BACK-END TESTS:

2nd of March

Data: First API testing - Facebook.

Description: Hard Authentication with special credentials in order to test the API of Facebook.

Status: Some problems with scopes but it ended well.

Data: First API testing - Gmail.

Description: Hard Authentication with special credentials in order to test the API of Google/YouTube.

Status: Some problems with scopes but it ended well.

Data: First API testing - Trello.

Description: Hard Authentication with special credentials in order to test the API of Trello.

Status: Some problems with scopes but it ended well.

3rd of March

Data: First API testing - Twitch.

Description: Hard Authentication with special credentials in order to test the API of Twitch.

Status: Some problems with scopes but it ended well.

Data: First API testing - Discord.

Description: Hard Authentication with special credentials in order to test the API of Discord.

Status: Some problems with scopes but it ended well.

Data: Effective test of action-reaction system.

Description: Testing a hard coded script in order to test out the API's reactions.

Status: Signals are received for both action and reactions.

Data: Effective API tests – Gmail.

Description: Testing the Gmail/YoutTube services with both action and reaction of this services.

Status: Emails are received and sent well – Youtube needed some elbow grease but it works now.

Data: Effective API tests – Facebook.

Description: Testing the Facebook services with both action and reaction of this services.

Status: Weird token refreshment with Facebook, making it awful to test multiple times the API but it ended up working.

Data: First API testing - Github.

Description: Authentication with special credentials in order to test the API of Github.

Status: Github's API is so well documented that it worked pretty much instantly.

4th of March

Data: Effective API tests – Trello.

Description: Testing Trello with both action and reaction of this services.

Status: Tokens of Trello seem to last for ever (figuratively). Nothing is given in the header and everything in the query which is weird. I'm going to use this one as default to tests most action/reaction for it's very easy to interact with.

Data: Effective API tests – Twitch.

Description: Testing the Twitch services with both action and reaction of this services.

Status: Jeremy just started his Streamer's career on Twitch. All results have an extra "data" member because of Twitch's response policy. Hopefully we don't need to start a stream to test the follow count. Everything works ok for now.

Data: Effective API tests – Github.

Description: Testing the Github fork and issue of this services.

Status: The issues were easy, just needed to specify a few things about the name and keyword specifications. The fork was harder, we need to test it on small repository because it can be a bit long. Works fine now.

Data: Effective API tests – Discord.

Description: Testing the Discord action and reaction of this services.

Status: We need to change the way we planned the Discord Service. After checking for one of our Authorization errors we learnt we couldn't do much without a Bot Token. We thought about using a Bot of our own design, but it wouldn't be able to do much within the logged in account. We can't make a good reaction with what we have.

Data: Supplementary tests – Facebook.

Description: Adding another reaction for Facebook since Discord cannot work properly.

Status: We added a "story" option on Facebook. Since Facebook was working fin already, we were able to implement it without issues.

5th of March

Data: Tests of compatibility with the prototype front.

Description: Testing getting Reactions, Actions, and Scripts from fake front.

Status: Just tried to test our compatibility with a front (made some adjustment to the "requests" file on front which makes compatibility possible).

Data: Implementation of our back on to the host.

Description: Just putting our work on a host to test compatibility with real fronts.

Status: This still needs some work; flutter front seems to be doing fine in the authentication and getting other Object add little changes to be able to work but react needs a bit of elbow grease.

6th of March

Data: Testing the "cookies" issue we had with the authentication on several services on flutter.

Description: Flutter team complained about an issue with cookies, we just need to know if it comes from out end.

Status: There was an issue specifically on mobile devices, but it wasn't exactly about cookies. Correlation not causation. The issue was about how the user_id was saved on front side (not us then). But we still changed a few things for it to be simpler.

Data: Correction compatibility problems with react.

Description: Correcting some problems between back and react front (not sure if it's on our side).

Status: There is not much to do, just a few minor issues regarding error handling and script editing. No real work here in the end just needed to sort out the specifics about the way our body's content was spelled and specify the status of our API calls to handle errors.

7th of March

Data: Adding and testing Notification system.

Description: Adding something to notify the user in case a reaction trigger.

Status: We have some extra time. So, might as well have some fun on our front side with notifications. Ended by adding it to our back for it to be able to work on both flutter and react. Seems to be working fine in the backend at least.