Design Patterns that could make our system more flexible and more extensible

We could create a class called Activity that would implement StatData. This class would act as a super class to each activity (the only activity we have so far is running). This follows a design pattern called the Factory Method. This method would allow us to add more activities easily. We could also have a map in this class that would hold stats for every activity.

Another possibility for the design pattern of our activities could be a Decorator Pattern. Since dome users may not want to display some of their statistics, we can use a Decorator pattern that could allow a them to add statistics to an activity as they please.

Creating a class responsible for calculating statistical averages helps to make each class responsible for an individual task.

Delegating the task of editing stats to a class called StatEditor that will be responsible for changing desired statistics.

Delegating the responsibility of editing a profile to a ProfileEditor class helps to achieve cohesion.

Delegating the responsibility of viewing a friend's statistics to a StatsViewer class helps to make each class responsible for individual tasks.