

Lauren DiDonato  
Ariel Viggiano

## App Title: Fight Me

---

### Application Definition Statement

'Fight Me' is a simulator for users to virtually fight family, friends, and strangers. This is meant to help solve petty disagreements and let the users have some fun. We also want to encourage users to be active in their daily life and their daily steps will affect how well they 'fight'.

After creating a profile to fight with, the app will determine a fitness and stamina level based on the height, weight and age they provided. They can improve these stats by completing achievements and working out. These actions will also allow the users to gain skill points to buy weapons and equipment to help them fight.

The app will pull data from their step counter to determine if they're staying active. The more active they are the better their fitness level and stamina, but if they aren't active for several days in a row, their stats will start suffering.

---

### Features Implemented

- User Profile
    - height, weight, name, age, fitness level, stamina
  - Ability to equipment armor and weapons
  - Fighting a simulated opponent
  - After fight summary
  - Step counter and HealthKit integration
  - Inventory
  - Equipment
  - Equipment shop
  - Luck system
  - Skill Points
  - Special Event: Stabby the Crabby
-

## Features for the Next Version

- Finding and challenging opponents - not via bluetooth
  - 2 Fighting options: beat down (with equipment) or fisticuffs (no equipments)
  - Special weapons
  - Rematch
  - Equipment theft after beat down fight
  - Achievement tracking
  - Fitness Level degradation
  - Picking a fight location
- 

## Self Evaluation

We attempted to use two ios functionalities (HealthKit and Bluetooth) that were not covered in class and succeeded with one. We implemented the base functionality we envisioned for the app.

We did extensive object manipulation across multiple UIViews as well as live UI updating when changes are made.

We successfully implemented the challenge system for “fighting”, we were able to show a working product without the use of bluetooth.

We believe we earned an B+ / A-

---

## Documentation

### *UML / Class Definitions*

#### Classes

- Achievement
  - Generic achievement, contains a description, title, and requirement for the user to achieve
- AchievementHandler
  - Handles a list of all the available achievements and the user’s currently achieved achievements
  - Meant to take care of checking if a user has yet completed an achievement
- Equipment

- Generic piece of equipment with a name, skill point cost, buff and type (Weapon or Armor)
  - Used by fighters to fight, degrades over time with each use
- EquipmentHandler
  - Does functionality for the equipment including adding and deleting from inventory, and validation
  - Contains allEquipment (list of all possible equipment) and userEquipment (list of items in users inventory)
- Fighter
  - The fighter profile, extends FighterDetails
  - Contains a FitnessHandler for the fighter
  - Has informations for the user's stamina, fitness level, and currently equipped equipment
- FighterDetails
  - Takes care of generic fight details and determines their base fitness level
- FitnessHandler
  - Takes care of calculating a fighter's overall fitness and taking their steps overtime
  - Meant to increase their fitness level overtime as they walk more and participate in the fights and decrease it as they do no

#### Controllers

- EquipmentDetailController
  - Displays the details of the controller
  - Can be reached by 2 other pages, InventoryController and ShopController
  - If accessed by the InventoryController there will be a button to equip the it
    - The user will be alerted and asked to confirm
    - Once confirmed the application will double check that the user can equip it and will alert the user with an appropriate message
  - If accessed by the ShopController the user will have the option to buy the piece of equipment
    - The user will be prompted to confirm payment and the program will attempt to purchase the item with the user's skill points
    - The user will be alerted with an appropriate message detailing if it was correctly purchased or not and whi
- FightController
  - Meant to handle the bluetooth connection and fight of 2 different fighters, currently simulates a fight with the given information
  - The user can win, lose to the opponent, or lose to stabby the crabby
    - The user with be alerted of the result and the aftermath: stamina lost and any skill points gained
- NewUserController
  - Handles creation of a new Fighter with a username, birthdate, height, and weight. Once created the user is meant to be sent to the tab bar controller to begin using the application

- InventoryController
  - Table view controller for displaying all the items in the fighter's inventory
  - User can view all equipment, just armor, or just weapons
  - User can click on the equipment to view more details
- ProfileController
  - Displays all the fighters info including current fitness level, skill points, currently equipped items, height and weight
  - User can click on height and weight to update them
- ShopController
  - Table view controller for displaying all the available equipment for purchase
  - User can view all equipment, just armor, or just weapons
  - User can click on the equipment to view more details
- SplashPageController
  - Meant to be a loading screen to check if a user currently exists, if it does send the user to the tabBarcontroller otherwise send them to the NewUserController to create a profile. Sending the fighter between pages got a bit confusing and is not properly implemented
- TabBarController
  - Generic class from created a tabbed app, nothing changed

