Sprint plan 4

Context Project: Health Informatics

Group: HI4

User Story	Task	Task Assigned To	Estimated Effort per Task
The user wants to see the results of the files that are read or analyzed.	- Extend the GUI so that the output is shown (responsibility: make sure that it's linked with the extended writer, fully tested, that it's visually accepted by the user and merged with the master branch on time) - Format the output in the right way - Update the writer so it can use a TreeSet (responsibilities: make sure that it's implemented, fully tested and merged with the master branch on time)	Remi Elvan Elvan	3 3 3
The user wants to add labels to created records, so analysing is easier by mentioning labels instead of records.	- Create a mechanism to add labels to only the data records that comply with the condition. (responsibilities: make sure that it's implemented, fully tested and merged with the master branch on time)	Matthijs	6

The user wants to perform computations (such as counting, summation or statistical operations) on the data for exploratory analysis.	- Extend language construct for computing (responsibility: make sure that it's included in our report of the scripting language) - Implement (script) code for computations on the data (responsibilities: make sure that it's implemented, fully tested and merged with the master branch on time)	Elvan Elvan	2 5
The user wants to know how to use the scripting language	- Create examples and descriptions in a manual (responsibilities: make sure the document is created and the basics of the language are explained)	Sven	5
The user wants to perform data transformations by entering a scripting language in a script editor	- Improve the script editor by implementing the RichTextFX library (responsibility: make sure that it's implemented, fully tested and merged with the master branch on time) - Improve the script editor visually (responsibility: make sure that it's lay-out appeals to the user) - Display which data is available in a tree list view	Remi Remi Sven	6 2 4
The user wants the scripting language to be understood by the program in order to	- Implement basic structure for recognising the C's (responsibility: make sure that it's implemented and merged with the master branch on time) - Explain the basic structure for recognising the C's (responsibility: make sure that it's included in our report of the scripting language)	Hans Hans	6 2

	- Parse chunking (responsibilities: make sure that it's implemented, fully tested and merged with the master branch on time) - Parse constraints (responsibility: make sure that it's implemented, fully tested and merged with the master branch on time) - Detecting syntax errors before running the script (responsibility: make sure that all exceptional cases are tested and detected with an error message)	Hans Sven Hans	4 4 4
The user wants to import files with the least amount of effort	- Automatically detect column names if they are in the file - Let the user rearrange columns (and their names) by clicking and dragging	Matthijs Matthijs	4 4
The client wants an up to date emergent architecture document	- Update the architecture document (responsibilities: make sure that the document is updated, spell checked and merged with the master branch on time)	Elvan	2
The user wants a reactive GUI with error management (including dialog windows and error messages)	- Test the GUI on JavaFX Services with JUnit tests following the dependency injection pattern (responsibility: make sure that the GUI has at least 75% coverage afterwards)	Remi	4
The user wants the program to be fully tested and an explanation on warnings that are ignored	- Report why some warnings from Cobertura, PMD, FindBugs and CheckStyle don't have to be solved (responsibility: make sure this is documented in a PR)	Sven	2

Remaining Tasks

(non-sprint related tasks that are not mentioned above)

- Acceptance testing with Wenxin Wang
  On Monday (18/05) we are going to meet with Wenxin to get her feedback on the scripting language and script editor we are going to create.
- Extra room for flexibility
  We left two hours room for each group member for flexibility.

General explanation on responsibilities

- Programming tasks
  The group member who is assigned to a programming task has the responsibility to implement the corresponding feature and to fully test it (with at least 75% line coverage). After a feature is done, he should open a pull request for it and make sure that the code is approved by at least two other group members and merged with the master branch before the deadline of the current sprint.
- Documenting tasks
  The group member who is assigned to a documenting task has the responsibility to write the corresponding sections and to perform a spell check. After the part is written, he should open a pull request for it and make sure that the document is approved by at least two other group members and that all sections are merged with the master branch before the deadline of the current sprint. If the document also has to be uploaded on Blackboard, he is responsible for doing this on time.