

浙江大学

本科实验报告

课程名称: 移动平台技术开发

姓 名: 付嘉欣

学 院: 计算机科学与技术学院

专 业: 软件工程

学 号: 3190105562

指导教师: 章国锋、陶煜波

2022 年 3 月 30 日

浙江大学实验报告

课程名称: 移动平台技术开发 实验类型: 软件开发

实验项目名称: Clock

学生姓名: 付嘉欣 专业: 软件工程 学号: 3190105562

同组学生姓名: 无 同组学生学号: 无

指导老师: 章国锋、陶煜波

实验地点: 曹光彪西 304 实验日期: 2021 年 3 月 30 日

目录

1. 项目介绍	1
2. 任务概述	2
3. 总体设计	3
4. 程序详细描述	4
5. 数据结构设计	5
6. 开发结果	6
7. 讨论与总结	7

1. 项目介绍

iOS Clock 是一款时钟软件，陪伴你学习、工作与生活。

- 显示时间和显示日期
- 12 和 24 小时模式
- 提供世界时间显示
- 提供闹钟设置功能，自定义添加闹钟事件
- 提供计时器和秒表，倒计时

2. 任务概述

2.1 成员简介与分工、目标

项目目标为完成一个 UI 界面与 iPhone 中系统软件 Clock 尽可能相似的 App

2.2 运行环境

硬件设备：

MacBook Air(13-inch, 2017)

处理器 1.8GHz 双核 Intel Core i5

内存 8 GB 1600 MHz DDR3

图形卡 Intel HD Graphics 6000 1536 MB

软件：

Xcode Version 13.2.1 (13C100)

SF Symbol Version 3.2 (67)

2.3 程序目标分析与概述

此 App 有四个主界面：World Clock（世界时钟）、Alarm（闹钟）、StopWatch（秒表）

和 Timer（定时器），通过底部导航栏可以切换四个界面：

World Clock：可以查看 World Clock 界面内容，显示不同的 World Clock（包括国家地区以及它们的时区偏移和本地时间），通过界面左上方按钮 Edit 可以定向删除某个不需要的 World Clock。

Alarm: 可以查看 Alarm 界面内容, 闹钟分为作息和其他两栏, 分别表示起床/休息的闹钟和自定义事件的时钟, 通过左上方的 Edit 按钮可以看到每个闹钟的编辑效果, 但无法实际编辑。

