Mechatronics Final Project

Safe Velocity Indicator

Anthony Brill
Jong Bae
Matthew Moorhead

Undershoot

Too Slow

Overshoot

Too Fast

Objective

Safe Velocity Indicator

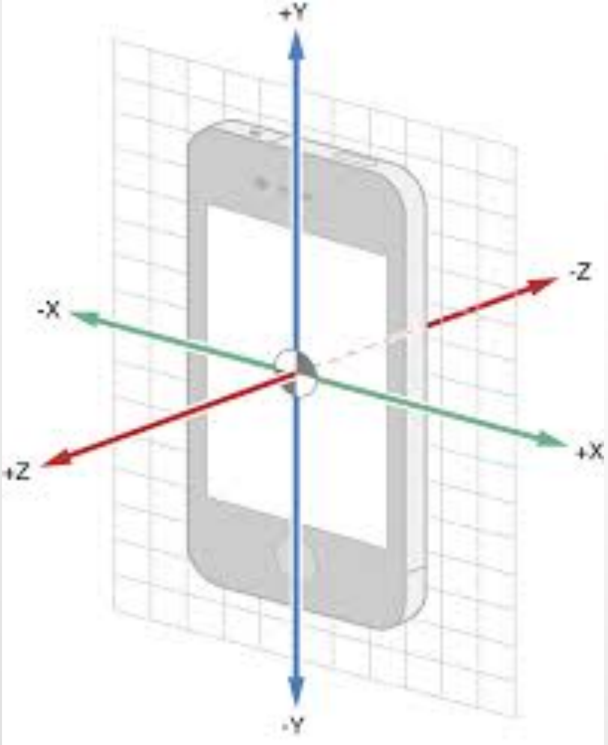
- To design a safe velocity indicator using iPhone programming to ensure rider's safety

- To provide feedback to the rider so that the rider is aware of one's speed as he/she approaches the jump

- To add flexibility to the application such that the rider can modify and personalize the range of "safe-speed"

Difficulties

Safe Velocity Indicator



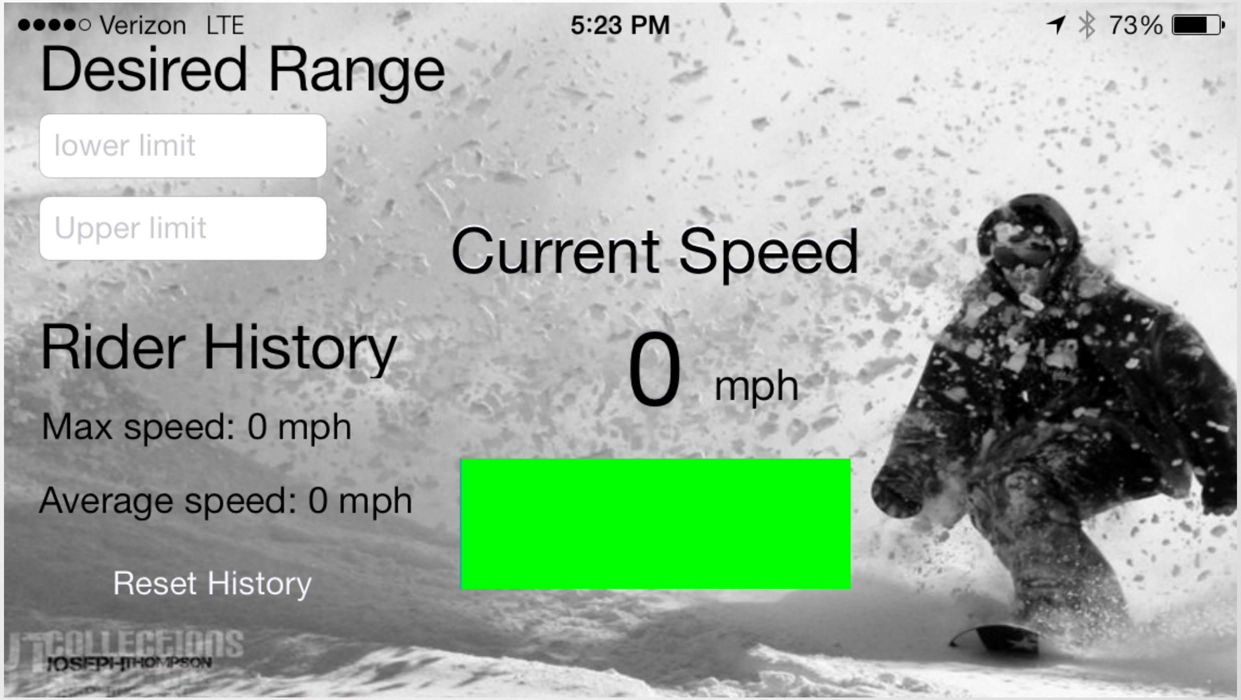
- **Our original intent was to use accelerometer in the iPhone to retrieve velocity data**

