There are more visual directions I would've liked to experiment with, and certainly more functionalities to think about.

However, I hope this'll give you a little more understanding of my creative and problem solving processes.

Please enjoy this as much as I did undertaking the task.

Cheers
Steve

After reading the brief, I wanted to understand the complexity of multi-timer interfaces.

I quickly discovered a couple of key problems I needed to solve …

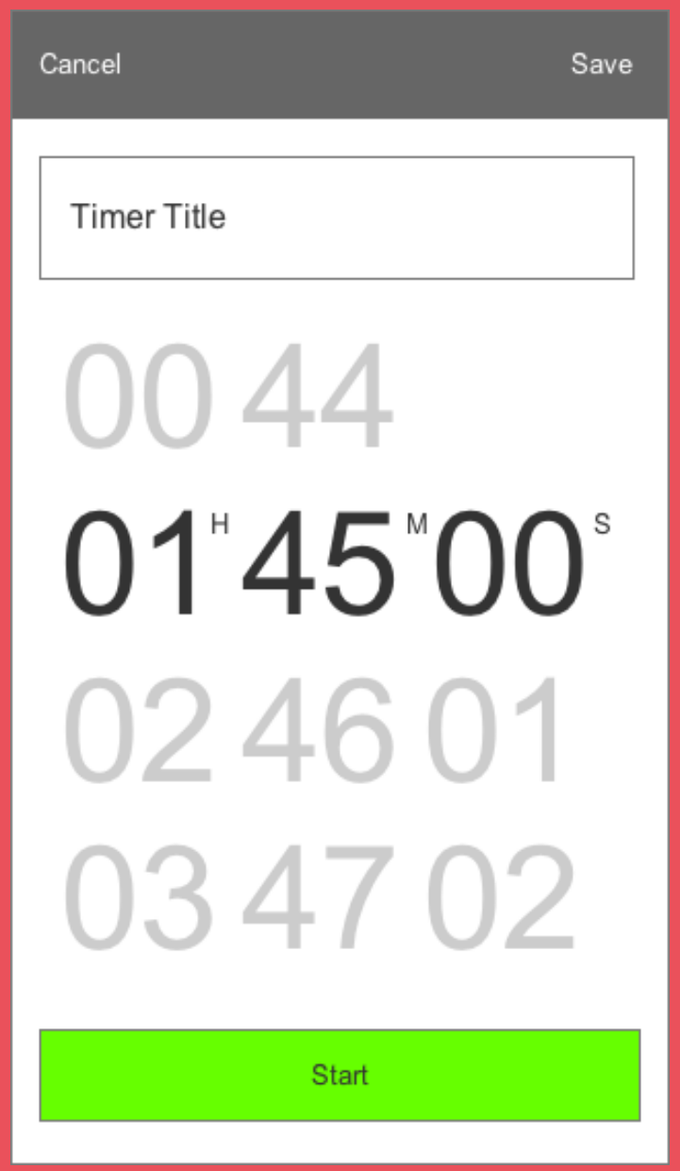
How can independent timers exist alongside ‘grouped' (relative) timers?

How to control the different ways 'grouped' timers behave?

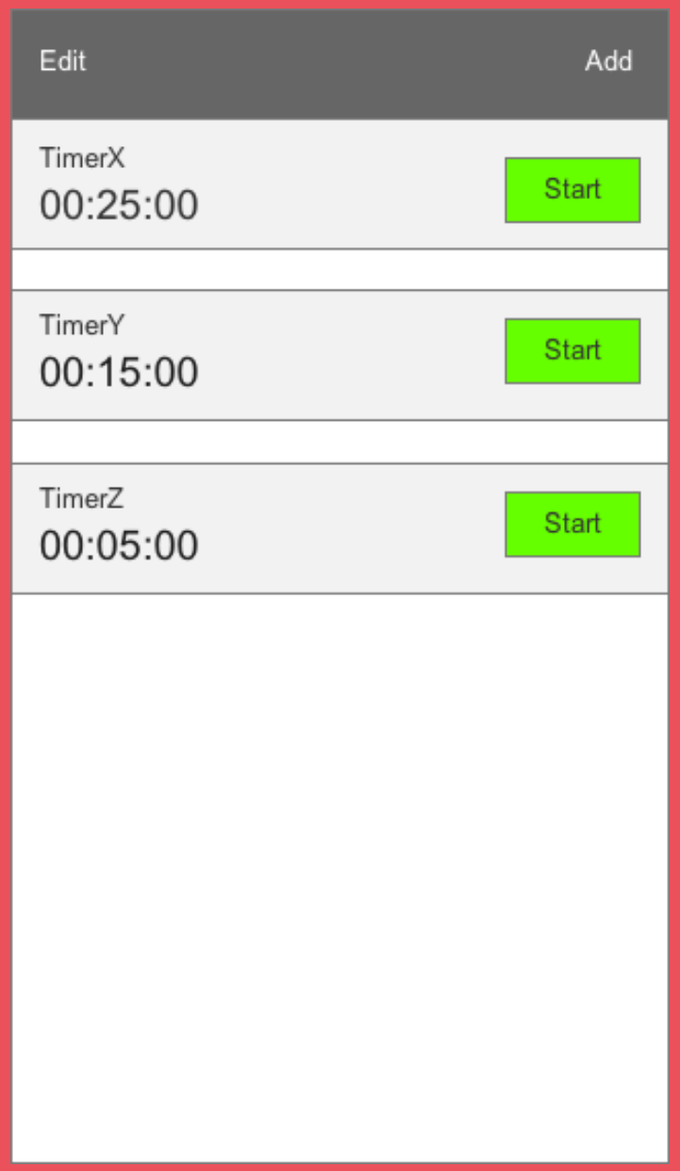
How can we use the watch to complement this app?

Using Axure, I quickly sketched out my thought process, starting with the basics.

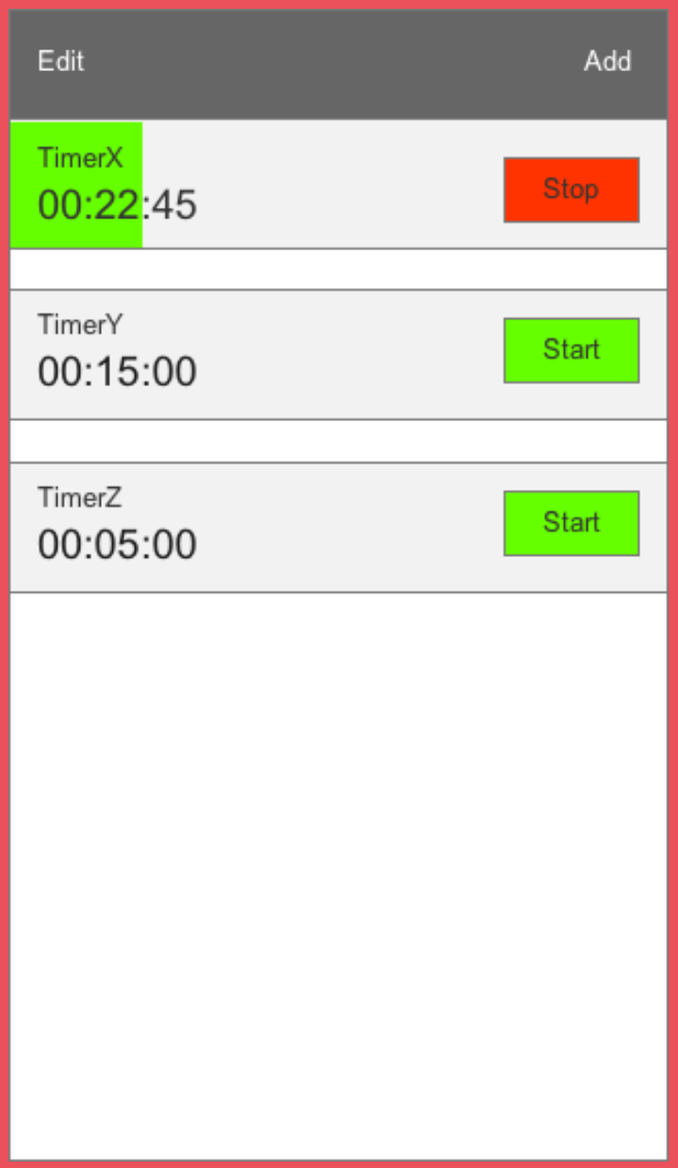
We know that adding and setting timers will be a common feature.



