

## **Axel Nilsson Work Diary**

### **14 February 13:00 -- 16:00 (3 hours):**

Set up the project directory through Stack and tested out importing external libraries  
Use 'stack build' then 'stack ghci ./app/Main.hs' to load imported libraries

### **15 February 14:00 -- 17:00 (3 hours):**

Got comfortable with Stack, imported Data.Aeson, Network.HTTP and removed Text.JSON.  
Added comments, references, a to-do list, and instructions on how to properly use Stack  
(install libraries, etc.)

### **16 February 11:00 -- 14:00 (3 hours):**

Meeting with Ghafour. Talked about plans for the project and what needs to be done. Talked about creating the first 2 pages of our report, (names and introduction). I then researched how to retrieve information from JSON files.

### **18 February 15:00 -- 18:00 (3 hours):**

Added some libraries and started working on parsing JSON from URL with Pavlos and Johan. We succeeded in calling the web API functions through our haskell program.

### **21-22 February 22:00 -- 02:00, 12:00 -- 16:00 (4 + 4 hours):**

Researched and messed around with JSON parsing. Created instances for FromJSON to be able to parse our custom data types, i.e 'JSONResponse'. We are now able to extract certain parts from the parsed JSON from our main function.

### **23 February 12:00 -- 17:00 (5 hours)**

Researched the different libraries and made the software able to return a list containing a users' owned games when provided a valid steam64 address. Separated the program into 4 parts, Main.hs, JSONParsing.hs, IDS.hs and KEY.hs

### **24 February 10:00 -- 14:00 (4 hours)**

Tweaked the returnFromJSON function so the returned value can be saved to pure variables for future comparisons. I also watched some youtube videos and read some documentation on Data.List and our other libraries to understand them better.

### **26-27 February 21:00 -- 04:00 (7 hours)**

Sat up until late and finished some tasks.

1. Finished some code that Johan had started working out which takes the retrieved list of common games and prints each game on its own line in a dynamically created CommonGames.txt file. I also fixed some errors where the .txt got corrupted because it didn't recognise certain characters
2. Added JSON parsing support for retrieving steam user summaries, including their platform alias which makes the software more user friendly and more comprehensible for users
3. Tidied up Main.hs and made it more comprehensible and readable, added comments and some new libraries, "System.IO" and "GHC.IO.Encoding"
4. Researched how to run the software as a .exe, go to .stack-work\install\94677cd0\bin where you will see a CommonSG-exe.exe

## 5. Fixed / Added diary entries

### **1 March 10:00 -- 18:00 (8 hours)**

Added playtimeFromJSON function and kept working out a solution for adding each users' playtime / game to the .txt file. I managed to create a custom intersect function for finding games users have in common while preserving their individual playtime on each of the games. I then created some auxiliary functions for combining these lists to the form of "gameName -- user1: x hours, user2: y hours"

### **2 March 9:00 -- 16:00, 21:00 -- 01:00 (7 + 4 hours)**

I continued with trying to implement playtime into the main function for an unlimited amount of users and managed to build it into our executable. I added some comments, specifications and fixed some run time errors regarding printing unsupported characters, and then sat with Johan and Pavlos and updated our report and our flowchart for the project. We had a meeting with Ghafour and I then worked on writing specifications. Later that night I continued with writing function specifications and some more on the report.

### **3 March 10:00 -- 13:00, 15:00 -- 17:00 (3 + 2 hours)**

I finished with writing specifications and examples for all of our pure functions and monad/IO functions. I then wrote Test cases using HUnit with Johan and worked on finishing the report, and later added some runtime instructions for running our software inside Main.hs and wrote down some code references. I finished this diary entry.