- **Accelerometer is not suitable to calculate velocity, because small error in acceleration will cause for the velocity to drift over time**

- **Alternative Solution : GPS chip in iPhone**

User–Interface

Safe Velocity Indicator



●●●●○ Verizon LTE 5:23 PM 73%

Desired Range

lower limit

Upper limit

Current Speed

0 mph

🟩

Rider History

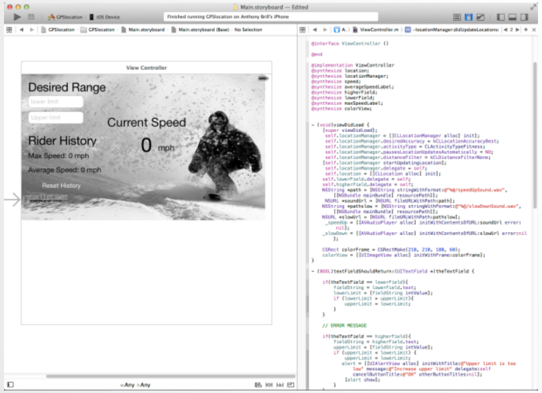
Max speed: 0 mph

Average speed: 0 mph

Reset History

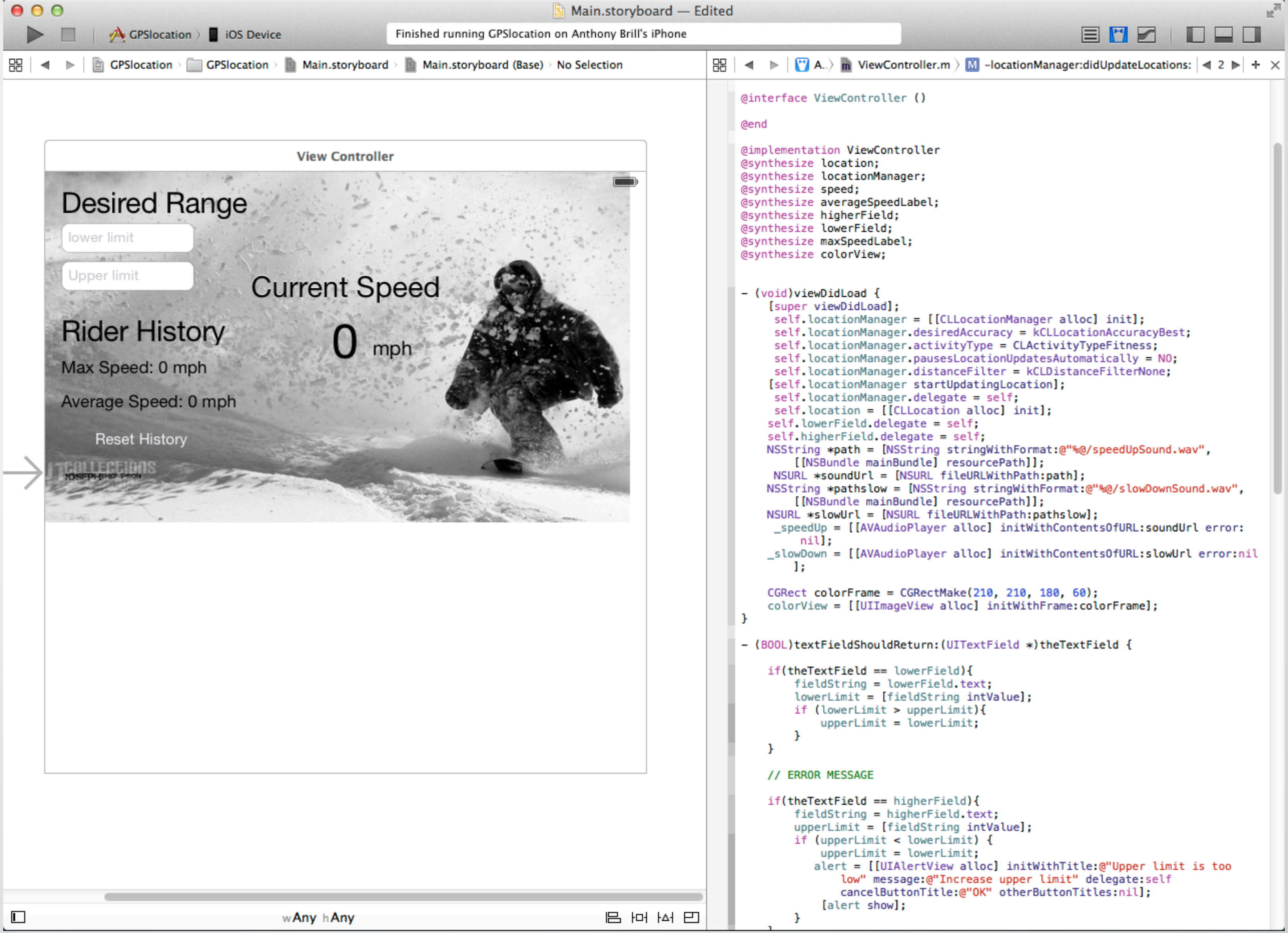
Implementation/Coding

Safe Velocity Indicator



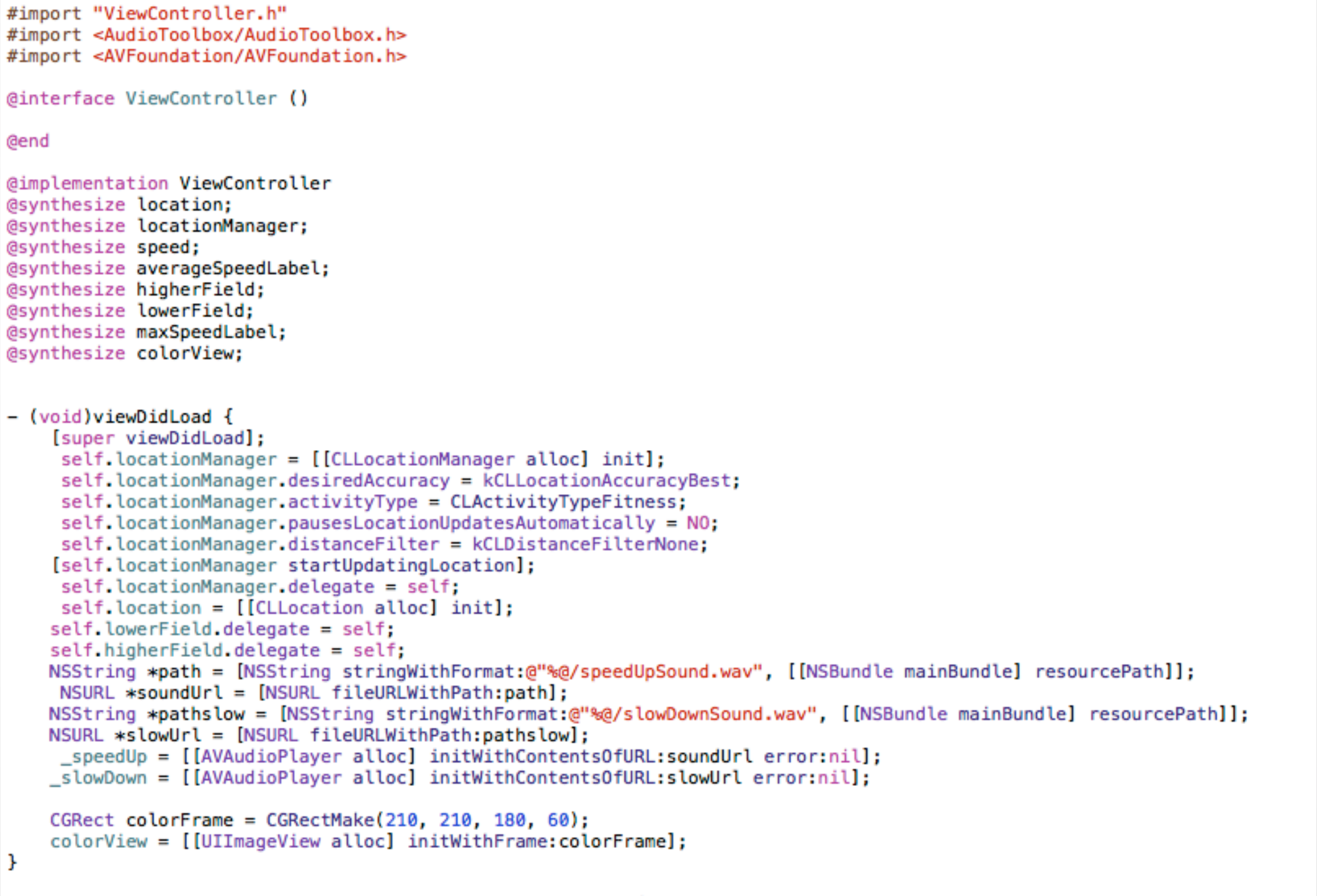
```
@interface ViewController ()

@end

@implementation ViewController
@synthesize location;
@synthesize locationManager;
@synthesize speed;
@synthesize averageSpeedLabel;
@synthesize higherField;
@synthesize lowerField;
@synthesize maxSpeedLabel;
@synthesize colorView;


- (void)viewDidLoad {
    [super viewDidLoad];
     self.locationManager = [[CLLocationManager alloc] init];
     self.locationManager.desiredAccuracy = kCLLocationAccuracyBest;