Stopwatch: 可以查看 Stopwatch 界面内容, 有开始、停止和计次的不同按钮以及计次时间显示, 点击按钮可以显示出实际秒表使用中的不同状态。

Timer: 可以查看 Timer 界面内容, 有时间选择功能、时间截止控制界面和开始、暂停和停止等按钮, 点击按钮可以切换定时器的不同状态。

2.4 运行条件与限制

必须使用 iOS13 以上版本

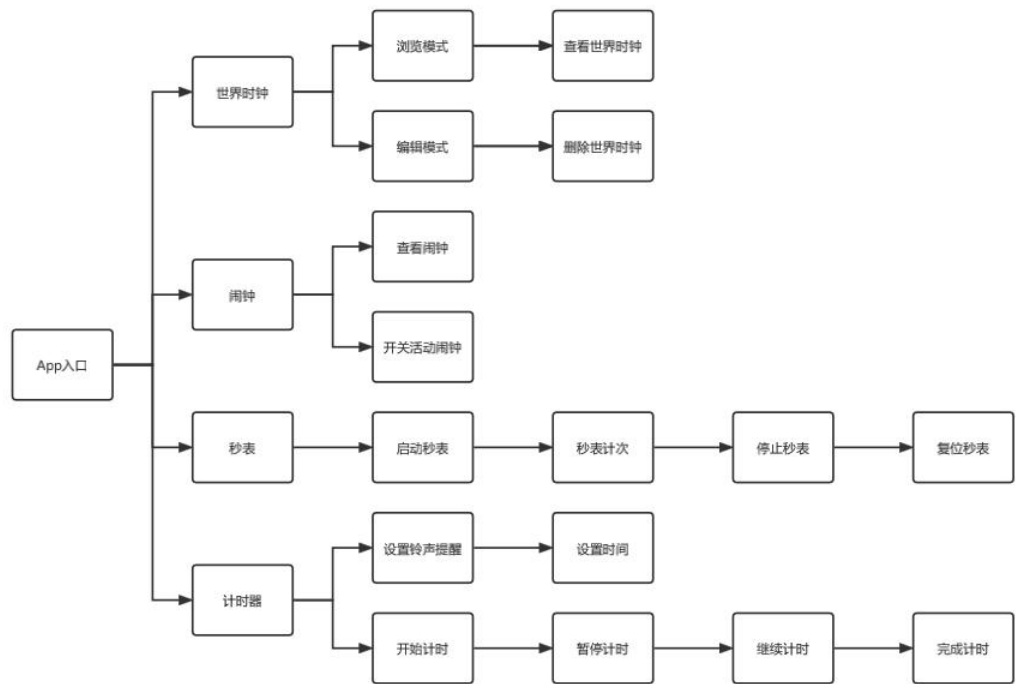
3. 总体设计

3.1 需求分析与设计

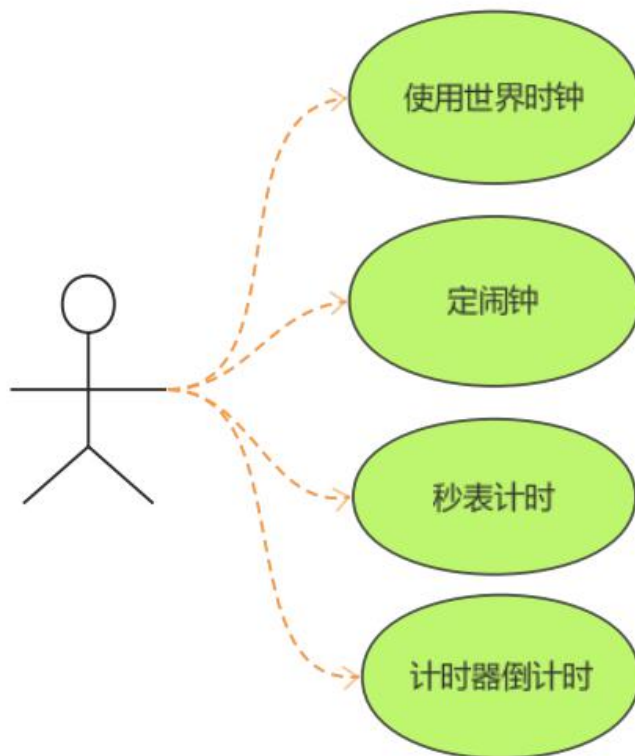
3.1.1 需求分析

目前苹果应用商店中的时钟 App 主要有 Clockology 和 Hi Clock 等, 时钟作为基础应用, 是用户生活中不可或缺的部分, 应该具备显示时间、正向计时和反向计时的功能, 本 App 具备这些功能的相关 UI。

