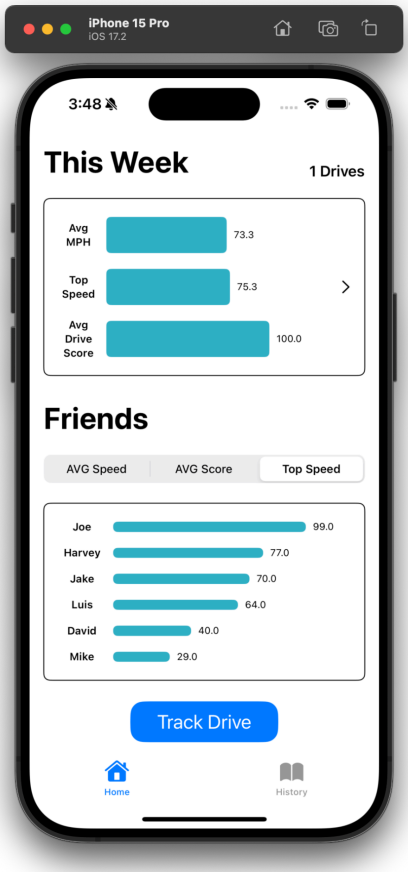
Andrew Kostryba

Project Name: DriveBuddy

Description: DriveBuddy is an iOS app built with SwiftUI that leverages Apple's Core Location and Core Motion frameworks to track and score user driving. Users can analyze their drives and compare driving metrics, such as average speed, average drive score, and top speeds with friends on the app. A single drive's "Drive Score" is calculated based on the rate of rapid acceleration and sudden braking occurrences that reflect dangerous driving maneuvers. When driving, users will see a live drive score, and once the drive is complete, they will be presented with a detailed summary of the drive data. All drives and data recorded are stored and saved by using Apple's Core Data framework.

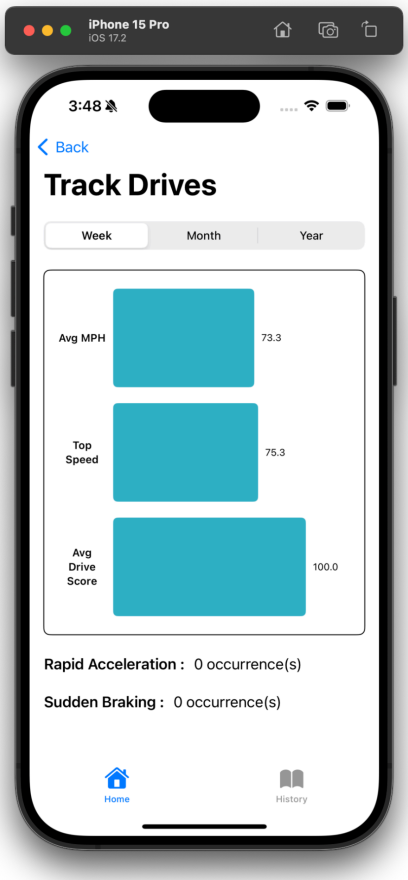
Screens:



Screen 1: Home Screen

This view is wrapped inside a TabView and is selected upon launch by default. The top half of the view displays driving data from drives completed in the current week. The top bar chart is wrapped in a navigation link to the Drive Stats view. In the bottom half, using the picker, the user can choose to view the average speeds, average scores, or top speeds recorded by friends. The "Track Drive" button is a navigation link to the Drive View where the user can begin tracking a drive.

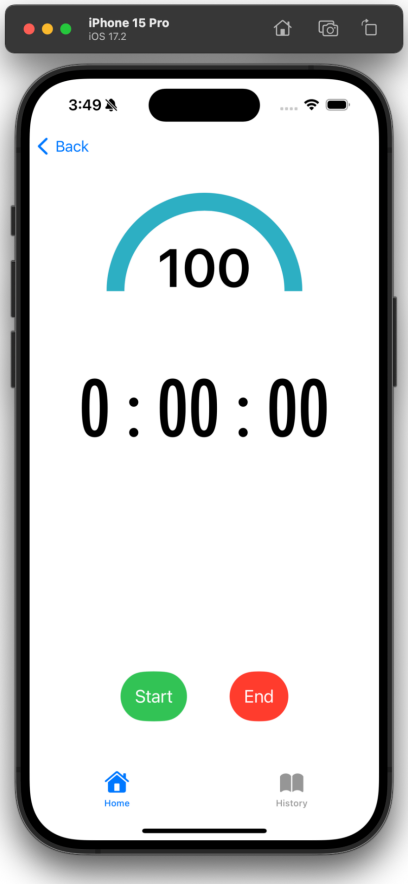
The horizontal bar charts were achieved with my custom **HorizontalBarChartView**. Using the GeometryReader to obtain parent view dimensions, I proportioned sizes of the Rectangles and Text labels in the view to fit accordingly.