#### Delegates

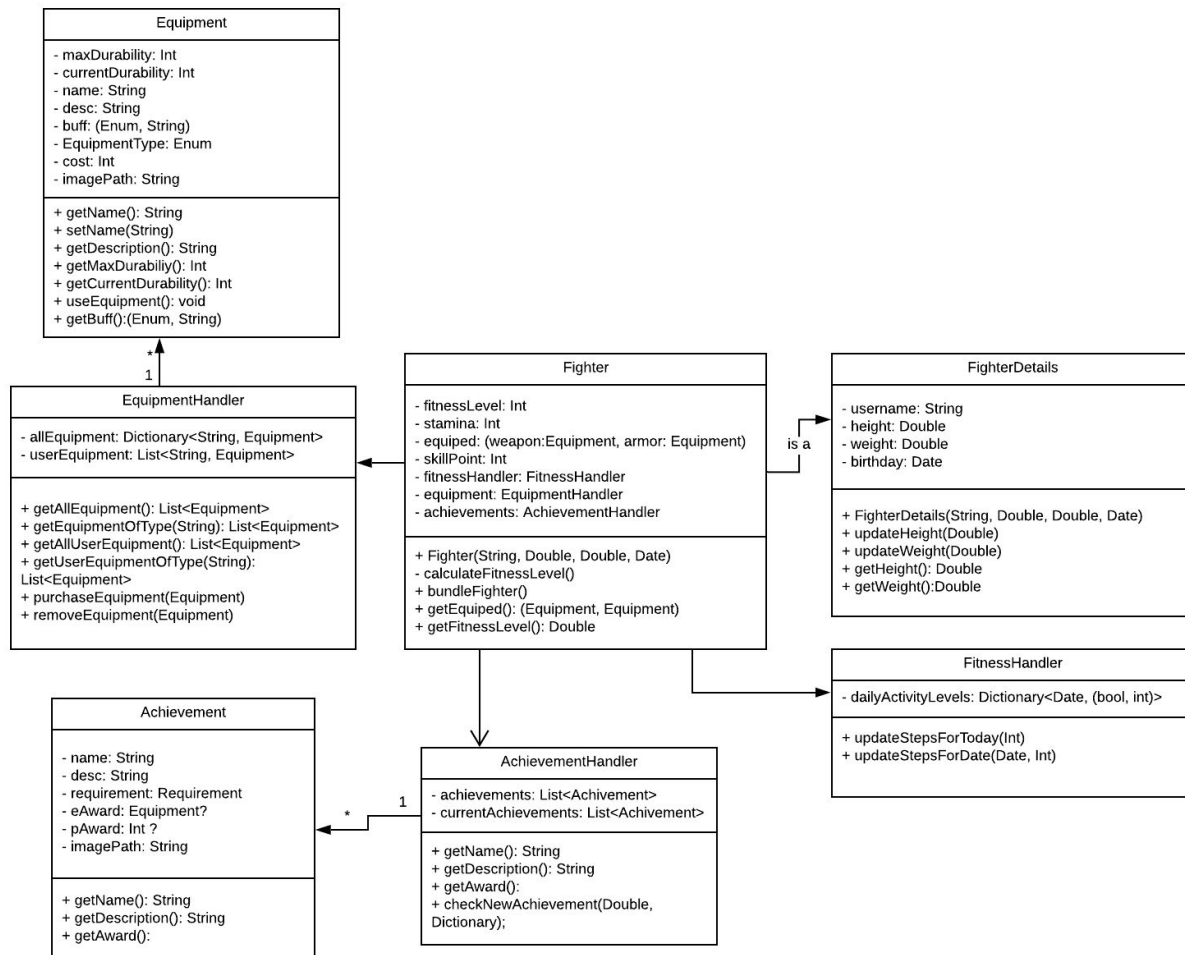
- AlertHelper
  - Helper class for creating and displaying alerts to the user
- DismissKeyboardDelegate
  - Delegate for TextInterface to handle dismissing the keyboard
- EquipmentPickerViewDelegate
  - UIPickerViewDelegate that takes in a list of equipment and displays their names in the picker view
  - Has a function for getting the name of the currently selected equipment for use by the FightController to simulate fights
- InputValidator
  - Helper class to validate UITextFields and change their background color accordingly

#### Data

- Achievements.plist
  - List of all the different achievement for the user to achieve
- Shop.plist
  - List of all the equipment available for the app. Loaded into the app and contained in the EquipmentHandler
  - Made available to the user via the Shop tab
- UserData.plist
  - Temporary data to store the user data

## Starter UML

This UML is how we started our app and the descriptions above is what we actually ended up with.



## Resources

### Frameworks

- Core Bluetooth
- Core Motion (didn't end up using but code still exists)
- HealthKit

### Tutorials and Resources Used

- CoreMotion
  - <https://wysockikamil.com/coremotion-pedometer-swift/>
- HealthKit:
  - <https://github.com/dwyl/learn-apple-watch-development/issues/43>
- Bluetooth:

- <https://itnext.io/whos-there-simple-swift-app-for-chatting-with-colleagues-around-a88ff765736f>
- Icons:
  - <https://www.flaticon.com/>