3.1.2 流程图

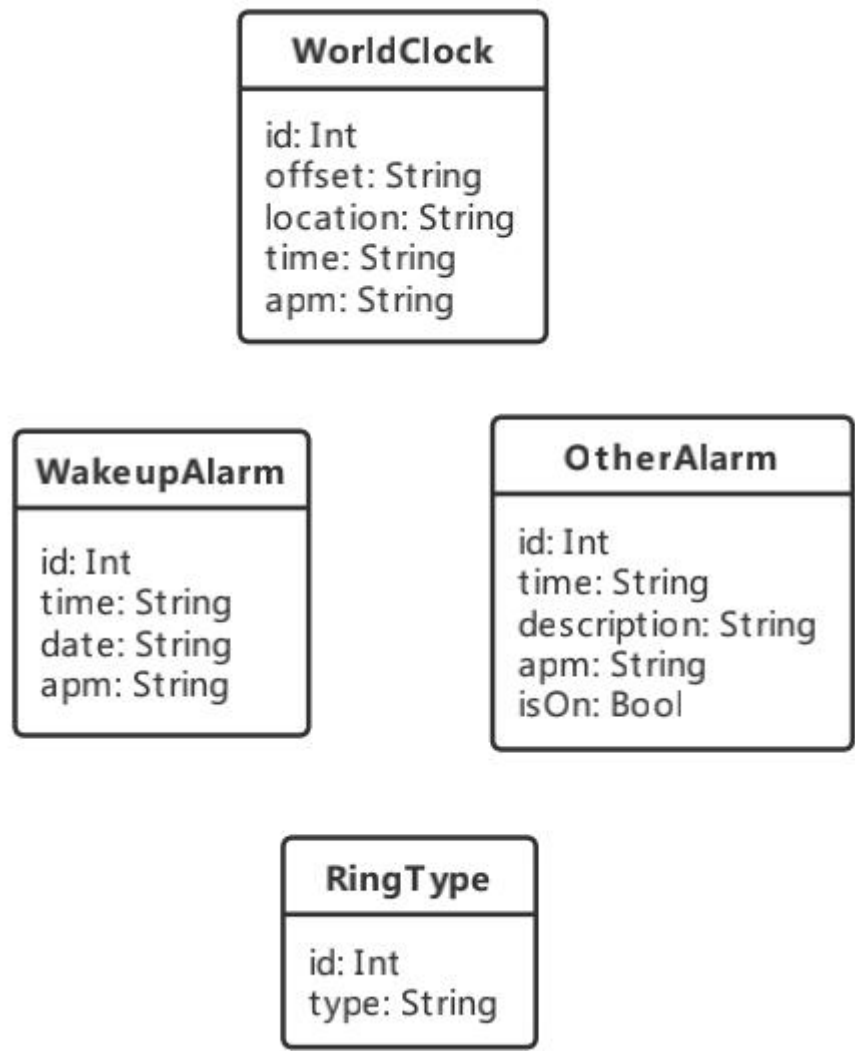


3.1.3 用例图



3.2 详细设计

3.2.1 类图



3.2.2 用例表

用例	工作流程
使用世界时钟	1.点击世界时钟导航栏 2.查看世界时钟 3.通过“Edit”按钮移除世界时钟
定闹钟	1.点击闹钟导航栏 2.查看设置的作息闹钟和活动闹钟

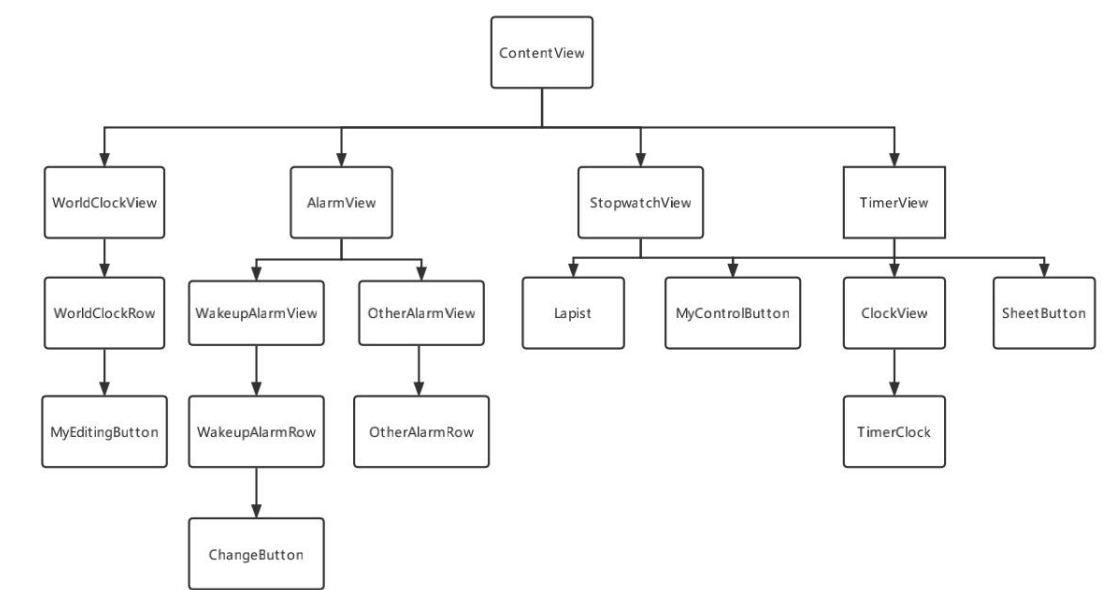
	3.点击右方按钮开关活动闹钟
秒表计时	1.点击秒表导航栏 2.点击“Start”按钮启动秒表 3.点击“Lap”按钮进行秒表计次（再次点击不会显示计次） 4.点击“Stop”按钮停止计时 5.点击“Reset”按钮将秒表复位
计时器倒计时	1.点击“When Timer Ends”按钮选择铃声类型 2.点击上方时间进行计时器时间设置 3.点击“Start”按钮开始倒计时 4.点击“Pause”按钮暂停倒计时 5.点击“Resume”按钮继续计时 6.点击“Cancel”按钮完成计时

