**Description of updates in UML diagram**

* The biggest change in our new UML diagram is the introduction of an interface called “Activities”. This interface seeks to remove one limitation of having a single class with all types of data to be stored (former StatisticalData class), that is - the redundancy of excess and unnecessary information needed when we want to add a new type of activity e.g. skiing, rowing etc which have their own exclusive attributes on top of not having all the attributes from running activity. Now, using this interface, it can be implemented by different classes for different types of activities and avoid this problem by using different sets of attributes for each.
* The aforementioned step results in significant changes in the StatisticalData class. Now, this class is only used for calculation of averages of values of attributes and doesn’t contain the attributes of activities anymore.
* Another update in our model is the introduction of a new class called CreateGraph. This will be used for drawing line graphs from the relevant data from Profile class and activity classes e.g. Running.

**Explanations of Relations**

* We can observe that all classes except the activity-related ones and the interface have association with the Profile class. This makes sense because all data and methods need to be connected to one single profile which belongs to the user.
* The Login class prompts the app’s functionality by allowing the user to log in. This class can hold only one object of Profile type at a time.
* ImportData class imports the relevant data from the device e.g. fitbit watch and stores it in the user profile. An user is allowed to do unlimited numbers of imports, so the Profile class can hold an unlimited number of ImportData object.
* The Friends class holds the list of user’s friends who he is connected to. It allows user to add or remove a friend. It also allows user to view his friends’ stats. The Friends class can hold unlimited numbers of Profile objects.
* The StatisticalData class allows user to do calculations of averages of attributes related to the activities. The Profile class can hold an unlimited number of StatisticalData objects.
* The Profile class is associated with the Activities interface which is implemented by the Running class. Further activities added as classes will also implement this interface.