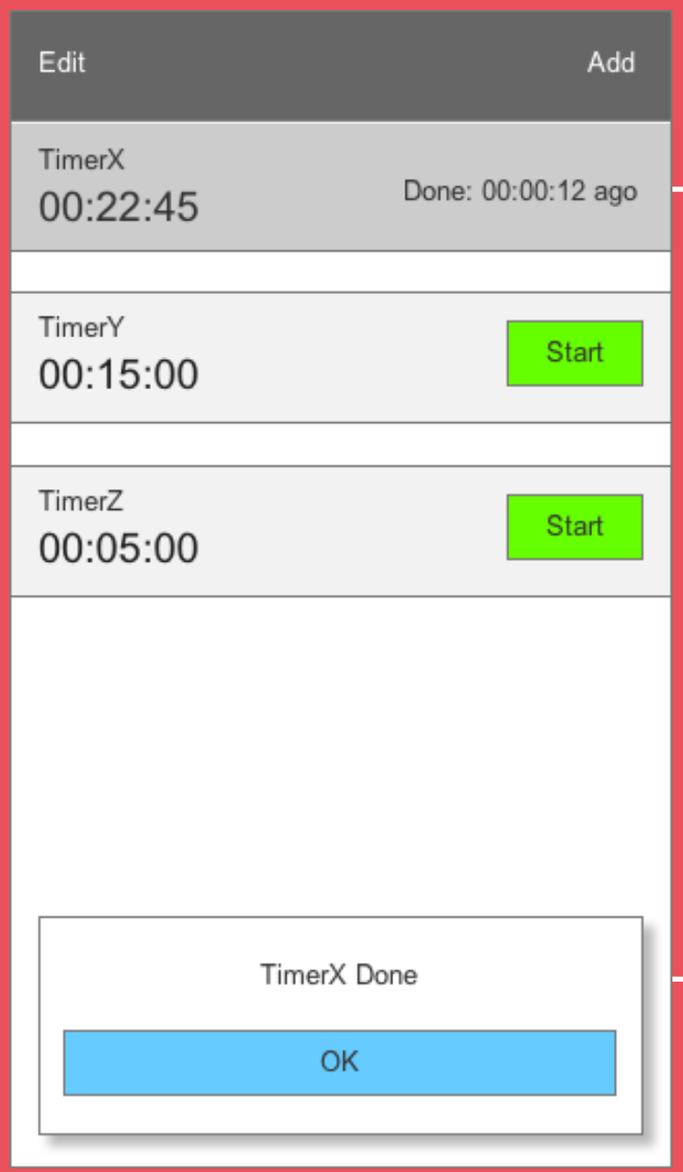
It’s also straightforward to have multiple, independent timers set up.



These timers can be started individually, and won't affect each other.



TimerX starts.



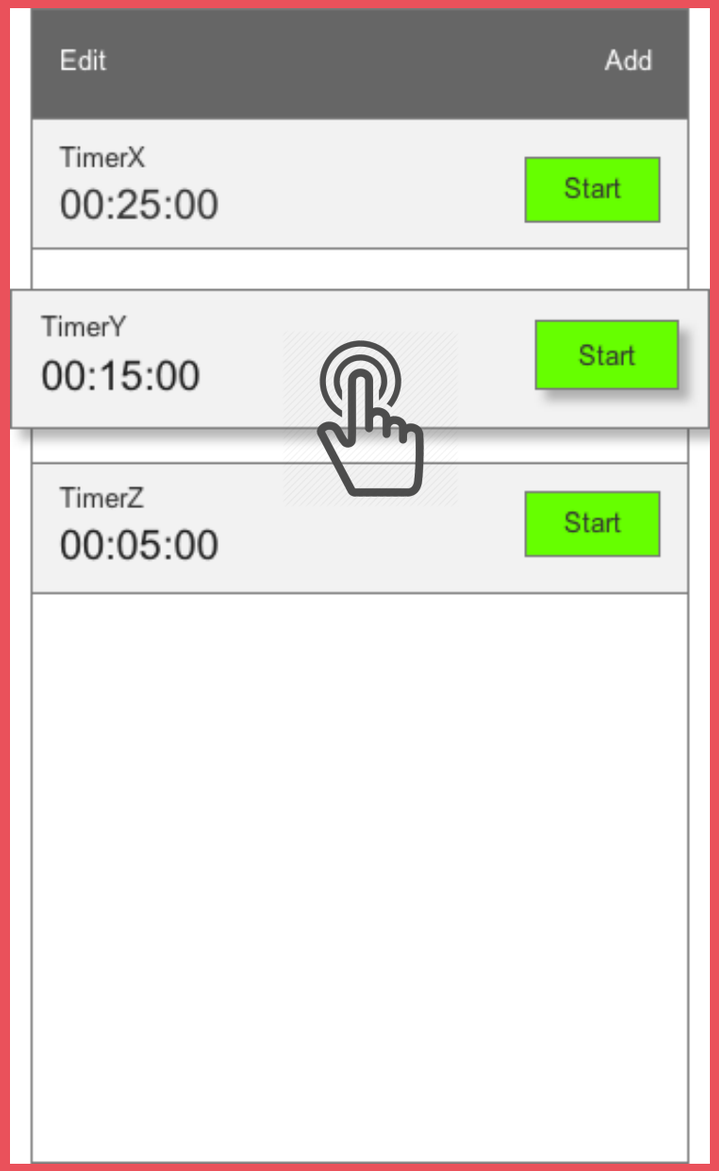
TimerX ends with alert.