4. 程序详细设计

4.1 程序唤醒界面

程序在模拟器上编译通过后，点击 App 图标即可打开应用，默认唤醒界面为 WorldClock 的对应页面，也即 App 打开后的第一个页面。

4.2 主程序 UI



4.3 核心算法

每个主界面遍历数据常量并且按照设定的 View 形式显示在页面上。

4.4 数据库设计

数据以常量形式存储在对应结构的文件中（如 worldclockData 以列表形式存储 WorldClock 类型的数据，则 worldclockData 被声明在 WorldClock 类型定义的文件中）。

4.5 其他设计

无

5. 数据结构设计

5.1 WorldClock

id: Int 类型, 每个世界时钟的唯一标识, 实现 Identifiable 接口

offset: String 类型, WorldClock 中时区偏移的内容

location: String 类型, WorldClock 中的国家地区

time: String 类型, WorldClock 中的 12 小时制当地时间

apm: String 类型, WorldClock 中的上午/下午

5.2 WakeupAlarm

id: Int 类型, 每个作息闹钟的唯一标识, 实现 Identifiable 接口

time: String 类型, WakeupAlarm 中设置的 12 小时制起床/睡觉时间

date: String 类型, WakeupAlarm 闹钟时间对应的日期描述

apm: String 类型, WakeupAlarm 闹钟时间的上午/下午

5.3 OtherAlarm

id: Int 类型, 每个活动闹钟的唯一标识, 实现 Identifiable 接口

time: String 类型, OtherAlarm 中设置的 12 小时制活动时间

description: String 类型, OtherAlarm 闹钟对应的活动描述

apm: String 类型, OtherAlarm 闹钟时间的上午/下午

isOn: Bool 类型, 标识该 OtherAlarm 闹钟是否开启

5.4 RingType

id: Int 类型, 每个铃声类型的唯一标识, 实现 Identifiable 接口

type: String 类型, 铃声音称

6. 开发结果

