Use case Description

Add a tracker

Precondition

1. User has created an account.
2. User has logged into their account.
3. User has selected tracker tab.
4. (User selects add a tracker option from the tracker page.)

Flow of Events

1. User selects add a tracker option from the tracker page.
2. List of trackers that it is possible to import data from is shown to user
    a. If the user selects an item from the list, they are redirected to an authentication page.
    b. If the user selects cancel, then the user is asked to confirm their decision, after which they are returned to the tracker page or prompted to select a tracker from the list.
3. Once on the authentication page the user will be asked to give permission to allow the app to access their data from the selected tracking service.
    a. If the user accepts, they are shown a “gathering information” page
    b. If the user rejects the authentication they are returned to the list of trackers.
4. Request is sent to the service the user selected
    a. If the request is declined/fails user is shown a box which tells them the data import failed, and to try again later, then they are returned to the select tracker screen.
    b. If request is accepted, then a token is returned which is used to get data using the chosen services api.
5. Request made to selected service.
    a. If errors occur with the requests they are tried again x times, if continued errors the process will be cancelled and the user will be returned to the select a tracker screen and shown a message telling them an error occurred.
    b. If no errors occur, then the system continues.
6. The data collected is put into apps databases etc to be viewable by the user in the app.
    a. The data will be handled differently depending on the app it was collected from.
7. User data stored on the cloud gets updated.
8. Once the process is complete the user is returned to the select tracker page and told that the import was a success.

Post Conditions

1. System updates the user’s information, using the new data imported from the added tracker, if a tracker was added.
2. The system returns to an idle state and waits for next input.

Actors

The user is the main actor in this use case as they will initiate the case by selecting the tracking tab in the app, and the tracking application that the data will be collected from. Another actor will be the cloud as the user data will be updated if the user makes any changes.

Scenarios

1. User A selects “Add a tracker” option from the tracker page and is prompted to select the tracker that they would like to import data from.
   They select Fitbit from the list of options.
   The user is redirected to the Fitbit OAuth 2.0 authorization page where the user selects “allow all” to give the app permission to collect all of their data from Fitbit.
   HTTP request is made to access the user’s data.
   User data is pulled from the server and imported to the in-app tracker.
   Data is added to local databases for use by other parts of the app etc.
   Data uploaded to cloud database.
   user is shown a message confirming their data has been imported and is returned to the selection screen.

2. User b selects “add a tracker” option from the tracker page in is prompted to select the tracker that they would like to import data from.
   They select Fitbit from the list of options.
   The user is redirected to the Fitbit OAuth 2.0 authorization page where the user selects “deny all”
   User is sent back to the tracker page.
   System waits for user input.