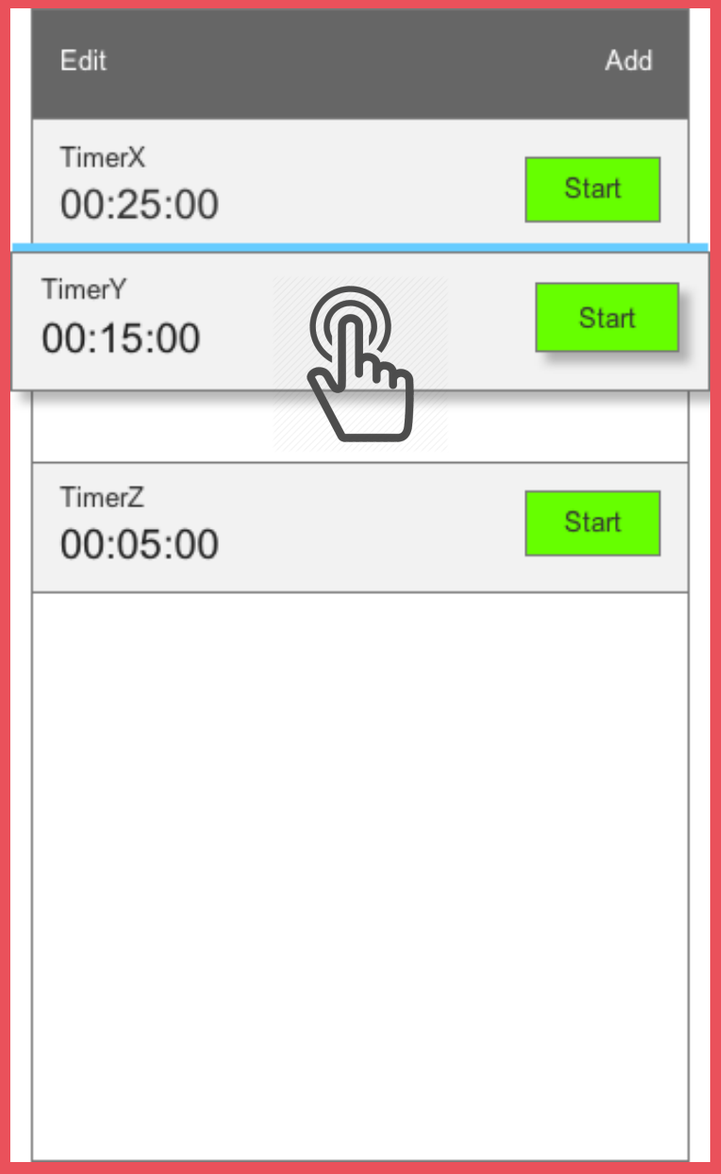
Helpful notice to accompany 'finished' state.

Hey! These alerts will be useful on the watch.

So far so good. But how do we group timers that are ‘relative’ to each other?

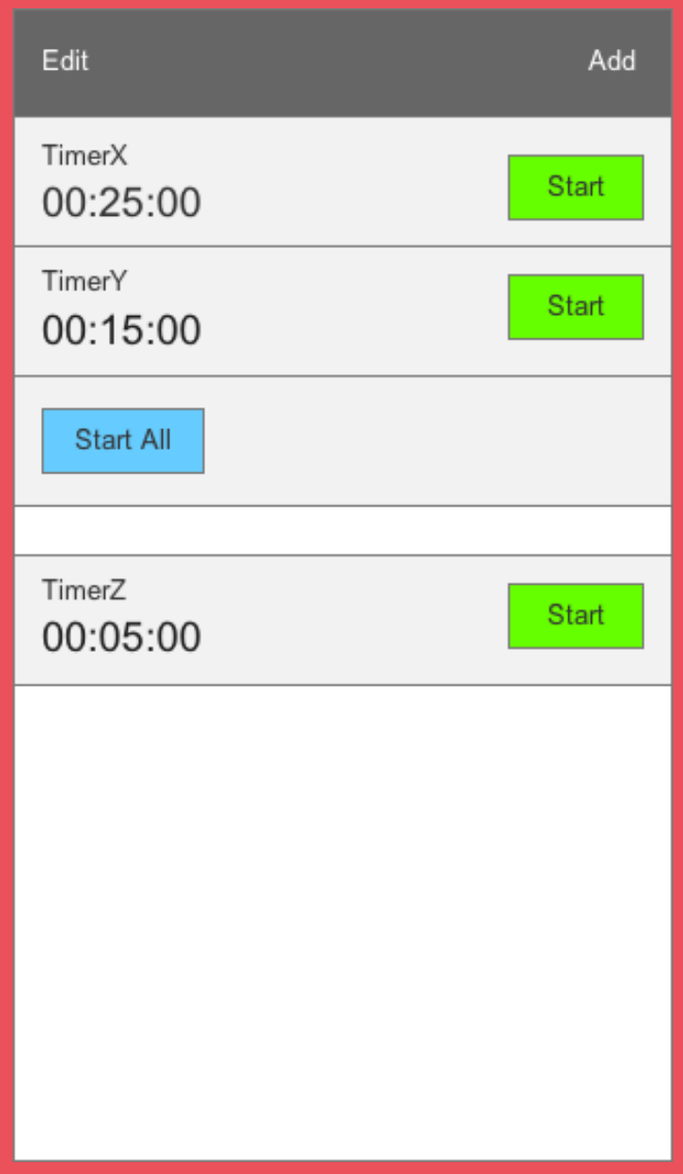
An intuitive way could be to simply click, hold and drag items to ‘snap’ with each other.





Hold & drag functionality to ‘group’ X & Y.

The same function can be used to reorder and breaking ‘groups’ apart.



TimerX is now ‘grouped’ with TimerY.

Extra controls for ‘group’ appears. This is unique to groups only.

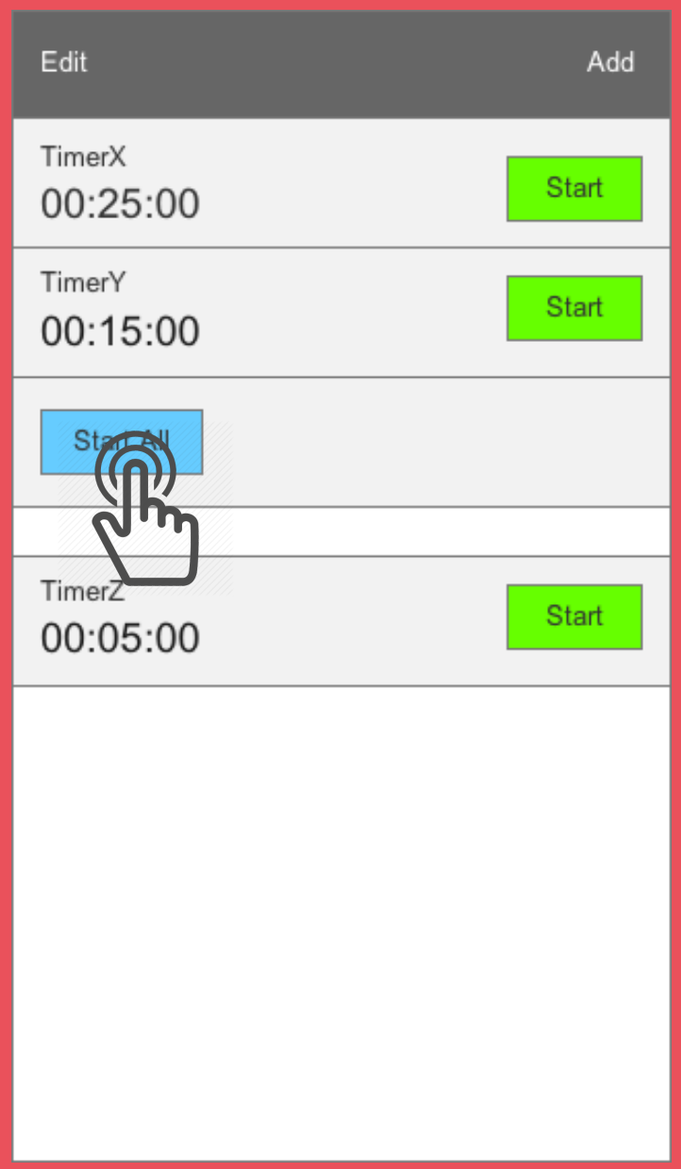
So now we have the ability to ‘group’ timers. What can the user do with these groups? Let’s think about some user cases.