     self.locationManager.activityType = CLActivityTypeFitness;
     self.locationManager.pausesLocationUpdatesAutomatically = NO;
     self.locationManager.distanceFilter = kCLDistanceFilterNone;
    [self.locationManager startUpdatingLocation];
     self.locationManager.delegate = self;
     self.location = [[CLLocation alloc] init];
    self.lowerField.delegate = self;
    self.higherField.delegate = self;
    NSString *path = [NSString stringWithFormat:@"%@/speedUpSound.wav",
        [[NSBundle mainBundle] resourcePath]];
     NSURL *soundUrl = [NSURL fileURLWithPath:path];
    NSString *pathslow = [NSString stringWithFormat:@"%@/slowDownSound.wav",
        [[NSBundle mainBundle] resourcePath]];
    NSURL *slowUrl = [NSURL fileURLWithPath:pathslow];
     _speedUp = [[AVAudioPlayer alloc] initWithContentsOfURL:soundUrl error:
        nil];
    _slowDown = [[AVAudioPlayer alloc] initWithContentsOfURL:slowUrl error:nil
        ];

    CGRect colorFrame = CGRectMake(210, 210, 180, 60);
    colorView = [[UIImageView alloc] initWithFrame:colorFrame];
}

- (BOOL)textFieldShouldReturn:(UITextField *)theTextField {

    if(theTextField == lowerField){
        fieldString = lowerField.text;
        lowerLimit = [fieldString intValue];
        if (lowerLimit > upperLimit){
            upperLimit = lowerLimit;
        }
    }

    // ERROR MESSAGE

    if(theTextField == higherField){
        fieldString = higherField.text;
        upperLimit = [fieldString intValue];
        if (upperLimit < lowerLimit) {
            upperLimit = lowerLimit;
           alert = [[UIAlertView alloc] initWithTitle:@"Upper limit is too
               low" message:@"Increase upper limit" delegate:self
               cancelButtonTitle:@"OK" otherButtonTitles:nil];
            [alert show];
        }
```

- **Used CoreLocation framework**
- **CoreLocation returns a user's latitude/longitude coordinates and makes use of them to calculate velocity**