Screen 2: Stats Screen

This screen allows users to view their average speed, top speed, and average drive score over the past week, month, or year. The data is displayed in the HorizontalBarChartView, allowing the user to visualize their driving statistics over the selected time periods.

At the bottom of the view are two text labels representing the total number of rapid acceleration and sudden braking occurrences over the selected time period.



Screen 3: Drive Screen

After tapping the “Track Drive” button on the Home screen, the app will navigate to this view where users begin tracking their drives and view a live score.

At the top, the arc is drawn with my own **Arc** struct that visually represents the drive score by filling a portion of the shape. Using the path function, Arc determines the start angle for the arc based on the drive score and adds it to the path within a rectangle. In the Drive View, two arcs are placed in a ZStack. The one on top represents the drive score, while the bottom one represents a perfect drive score with a lighter opacity.

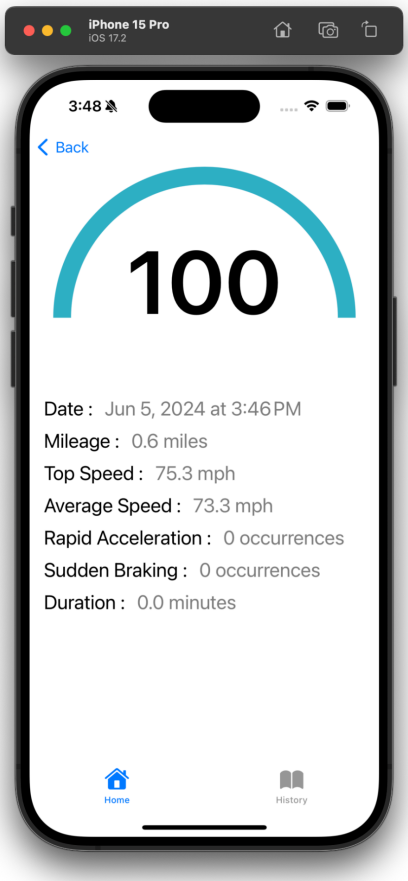
Andrew Kostryba

Below the score is a timer that is controlled by my **Duration** class. Using the Singleton design pattern, Duration allows me to start or stop a single timer, and retrieve the hours, minutes, and seconds to display in the view.

The start button starts the timer to track duration and begins motion and location tracking with my **LocationTracker** and **MotionTracker** classes. MotionTracker handles device motion updates to detect rapid acceleration and sudden braking. LocationTracker is a CLLocationManagerDelegate that handles location updates to track device location and speed.

The end button stores the drive data with Core Data, stops location and motion tracking, and presents the Drive Summary View.

To avoid false rapid acceleration or sudden braking detections, the device must be kept vertical and facing the direction of travel, like if it were in a car phone mount. MotionTracker will detect when the device isn't vertical, which will trigger an alert informing the user to keep the phone vertical.

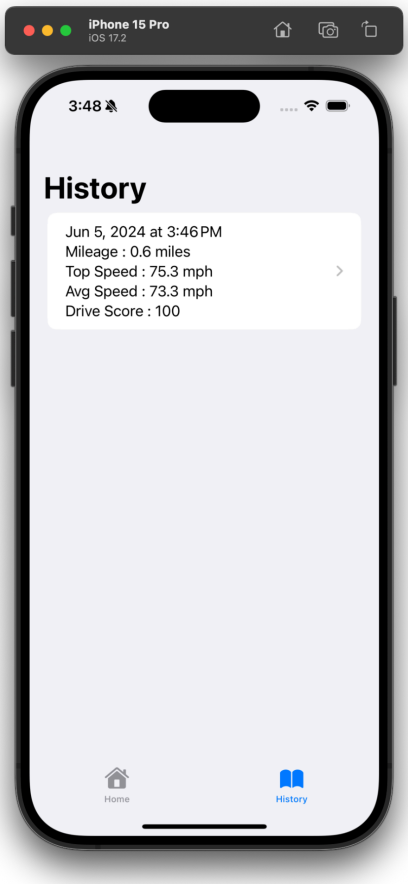


Screen 4: Drive Summary Screen

This screen presents the user with a summary of all the data collected from their drive.

The score is visualized with the Arc struct, like in the Drive Screen.

Below the score is a list of information about the drive.



Screen 5: Drive History Screen

This view is wrapped inside a tab view and is accessible by tapping the history tab at the bottom of the screen.

This view shows a list of completed drives fetched from what's been saved with CoreData. For each recorded drive, the list displays the date, distance, top speed, average speed, and drive score.

Tapping on a drive will navigate the user to the Drive Summary screen, which will show further details about the drive.