6.1 软件使用说明

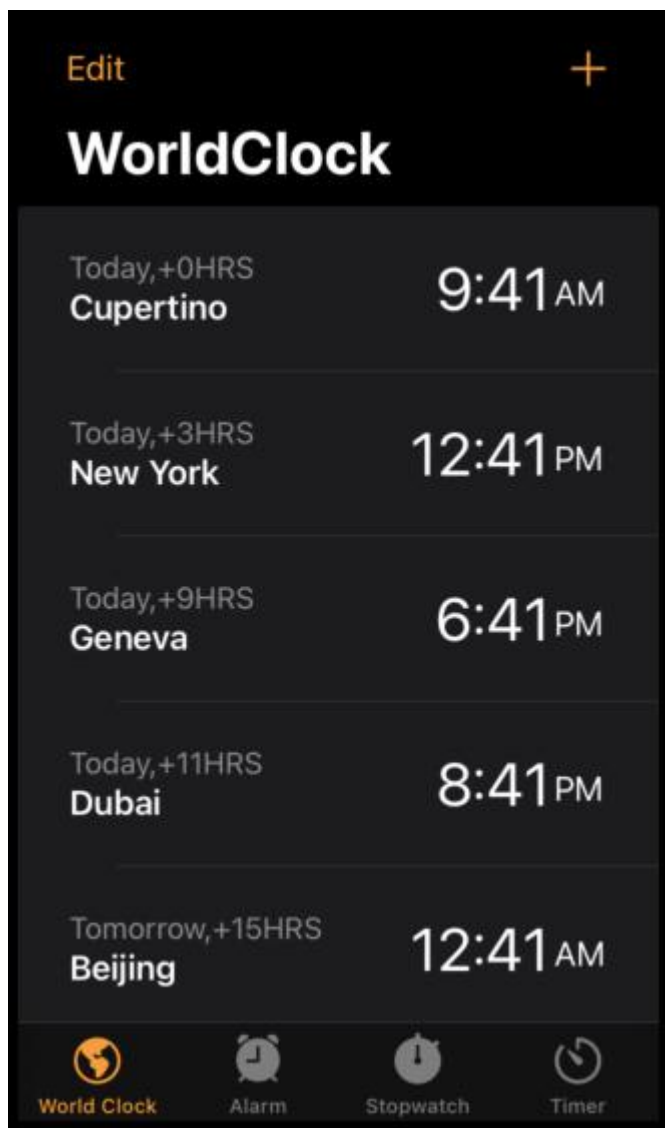
权限要求：所有用户

设备要求：iOS13 及以上

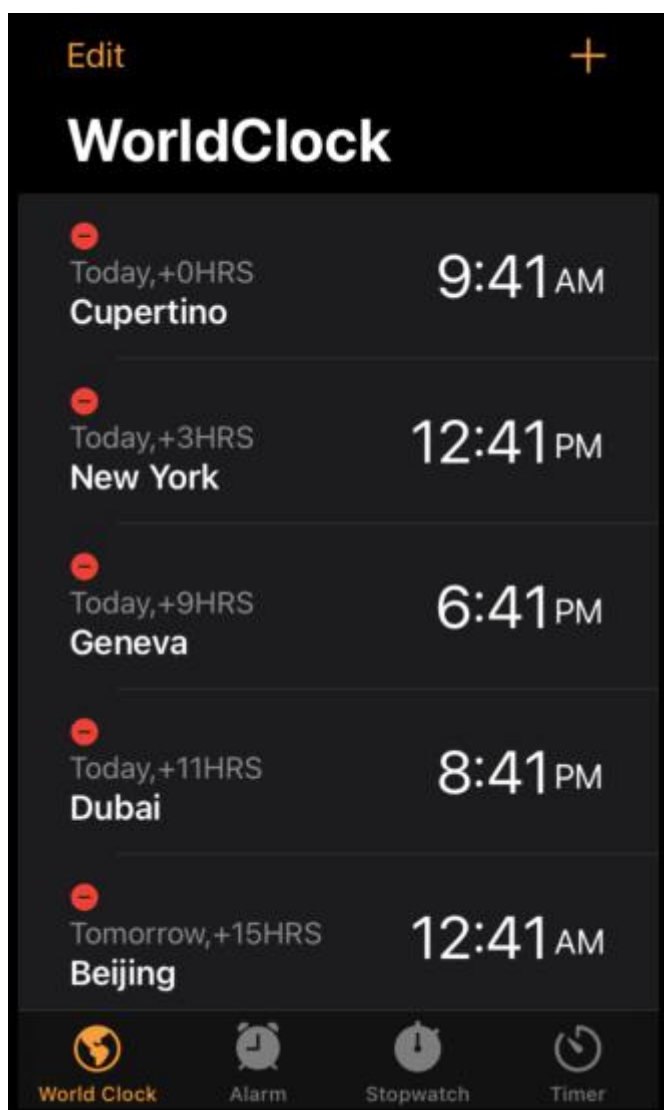
使用场景：用户需要查看世界时间或者正反向计时

6.2 软件预览

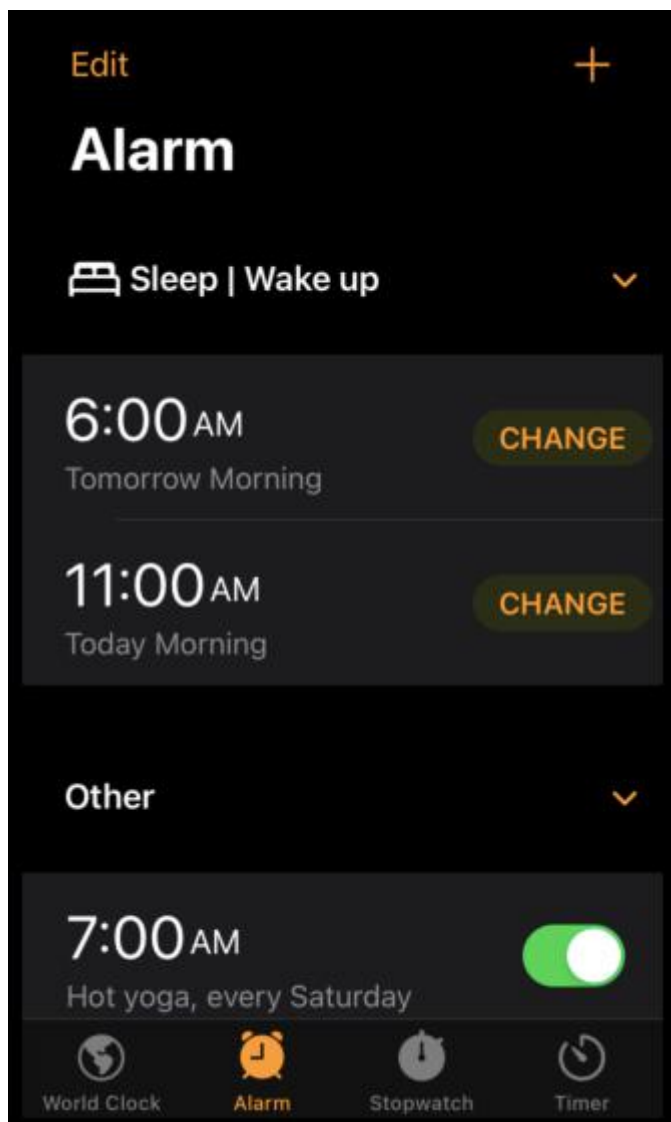
各个界面通过点击底部导航栏图片切换:



WorldClock 界面，展示多个国家地区及其本地时间、偏移时区。



点击“Edit”按钮后进入编辑模式，可以点击每个世界时钟左上角的红色按钮删除此时钟，删除按钮的显示和消失渐渐弹入/弹出，是此项目的重要动画设计之一。



闹钟界面，分区显示作息闹钟（上）和活动闹钟（下），作息闹钟包含闹钟的时间、日期和右部的更改按钮，活动时钟包含闹钟日期时间、对应事件和右部的开关按钮。



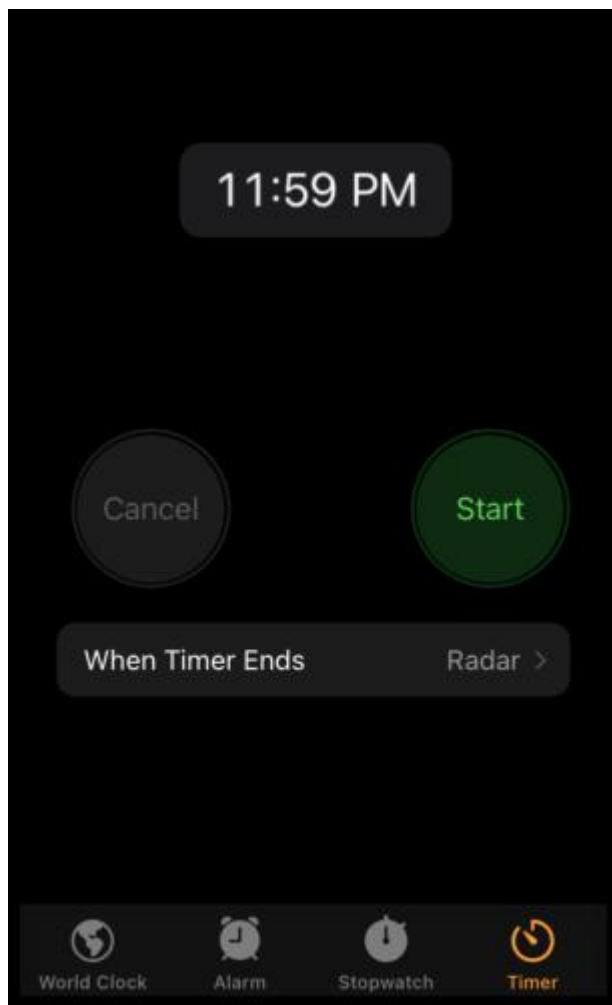
秒表未开始计时的界面，时间显示为 00:00:00，可以点击“Start”按钮开始秒表计时，左侧“Lap”按钮为不可选中状态。