For cooking and baking, timers need to
end together.

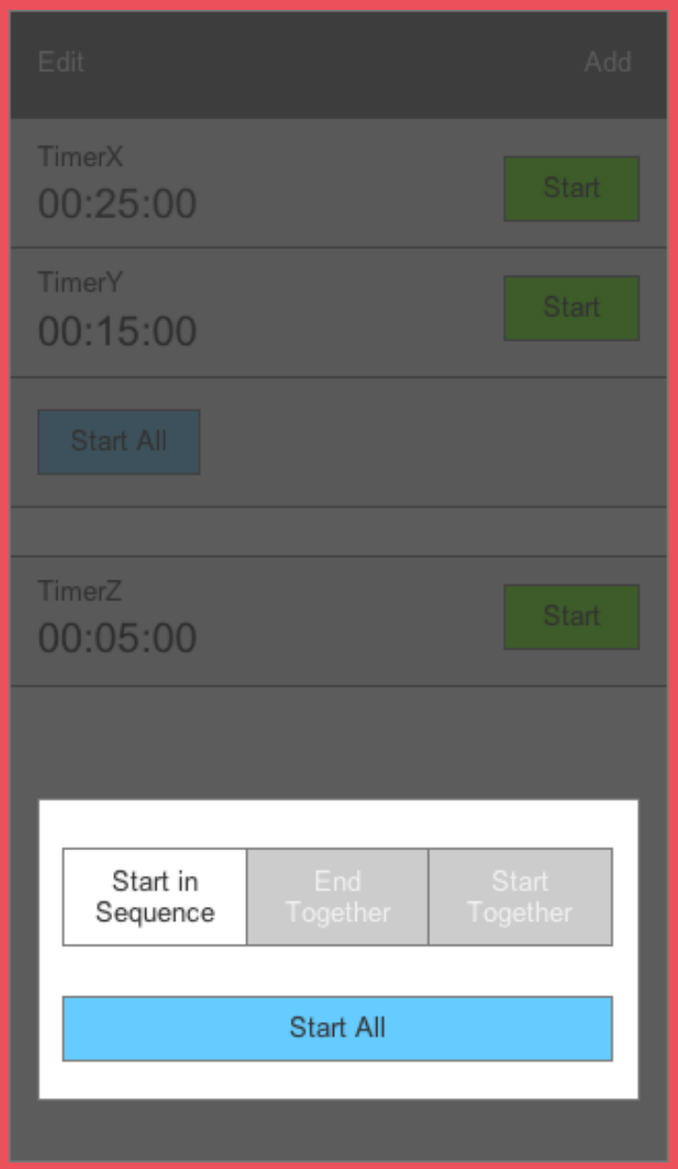
For interval training and running workshops, timers need to **start in sequence**.

For tasks of different lengths but start together, timers need to … well, **start together**.

Let’s give users these option during the ‘start all’ call to action.



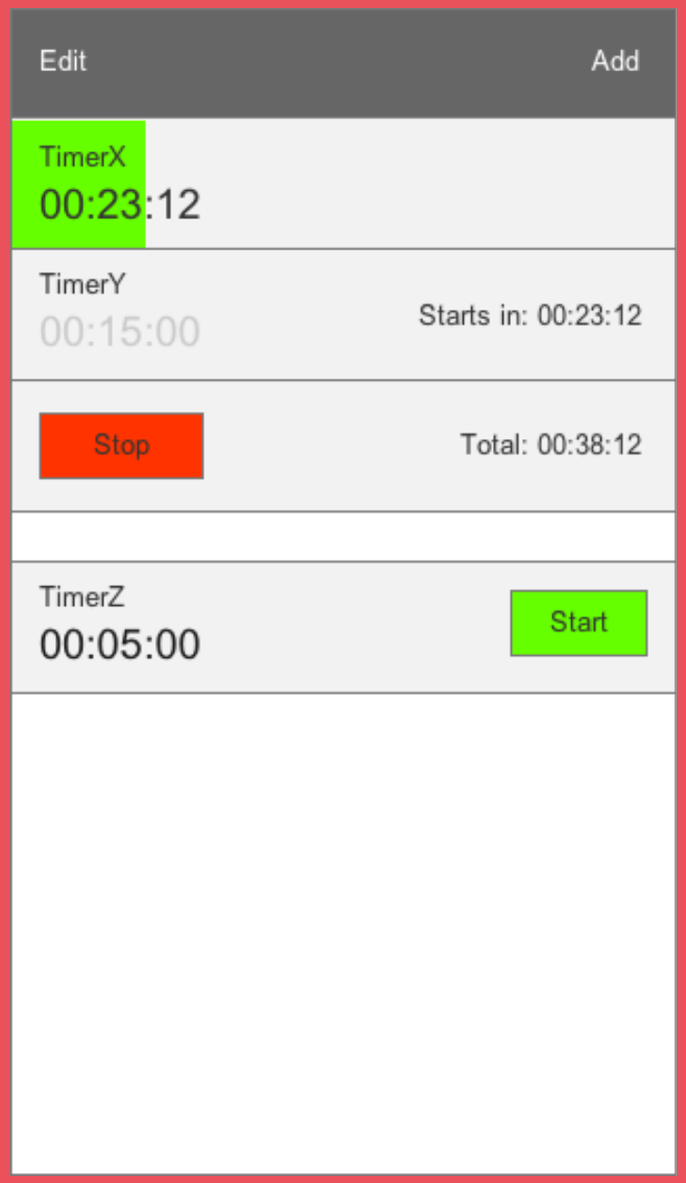
When ‘group’ CTA is clicked.



