**USER MANUAL**

**Setup**

Prerequisites:

* Matlab 12 or later
* Windows

1. Download (or clone) the program at <https://github.com/jakelamotta/dataformatter>
2. Extract the zip file anywhere on the computer.
3. Open matlab
4. Set a path to the location of the program folder
5. Done! Launch the program by typing “main” in the Matlab command window

**Usage**

**Importing data**

As a new user of ”main” the first thing you need to do is import some data, otherwise there is not going to be anything to export. Importing is done by pressing the “Import Data” button. The following window will ask you to enter a date, a Flower name and choose positive or negative. Make sure this information is correct as it will be used to identify the observation.

The next step in the import data process is importing the actual data. It is still possible to change the suggested ID.

This will load the data into a folder system which is a structure that the functionality of the program depends on.

**Loading data**

When data is imported onto the computer the user can now start to load it into “main”. The user needs to select what data type to load (only one type can be loaded at a time) and then which folder to search in. Here it is important to note that the user do not need to select exactly the folder where the data is stored as long as it is a parent folder to the datafolder. Then “main” will automatically load all data from all subfolders matching the selected type into the program.

For example if the user wants to load all data, she can just select the data folder as that is where the raw data is stored.

If the user wants all observations from one specific date she can select the folder corresponding to that date and only that data will be loaded.

**Managing data**

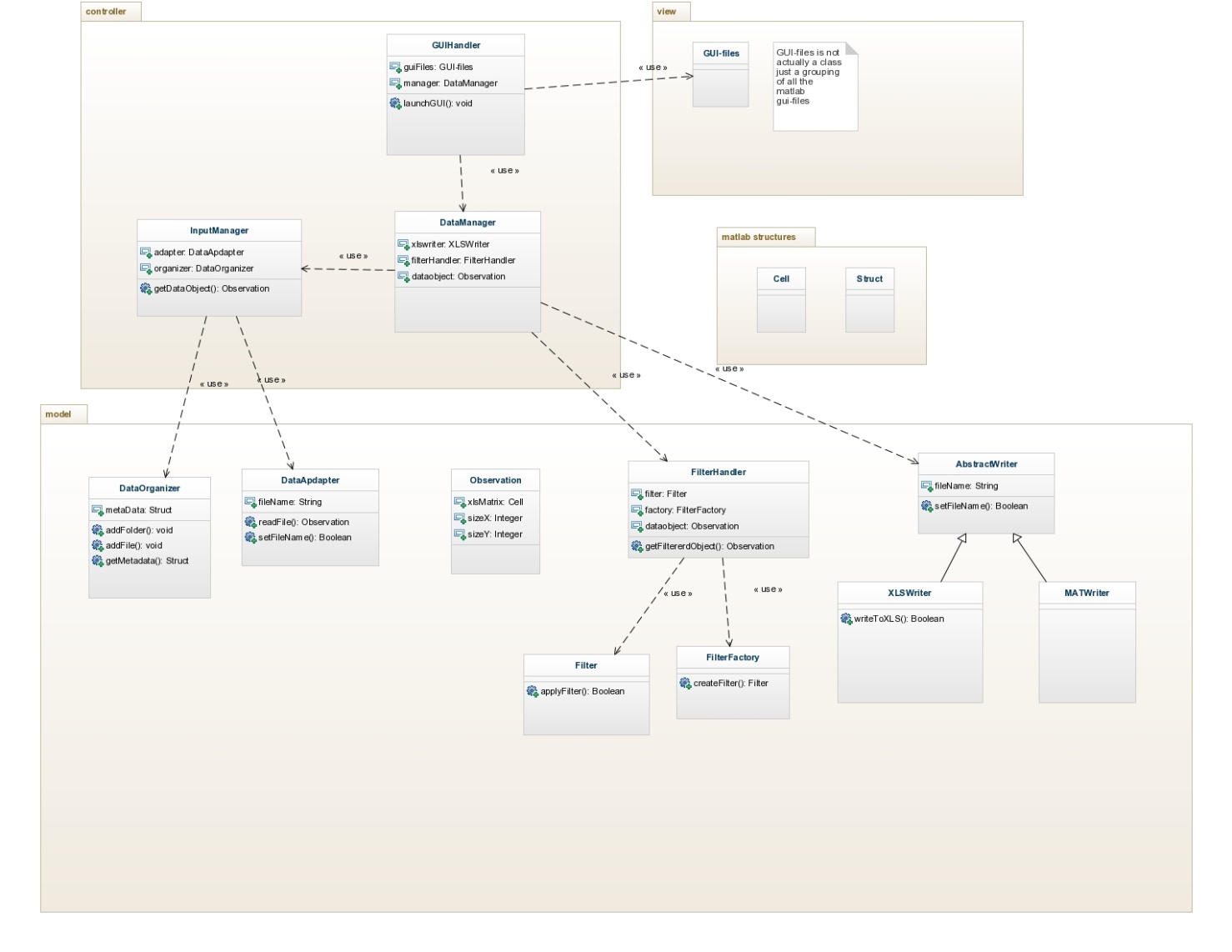
A

**Exporting**

Exporting the data to an excel file is a trivial step, as the data is loaded into the system. The user can simply press “Export” and then select a xls file to export to, this file can either be an existing one or a new file. If the file already exists the old data will not be overwritten but rather the new data will be appended to the file.

**For advanced users**

**Class tree**

****

The class diagram is from the early design of the system so it is not exactly accurate regarding methods and naming of fields and methods but the classes and interaction between classes are correct. This should provide the advanced user with a clearer picture of the structure of the program. The idea of the design is to decrease coupling between modules and classes, no data processing or similar should be done in the GUI files but rather they should just get data from the underlying data layer and present it to the user at the same time as it passes user input to the data layer.

The Observation object is maybe the most important class of the program as it is the internal representation of what an observation is. It uses a cell array to store all data, just using a matrix was considered as it would have made operations on the data simpler. But since matrices only stores integers using a cell was the only way I could keep all the information in one structure.