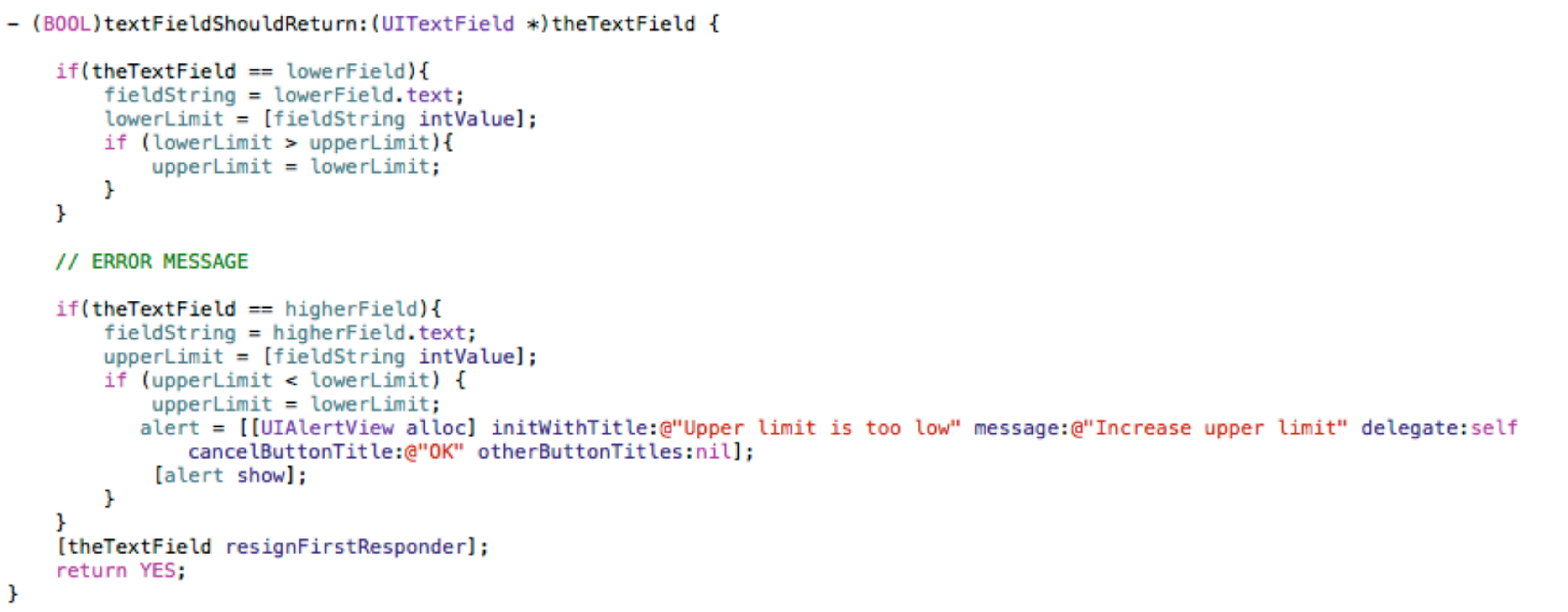


Finished running GPSlocation on Anthony Brill's iPhone

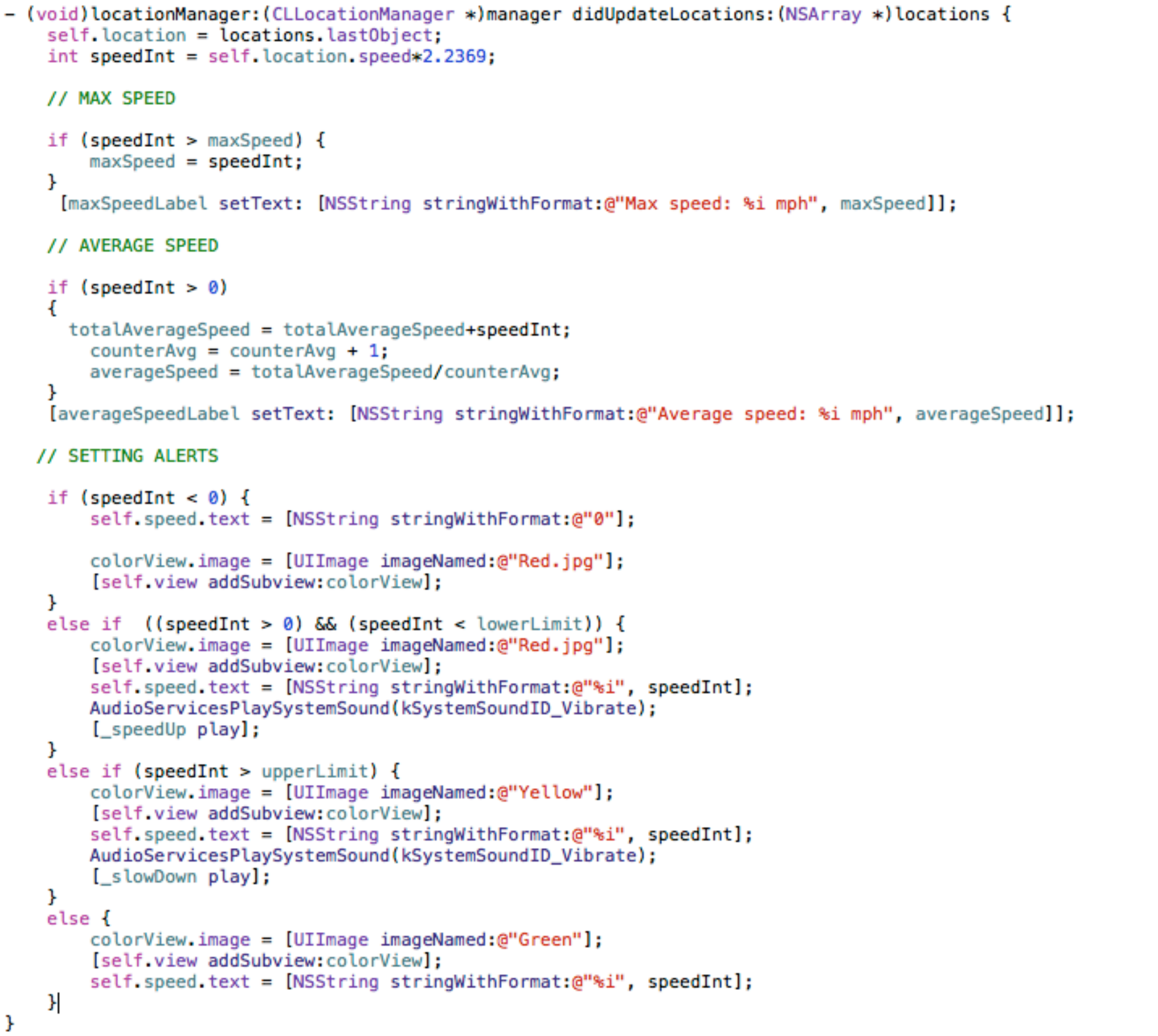
```
@interface ViewController ()

@end

@implementation ViewController
@synthesize location;
@synthesize locationManager;
@synthesize speed;
@synthesize averageSpeedLabel;
@synthesize higherField;
@synthesize lowerField;
@synthesize maxSpeedLabel;
@synthesize colorView;


- (void)viewDidLoad {
    [super viewDidLoad];
     self.locationManager = [[CLLocationManager alloc] init];
     self.locationManager.desiredAccuracy = kCLLocationAccuracyBest;
     self.locationManager.activityType = CLActivityTypeFitness;
     self.locationManager.pausesLocationUpdatesAutomatically = NO;
     self.locationManager.distanceFilter = kCLDistanceFilterNone;
    [self.locationManager startUpdatingLocation];
     self.locationManager.delegate = self;
     self.location = [[CLLocation alloc] init];
    self.lowerField.delegate = self;
    self.higherField.delegate = self;
    NSString *path = [NSString stringWithFormat:@"%@/speedUpSound.wav",
        [[NSBundle mainBundle] resourcePath]];
     NSURL *soundUrl = [NSURL fileURLWithPath:path];
