**Hardware/Software Mapping**

Regarding the software aspects, Farmio is to be implemented in Java, using Java Development Kit (JDK) 8. Therefore, as long as Java Runtime Environment is installed, the game can be played on Windows, Mac and Linux platforms.

To play the game, in terms of the hardware requirements, the player will need an ordinary computer with a mouse and screen.

**Persistent Data Management**

Farmio involves data storage in a filesystem rather than a database. More specifically, data will be stored in “text files” on the hard drive. As the player makes progress, planting new seeds or harvesting grown crops for instance, these text files will be updated. Thus, in the next run, objects can be initialized properly by reading these saved information.

Rendering the map of the game relies heavily on these data since there will be many blocks of soil (i.e. farm slots) with possibly different seeds at different conditions. Any changes to these farm slots are supposed to be reflected in these text files.

Additionally, as the player may maintain multiple games with each having a different progress, there will be separate text files for different games. Depending on the player’s choice about whether to start a new game or load an existing one, the text file(s) to read and write will vary.

Finally, the player’s preferences are kept in text files, too.

**Access control and security**

Farmio does not have any network-related uses, meaning that there will not be any outside users other than the user(s) of the computer in which it is stored. As a result, no measurement regarding access control and security needs to be taken.

**Boundary conditions**

***Initialization***

The game is to be launched using a “.jar” file. It will not require any prior installation.

***Termination***

To terminate the game, the player is supposed to switch to the Pause Menu. Then, after choosing whether to save the game or not, the player will be able to quit the game.

***Failure***

To prevent the unfortunate effects of any possibly failures, reading / writing the text files should be given the proper priority. In other words, in case the game fails and needs to be started again, the text files should have already been updated very recently. Therefore, when the game crashes somehow, in the next run, the most recently saved configuration of the game can be loaded.

Note that this will not cause any conflict regarding user-determined game saving. That is, regardless of the player’s choice, all the changes are constantly saved in text files. Then, when the player wants to exit the game, these information is either to be deleted or kept as a “saved game” for following use.