17 Nov 2022

Draft 1: Activity Diagram

Problem:
To acquire new data from the pubg API, the user must navigate between various files and run the program(s) multiple times. If new data is found, it is successfully stored, but does not intuitively and automatically show up in the IDE.

Solution:
Create an activity diagram showing the sequence of events from when a user asks for information about a player, all the way to when the user can access that information (or receives an error message). Deliberately address problem-points.

Solution Requirements:

1. The user should be asked for minimal input.

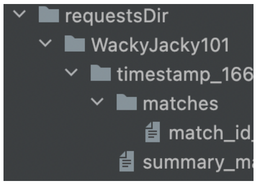
2. Upon receiving a valid request and valid response from the API, the program should automatically save the response to a file that is visible in the IDE project files.

3. Further, the user should not have to manually state that they want to also store the telemetry corresponding to the match_ids listed within the response file. Instead, given a valid response file, the program should attempt to get and store telemetry data for each match within the response file.

4. The visual representation should make clear who needs to do what, and when they need to do it. The get data portion of the project will be adjusted accordingly.

Solution Requirements (Status):

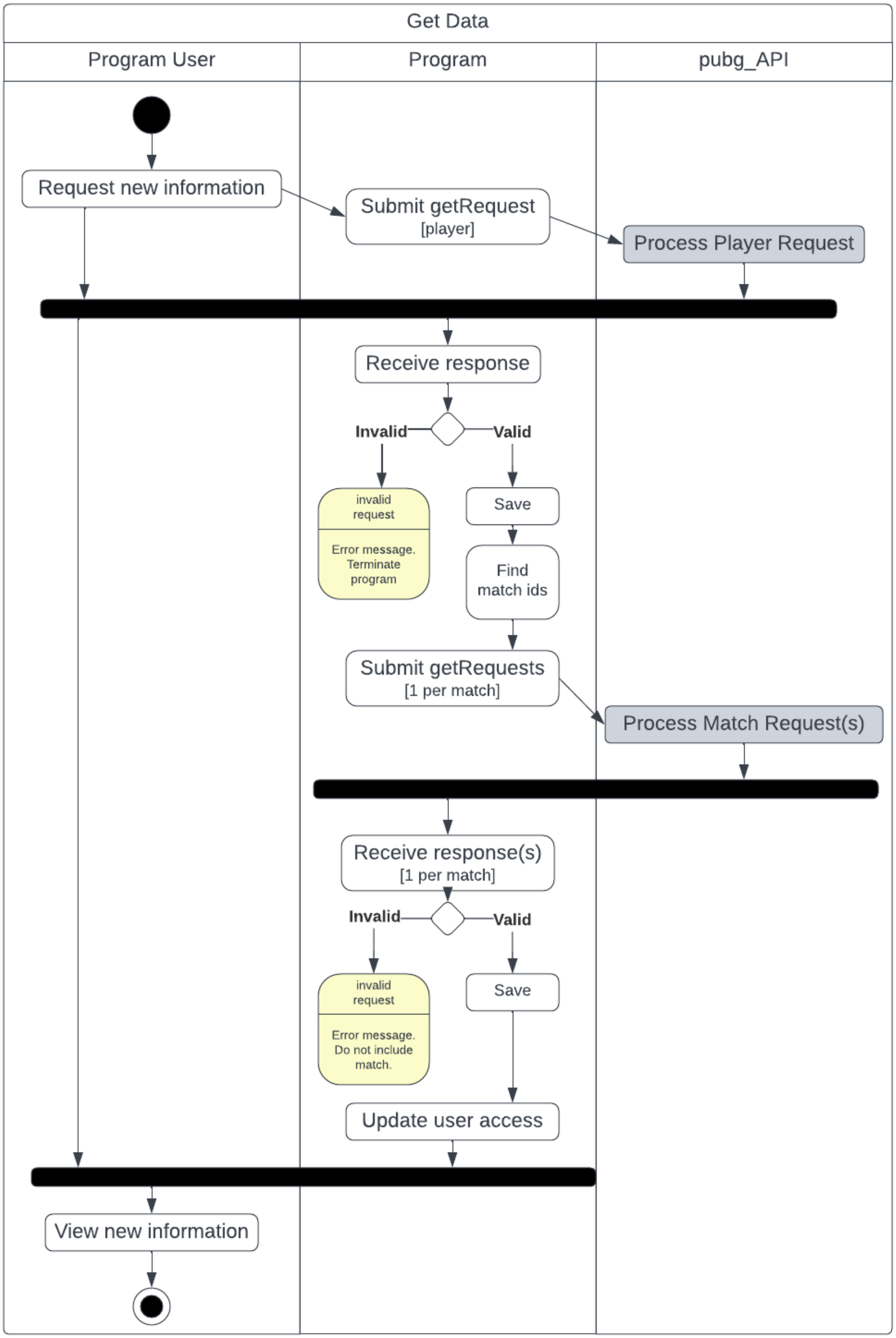
1. User input: player name.
2. Valid request and valid response now lead to automatic file saving and visible display.
    a. Example:



```
v requestsDir
  v WackyJacky101
    v timestamp_166
      v matches
          match_id
        summary_ma
```