    NSString *pathslow = [NSString stringWithFormat:@"%@/slowDownSound.wav",
        [[NSBundle mainBundle] resourcePath]];
    NSURL *slowUrl = [NSURL fileURLWithPath:pathslow];
     _speedUp = [[AVAudioPlayer alloc] initWithContentsOfURL:soundUrl error:
         nil];
    _slowDown = [[AVAudioPlayer alloc] initWithContentsOfURL:slowUrl error:nil
        ];

    CGRect colorFrame = CGRectMake(210, 210, 180, 60);
    colorView = [[UIImageView alloc] initWithFrame:colorFrame];
}

- (BOOL)textFieldShouldReturn:(UITextField *)theTextField {

    if(theTextField == lowerField){
        fieldString = lowerField.text;
        lowerLimit = [fieldString intValue];
        if (lowerLimit > upperLimit){
            upperLimit = lowerLimit;
        }
    }

    // ERROR MESSAGE

    if(theTextField == higherField){
        fieldString = higherField.text;
        upperLimit = [fieldString intValue];
        if (upperLimit < lowerLimit) {
            upperLimit = lowerLimit;
           alert = [[UIAlertView alloc] initWithTitle:@"Upper limit is too
               low" message:@"Increase upper limit" delegate:self
               cancelButtonTitle:@"OK" otherButtonTitles:nil];
            [alert show];
        }
```