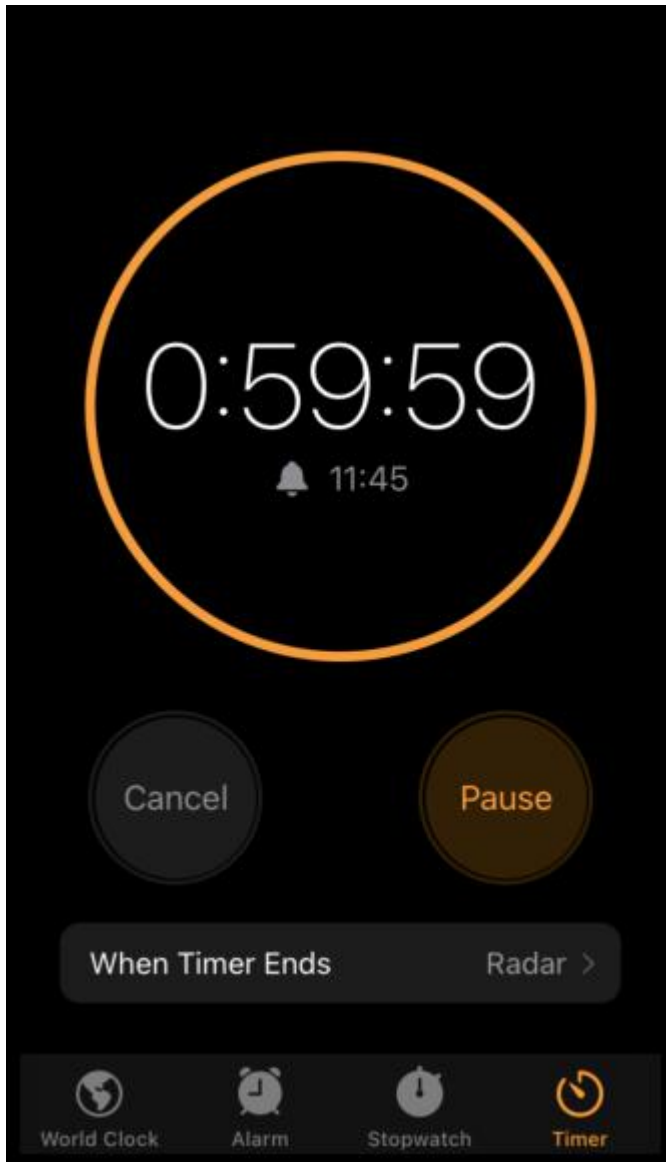
秒表计时中的页面。页面上方时间有变化，右方按钮为“Stop”，点击可以结束秒表计时，左侧按钮“Lap”变为启用状态，点击后在下方以淡入/出的动画显示/隐藏秒表计次信息。



秒表处于停止状态时的页面。此时可以点击“Start”按钮继续计时，则页面回退为上
一个，表示正在计时；也可以点击“Reset”按钮将秒表复位，则页面回退为秒表界面下的
第一个页面，表示尚未开始计时。