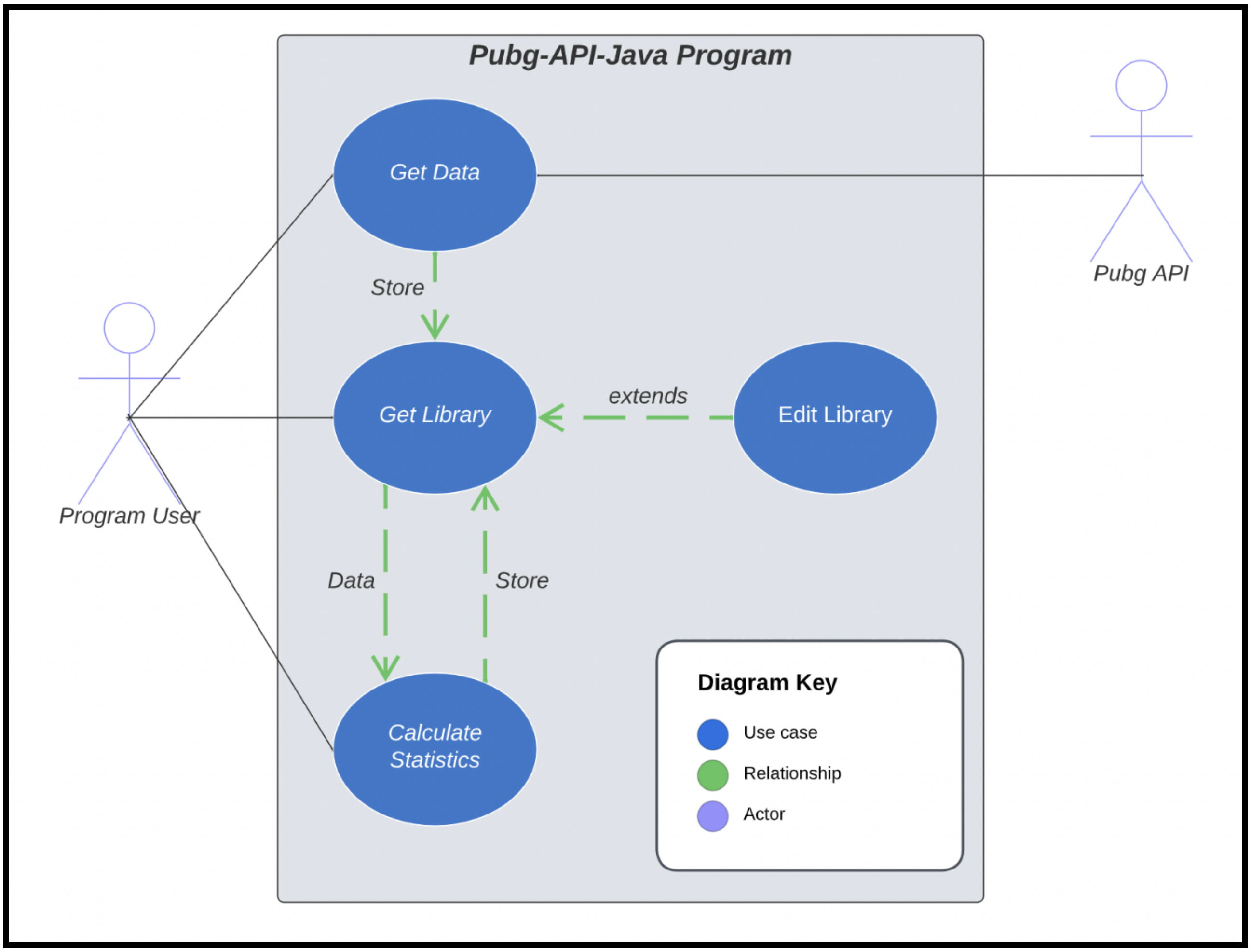
3. Automated.
4. Swimlanes for clear responsibilities and dependencies. Project is being updated.

Draft1 - Activity Diagram: Get Data



Note: “Get data” expands on the provided Draft1 - Use Case Diagram.

Draft1 - Use Case Diagram (Basis for Activity Diagram)



GitHub Link:
https://github.com/JS1936/PubgAPI-Java/blob/work2/src/main/resources/Drafts/Draft1-%20Use%20Case%20Diagram%20-%20Pubg-API-Java%20Program.pdf