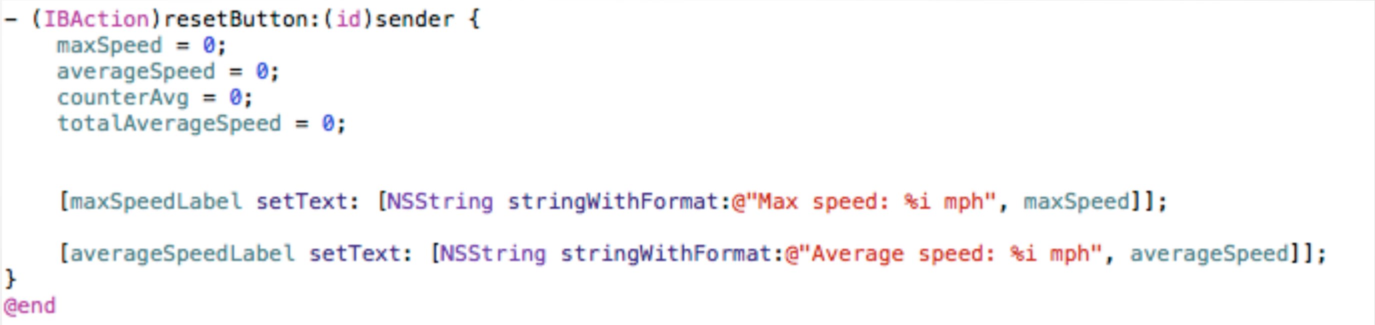
```
#import "ViewController.h"
#import <AudioToolbox/AudioToolbox.h>
#import <AVFoundation/AVFoundation.h>

@interface ViewController ()

@end

@implementation ViewController
@synthesize location;
@synthesize locationManager;
@synthesize speed;
@synthesize averageSpeedLabel;
@synthesize higherField;
@synthesize lowerField;
@synthesize maxSpeedLabel;
@synthesize colorView;


- (void)viewDidLoad {
    [super viewDidLoad];
     self.locationManager = [[CLLocationManager alloc] init];
     self.locationManager.desiredAccuracy = kCLLocationAccuracyBest;
     self.locationManager.activityType = CLActivityTypeFitness;
     self.locationManager.pausesLocationUpdatesAutomatically = NO;
     self.locationManager.distanceFilter = kCLDistanceFilterNone;
    [self.locationManager startUpdatingLocation];
     self.locationManager.delegate = self;
     self.location = [[CLLocation alloc] init];
    self.lowerField.delegate = self;
    self.higherField.delegate = self;
    NSString *path = [NSString stringWithFormat:@"%@/speedUpSound.wav", [[NSBundle mainBundle] resourcePath]];
     NSURL *soundUrl = [NSURL fileURLWithPath:path];
    NSString *pathslow = [NSString stringWithFormat:@"%@/slowDownSound.wav", [[NSBundle mainBundle] resourcePath]];
    NSURL *slowUrl = [NSURL fileURLWithPath:pathslow];
     _speedUp = [[AVAudioPlayer alloc] initWithContentsOfURL:soundUrl error:nil];
    _slowDown = [[AVAudioPlayer alloc] initWithContentsOfURL:slowUrl error:nil];

    CGRect colorFrame = CGRectMake(210, 210, 180, 60);
    colorView = [[UIImageView alloc] initWithFrame:colorFrame];
}
```



```
- (BOOL)textFieldShouldReturn:(UITextField *)theTextField {

    if(theTextField == lowerField){
        fieldString = lowerField.text;
        lowerLimit = [fieldString intValue];
        if (lowerLimit > upperLimit){
            upperLimit = lowerLimit;
        }
    }

    // ERROR MESSAGE

    if(theTextField == higherField){
        fieldString = higherField.text;
        upperLimit = [fieldString intValue];
        if (upperLimit < lowerLimit) {
            upperLimit = lowerLimit;
           alert = [[UIAlertView alloc] initWithTitle:@"Upper limit is too low" message:@"Increase upper limit" delegate:self
               cancelButtonTitle:@"OK" otherButtonTitles:nil];
            [alert show];
        }
    }
    [theTextField resignFirstResponder];
    return YES;
}
```



```
- (void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations {
    self.location = locations.lastObject;
    int speedInt = self.location.speed*2.2369;

    // MAX SPEED

    if (speedInt > maxSpeed) {
        maxSpeed = speedInt;
    }
     [maxSpeedLabel setText: [NSString stringWithFormat:@"Max speed: %i mph", maxSpeed]];

    // AVERAGE SPEED

    if (speedInt > 0)
    {
      totalAverageSpeed = totalAverageSpeed+speedInt;
        counterAvg = counterAvg + 1;
        averageSpeed = totalAverageSpeed/counterAvg;
    }
    [averageSpeedLabel setText: [NSString stringWithFormat:@"Average speed: %i mph", averageSpeed]];

   // SETTING ALERTS

    if (speedInt < 0) {
        self.speed.text = [NSString stringWithFormat:@"0"];

        colorView.image = [UIImage imageNamed:@"Red.jpg"];
        [self.view addSubview:colorView];
    }
    else if  ((speedInt > 0) && (speedInt < lowerLimit)) {
        colorView.image = [UIImage imageNamed:@"Red.jpg"];
        [self.view addSubview:colorView];
        self.speed.text = [NSString stringWithFormat:@"%i", speedInt];
        AudioServicesPlaySystemSound(kSystemSoundID_Vibrate);
        [_speedUp play];
    }
    else if (speedInt > upperLimit) {
        colorView.image = [UIImage imageNamed:@"Yellow"];
        [self.view addSubview:colorView];
        self.speed.text = [NSString stringWithFormat:@"%i", speedInt];
        AudioServicesPlaySystemSound(kSystemSoundID_Vibrate);
        [_slowDown play];
    }
    else {
        colorView.image = [UIImage imageNamed:@"Green"];
        [self.view addSubview:colorView];
        self.speed.text = [NSString stringWithFormat:@"%i", speedInt];
    }
}
```



```objc
- (IBAction)resetButton:(id)sender {
    maxSpeed = 0;
    averageSpeed = 0;
    counterAvg = 0;
    totalAverageSpeed = 0;


    [maxSpeedLabel setText: [NSString stringWithFormat:@"Max speed: %i mph", maxSpeed]];

    [averageSpeedLabel setText: [NSString stringWithFormat:@"Average speed: %i mph", averageSpeed]];
}
@end
```