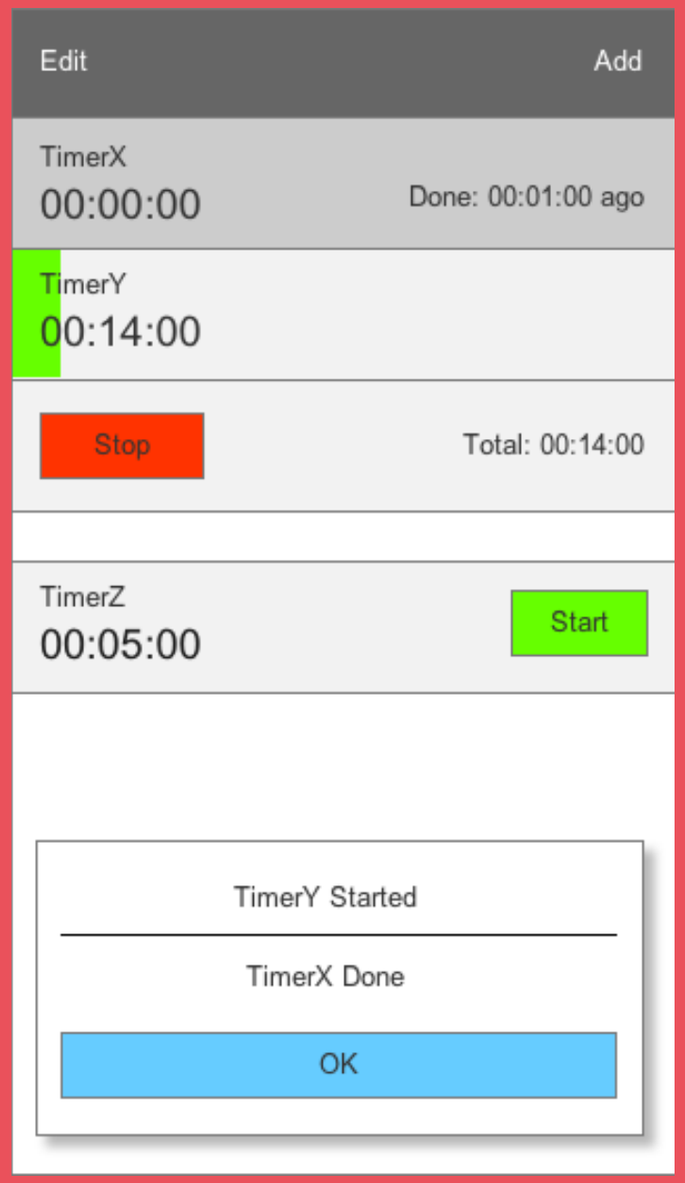
Options appear where users can select how they want timers to interact with each other, before starting all timers.

Now let’s see these three options in action.

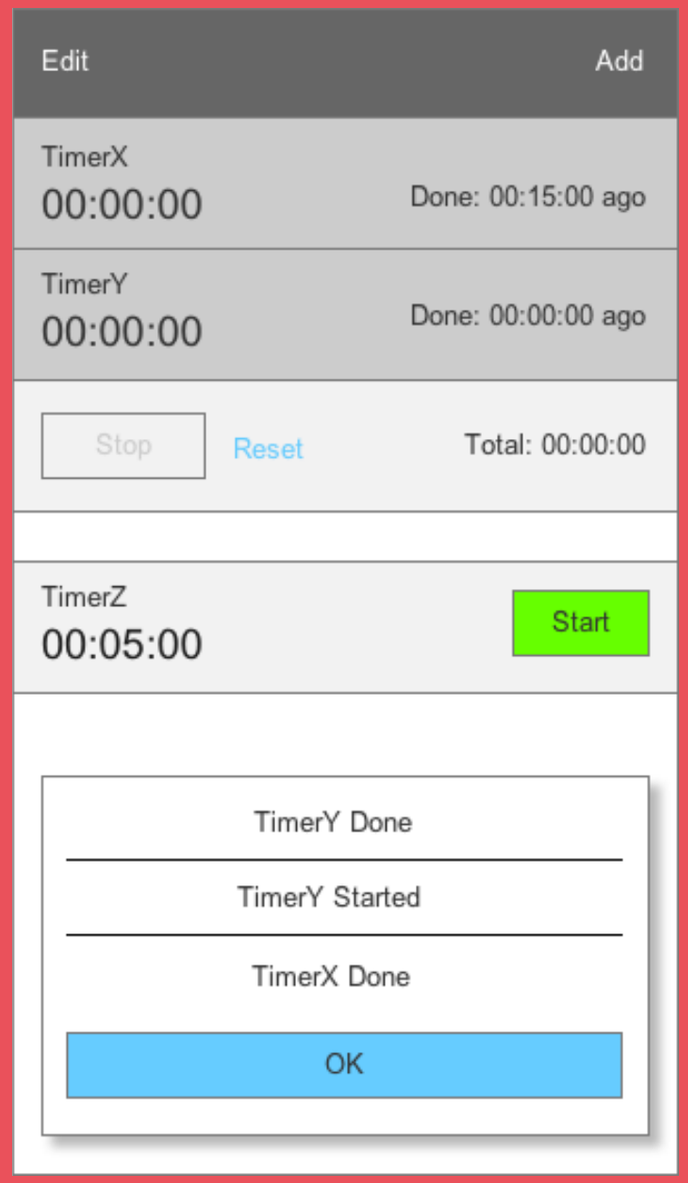
1 - Starting in Sequence



TimerX begins. TimerY waits.

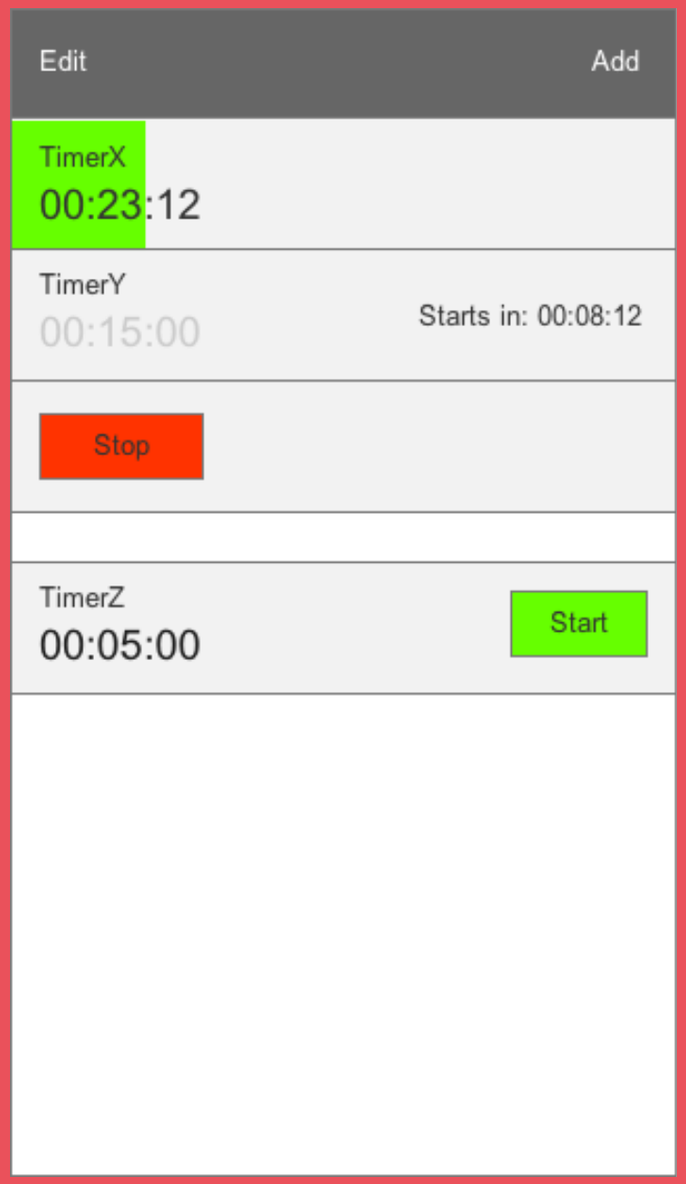


TimerY starts when TimerX ends.

