Name of project: Task Tiger

Team members: Aneesa Bhakta, Nicole Estabrook, Kelly Hyun, (Salomone Martinez)

Dependencies: Firebase

Special Instructions: Application optimized for iphone 13 plus

General Overview: Our application, Task Tiger, is a productivity app to help people who struggle with deciding what to do first in their busy schedules. The user begins at a login/signup page powered by Firebase. From the sign in they can also view the info page to read more about our app. After logging in, the user is taken to the main page, from which they can navigate the rest of the app. The user can set default times and frequencies for their tasks and edit the theme (color and font) of the entire application from the settings page. From the add task page, the user can add tasks to their to do list. The two types of tasks (one-time and recurring) are sorted into two different tables and can be viewed all at once from the task list page. They can also delete tasks by swiping and edit them by clicking on the task. To get assigned a task, the user can go to the task generator page and enter the amount of time they have available. They can swipe up to reshuffle the tasks and possibly receive a new one. Once satisfied with the task they have been assigned, the user can press start and begin the task timer. The user can then wait for the timer to run out while completing their task, and receive a vibration alert. If they finish early, shake to finish the timer. Finally, the task is deleted if it's a one-time task or the date last done is updated if it's a recurring task. If the user exits the application without completing all their overdue tasks, they will receive a notification upon exiting the application notifying them of the number of overdue tasks they have left.

Required feature checklist

☑ Login/register path with Firebase.
☑ "Settings" screen. The three behaviors we implemented are:
Color scheme, default duration, default frequency
☑ Non-default fonts and colors used

Two major elements used:

☑ Core Data - Saving tasks across launches
☐ User Profile path using camera and photo library
☑ Multithreading - Timer counts down task duration
☐ SwiftUI

Minor Elements used

☑ Two additional view types such as sliders, segmented controllers, etc. The two we implemented are:

Date picker in AddTask VC, Segmented Controller in Settings VC

One of the following:

☑ Table View - View of all tasks currently saved
☐ Collection View
☐ Tab VC
☐ Page VC

Two of the following:

☑ Alerts - Shown when task info not filled out, and when task added
☐ Popovers
☐ Stack Views
☐ Scroll Views
☑ Haptics - When timer finishes, phone vibrates
☑ User Default - Store defaults chosen in settings

At least one of the following per team member:

☑ Local notifications - Triggered when exiting the app if any tasks are overdue
☐ Core Graphics
☑ Gesture Recognition - Swipe up to suggest new task
☑ Animation - Occurs when task completed
☐ Calendar
☑ Core Motion - Shake to end task early
☐ Core Location / MapKit
☐ Core Audio
☐ Others (such as QR code, Koloda, etc.) with approval from the instructor – list them

Work Distribution Table

Disclaimer: We all helped each other out with troubleshooting/fixing different parts of the code so we all helped with each VC/required feature

VC/Required Feature	**Description**	**Who/Percentage**
UI Design	Layout of app, page flow, colors and fonts, use of user defaults, logo	Nicole 30% Aneesa 40% Kelly 30%
Login/Register VC	Firebase Auth to sign in/register	Nicole 100%
Main VC	App home screen, contains toolbar and buttons to navigate app	Aneesa 100%
Settings VC	Allow user to set options for color theme, default frequency and duration of tasks, and whether to allow haptics, all stored in User Defaults.	(Assigned to Salomone) Nicole 90%, Aneesa 10%
Task List Table VC	Allow users to view all tasks saved in Core Data	Nicole 50% Kelly 50%
Edit Task VC	Allow users to edit previously-entered tasks and update Core Data	Nicole 45% Kelly 45% Aneesa 10%
Add Task VC	Allow user to add new task, save to Core Data	Kelly 100%
(Random) Task VC	Generate random task based on time user enters, gesture recognition (swipe up), use of core data (to store tasks)	Aneesa 100%
Timer VC	Countdown time of task using multithreading, delete task (one-time) or updates lastDone	Kelly 75% Nicole 5% Aneesa 20%

	(recurring) once timer finished or phone shaken (core motion) to indicate task done	
Info VC	Info about app	Aneesa 100%
Login/register path with Firebase	Login/Register VCs use Firebase Auth to sign in/register	Nicole 100%
Non-default fonts/colors	All VCs use 4 different background and text color pairs with 4 different non-default fonts, set through setTheme() function	Aneesa 100%
Core Data	Add Task VC uses core data to store tasks as two entities (recurring tasks and one-time tasks), Task List VC and Random Task VC use core data to retrieve and display/use tasks, Edit Task vC updates tasks in Core Data	Aneesa 33% Kelly 33% Nicole 33%
Multithreading	Timer VC uses multithreading to run and update the timer	Kelly 100%
Segmented Controller	Settings VC uses segmented controller to choose the theme	Nicole 80% Aneesa 20%
Date Picker	Add Task VC uses date picker to set the date of one-time tasks	Kelly 100%
Table View	Task List VC uses a table view to display all the tasks, divided by type	Nicole 50% Kelly 50%
Alerts	Add Task VC uses alerts	Kelly 70%

	when fields aren’t filled out and when task successfully added	Nicole 30%
User Defaults	Settings VC stores user settings as user defaults and other VCs retrieve to use them	Nicole 100%
Local Notifications	SceneDelegate’s SceneDidEnterBackground() function checks for any overdue tasks when app is exited and uses local notifications to alert the user if any overdue tasks remain	Aneesa 100%
Gesture Recognition	Random Task VC uses gesture recognition to allow user to request a new task without choosing a new duration	Aneesa 100%
Animation	Timer VC uses animation when tasks completed	Kelly 100%
Core Motion	Timer VC uses core motion to detect shaking to terminate timer early	Nicole 100%
Haptics	Timer VC uses haptics to vibrate when the timer runs out	Aneesa 70% Kelly 30%