Issues with CoreLocation

Safe Velocity Indicator

- Velocity measurements are less reliable at low speed
- GPS restricts the user to outdoor activities
- CoreLocation framework does not allow the sampling frequency to be directly set
- Transmitting GPS data requires heavy battery usage

Feedback Mechanism

Safe Velocity Indicator

	Light	• The light will be RED if the rider's speed is too slow and YELLOW if too fast • If the speed is in the predefined range, the light will be GREEN
	Vibration	• Phone will be vibrating if the rider's speed is outside of defined range (User-defined) • Vibration will stop as soon as rider's speed resides in safe-range
	Sound	• If the rider's speed is greater than the safe speed range, the app will transmit the sound "Slow-down" • If the speed is lower than the range it will transmit "Speed-up" sound

Video: Working Prototype

Safe Velocity Indicator

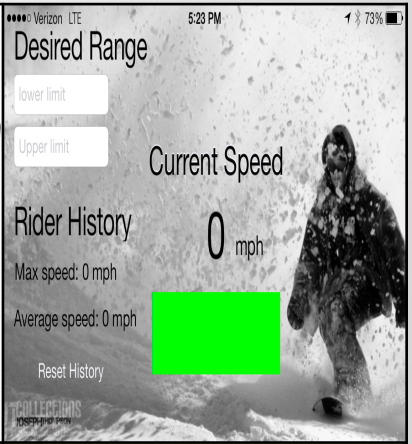
Real–Life Implementation

Safe Velocity Indicator

- iPhone can be easily mounted on a helmet
- Rider can check one's speed in real-time
- Without spending hundreds of dollars on HUD goggles which doesn't provide any safety alert on speed control, the rider can simply use the iPhone to ensure their safety

HUD Product Comparison

Safe Velocity Indicator

			 ●●●●○ Verizon LTE 5:23 PM 73% Desired Range lower limit Upper limit Current Speed Rider History 0 mph Max speed: 0 mph Average speed: 0 mph 🟩 Reset History
Oakley AirWave	Recon SNOW2 HUD	Zeal Z3 GPS Goggle	Our Product
$650	$550	$488	$0

Future Development

Safe Velocity Indicator

- Remember speeds for different jumps
- Download jump data from Mountain Website
- Upload jump data to Mountain Website
- Allow warning sounds to run while playing music
- Ability to turn off specific alerts
- Secret features we're not allowed to talk about

Questions

?

?

?

?