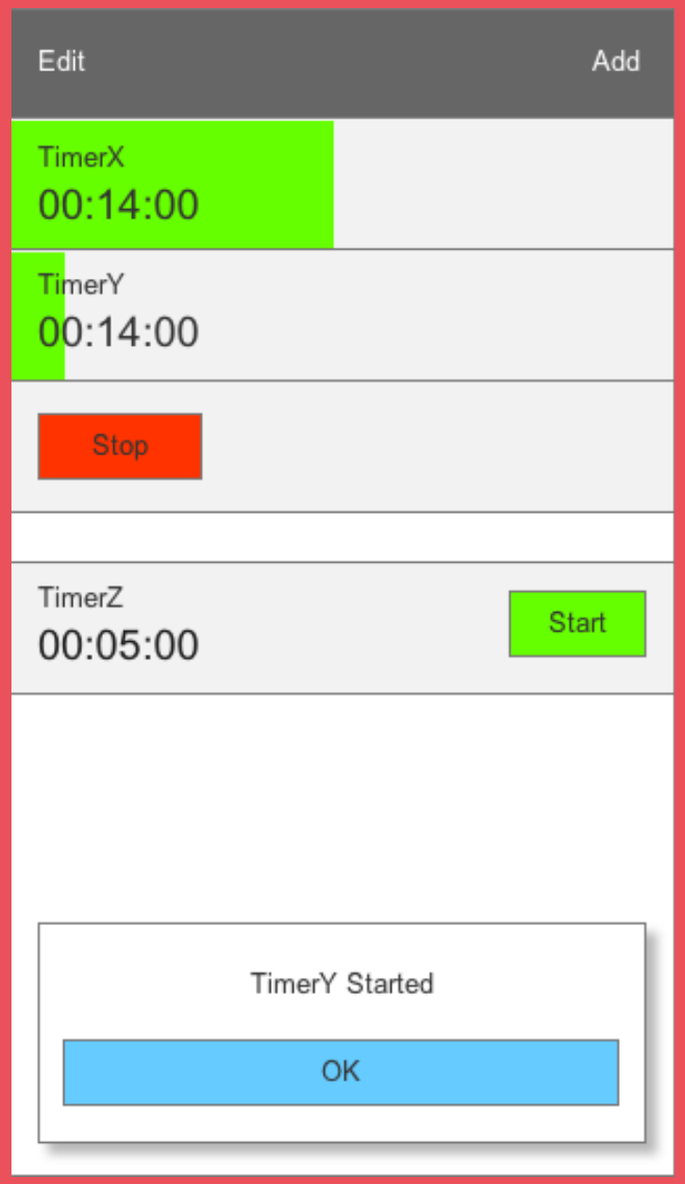


Time Y ends.

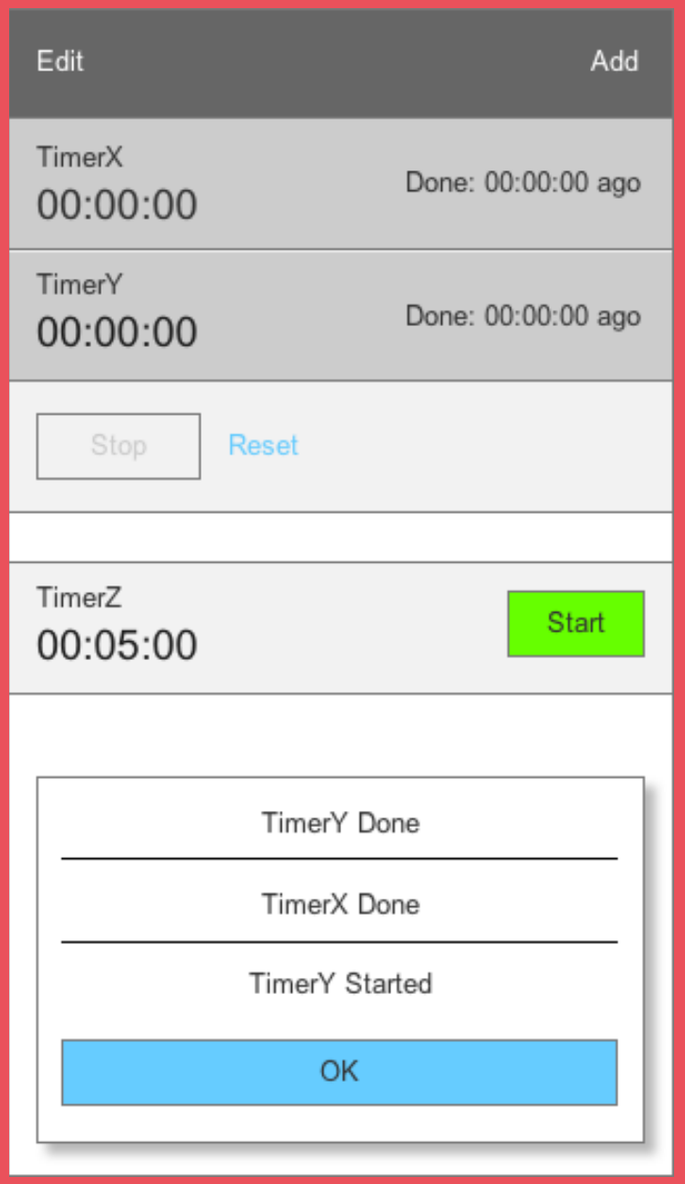
2 - Ending Together



TimeX starts first. Timer Y waits.

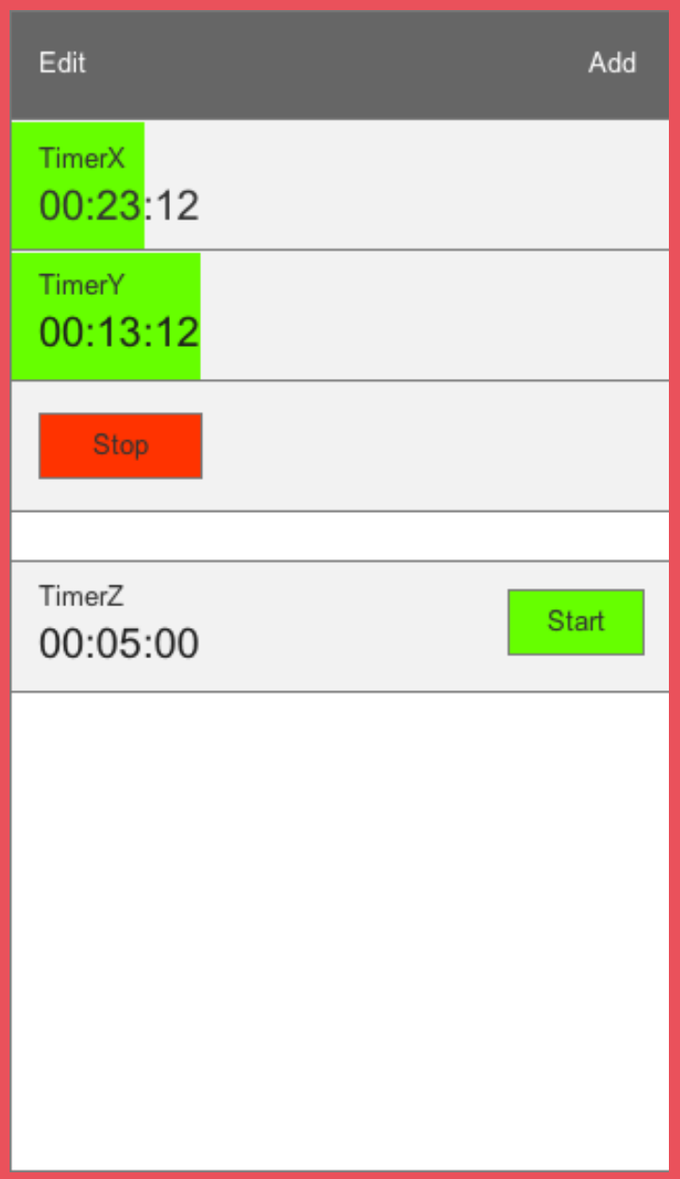


TimerY starts when TimerX hits 15:00.
This is calculated automatically.