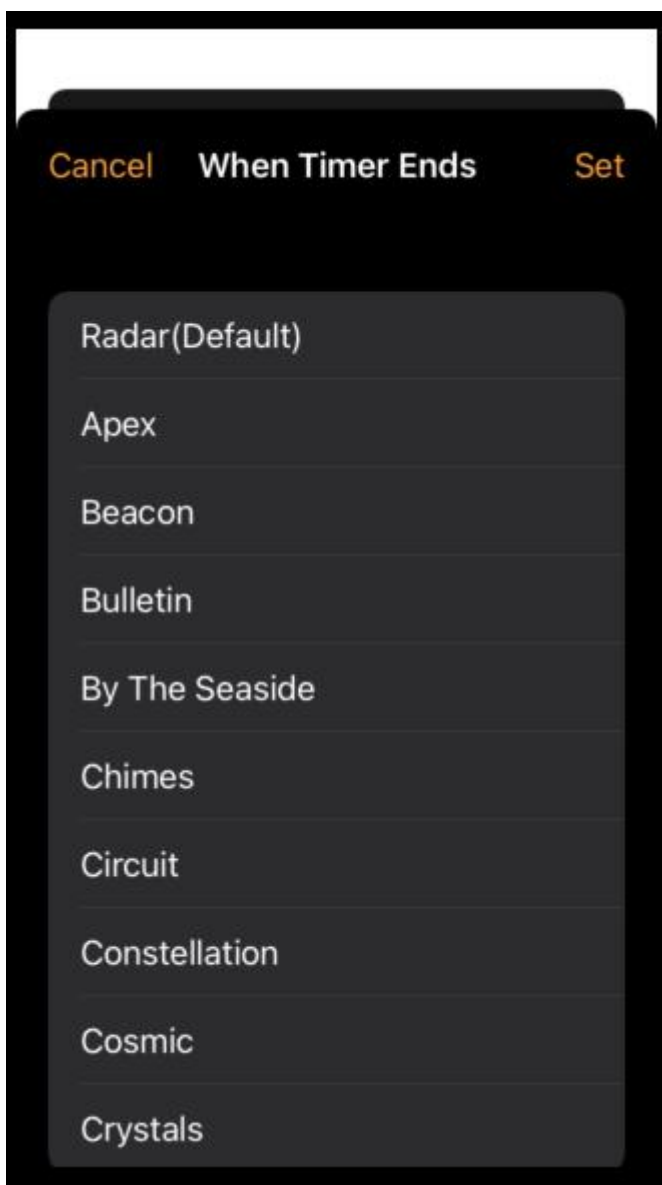
定时器未设置时的界面，上方是时间选择器，可以点击并进行时间设置。下方按钮“When Timer Ends”可以点击显示铃声类型选择。“Cancel”按钮为禁用状态（还没有开始定时器倒计时自然不能结束计时），可以点击“Start”按钮开始定时器计时。



定时器计时中的页面，原来的时间选择器的位置显示为剩余时间，铃铛右侧的时间表示定时器计时结束时的具体时间。此时“Cancel”按钮为启用状态，可以点击结束计时，页面回退为未设置定时器状态；也可以点击“Pause”按钮暂停计时。



暂停计时时的页面，同样可以点击“Cancel”结束计时；也可以点击“Resume”继续倒计时，页面将回退为上一个，表示定时器正在计时。



点击“When Timer Ends”按钮后显示的页面。可以点击“Cancel”或者“Set”按钮进行铃声类型设置（实际不会修改），此时会返回父视图，表示铃声类型设置完成。

7. 讨论与总结

通过前一段时间的学习，我学习到了 SwiftUI 中的视图元素（如 Text、Image 和 Button）、布局方式（如 List、Section、HStack、VStack、ZStack、NavigationView 和 TabView）和动画（Animation），并已经将上述提到的所有元素应用于此项目中。

编写的过程中遇到的主要困难是感觉 SwiftUI 相关文档太少且使用不易，一些不熟悉的元素如 Tabbar 的参数难找途径去设置，我在此类问题的解决中花费了大量时间，希望后面的学习中能找到一些更详细的说明。