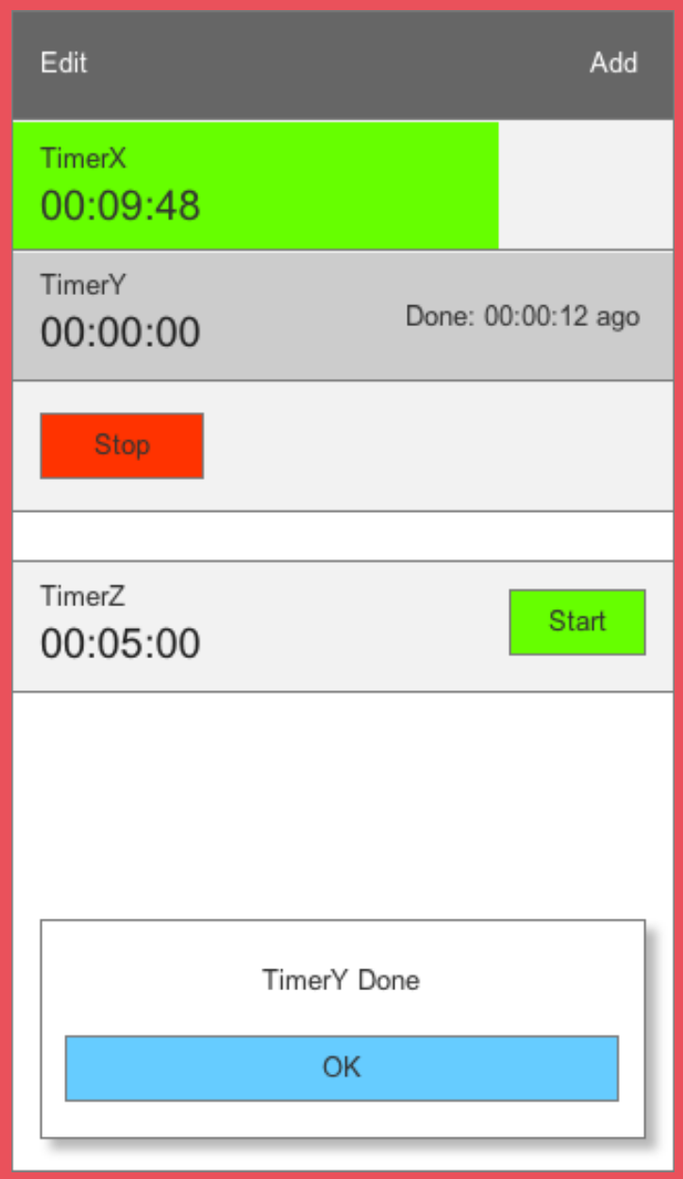


Both timers end at same time.

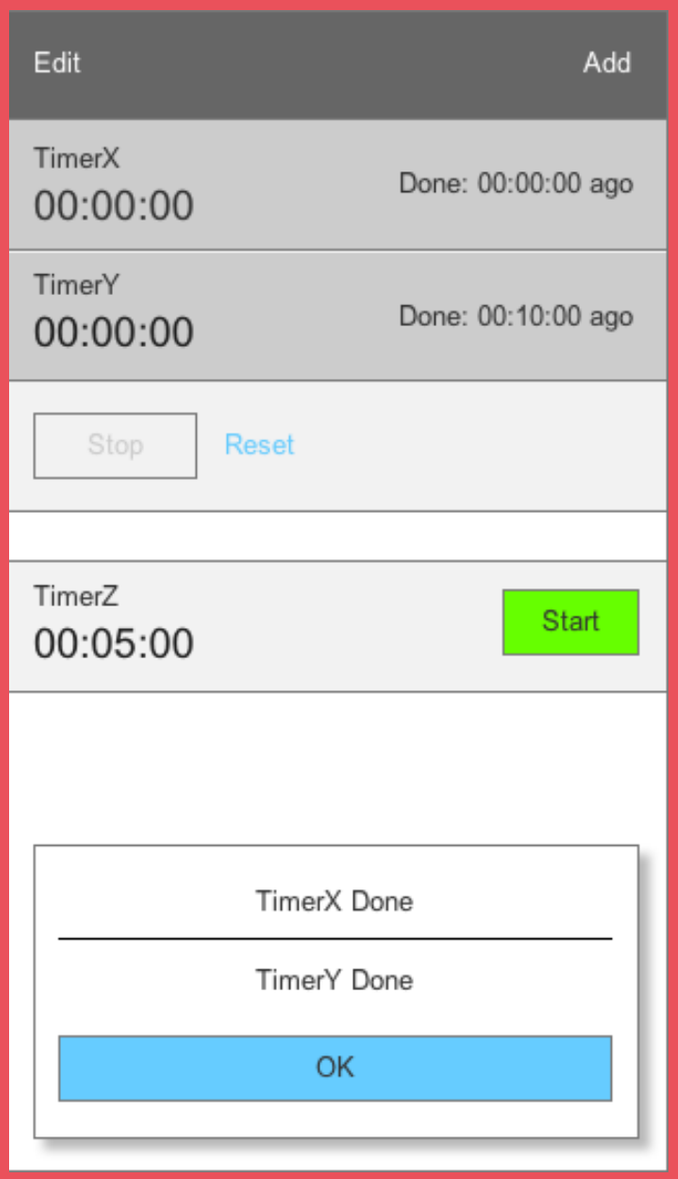
3 - Starting Together



Both TimerX and TimerY start at the same time.



TimerY completes first. TimerX continues.



TimerX completes next.

Why didn’t I use a gantt chart or another visual language to display the timers?

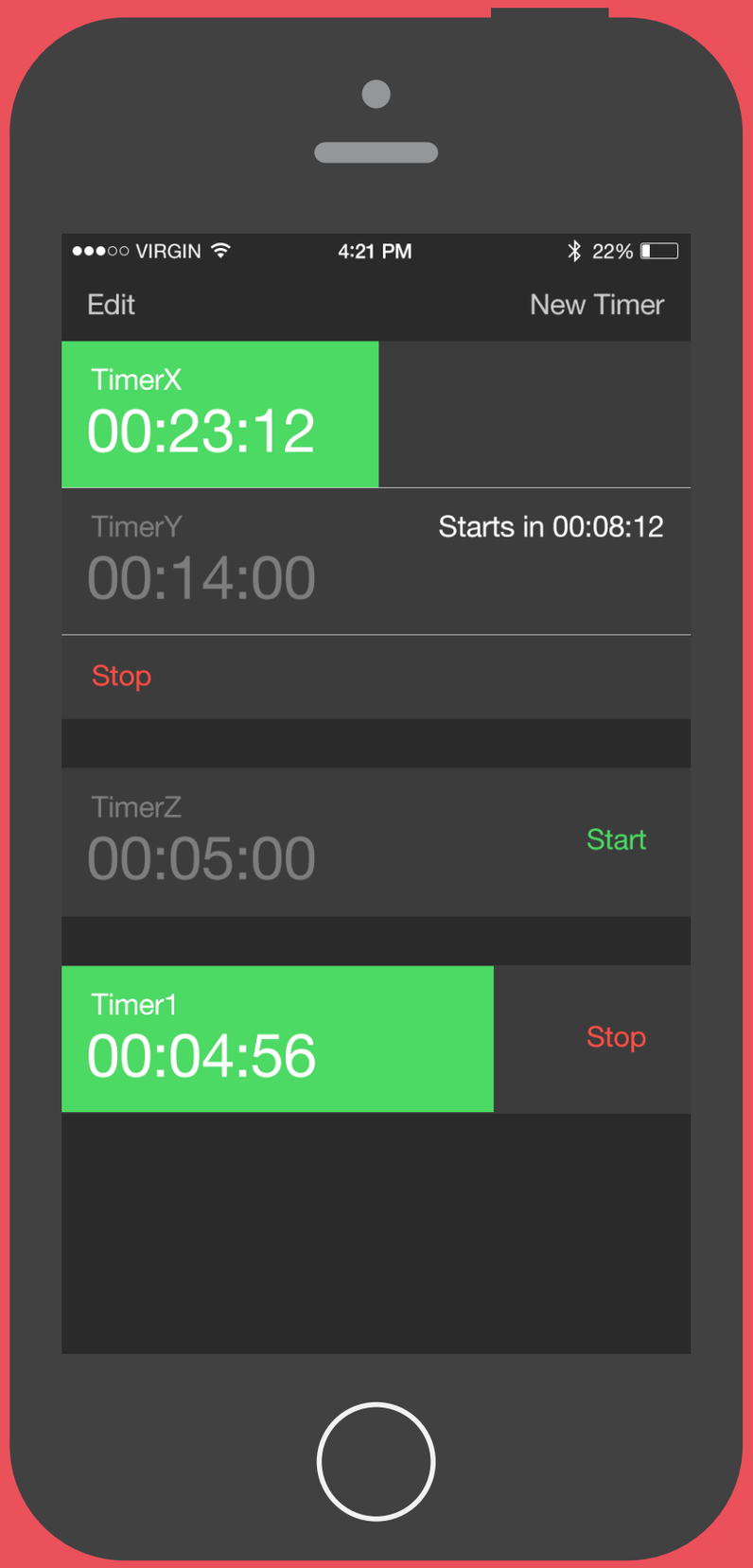
With the restricted real estate, readability becomes an issue very quickly once you have multiple timers working.

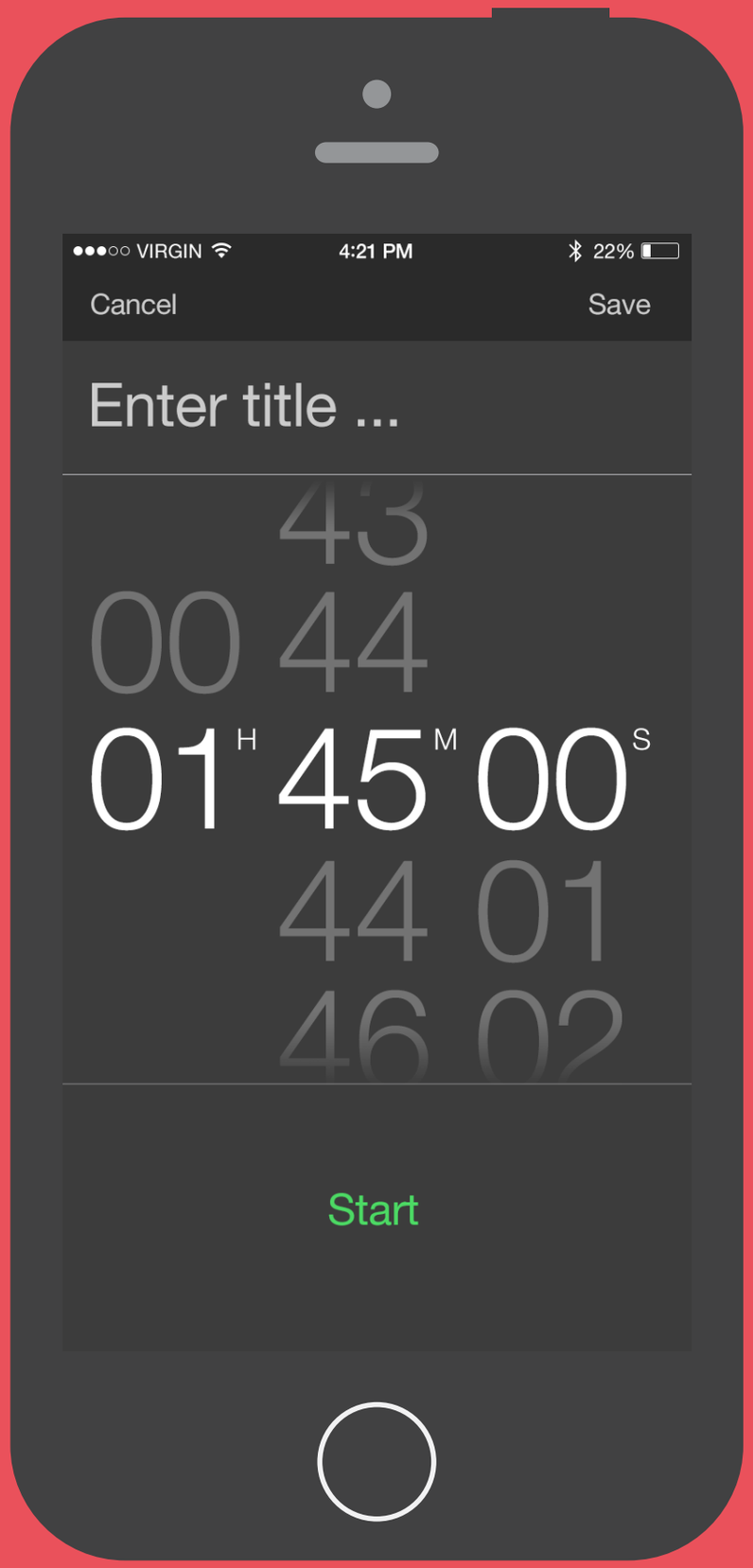
I didn’t want users to have to scroll more than they needed to view relative timers, next to other independent timers.

Now that we have the basic core features working,
let’s take a look at design.

Being a generic multi-timer, we need to distance ourselves from any single theme or colour.

It needs to be intuitive and delightful - so choice of colour for is still important to higlight CTAs and create visual hierachy.





With the visuals, I decided to go for a dark theme.

It's easier to use different shades of greys to create visual hierachy, and colours tend to pop more on darker backgrounds.

The colour palette is neutral, aside for highlights that need to draw attention to the actions.

Clean, simple typography keeps the theme generic and can be used by a variety of use cases.

One more thing …

The status updates in the wireframe lends itself really well for the watch. It allows users to move around while staying on top of alerts.

Perfect.

Thank you