Group 29 – Microbiology Lab Information Management and Visualization System

Benjamin Vogel, Brittany McPeek, Samuel Jungman, Rob Reinhard, Kyle Gansen, Ben Alexander

Background of Technical Problem

- Our project focuses in on data importation, synthesis, storage, and exportation
- Our project also is meant to be a useful tool to individuals that are not as tech-savvy
- Consequently, our program needs to be very adaptable and easy to use in a variety of ways if it’s to be at all helpful
- That means we need to create tools to accomodate all data collection methods currently in place within the micro bio lab that acts as our client.
- Bottom line, our program has to service multiple data types, formats, and file extensions

Technical Problem

- Data files come in a variety of different formats (.csv, .xlsx, .tsv)
- Different file formats needs to be parsed in a different manner
- The data within those files may be formatted in a variety of ways
- All of the different ways data can be organized within those files must be accounted for
- The data from each file needs to be formatted in a manner allowing it to be graphed

Possible Solutions

- Only allow the user to import certain types of files
- We will have to have separate classes for importing each file type
- For each file type, the data format can vary. Within each class, parse the data into a standard form to be graphed
- If this isn’t possible, notify the user of how the format needs to change
- Another option: for each file type, have the user specify the data format and have differing file formats

Feasibility of Solutions

- Have the user specify the data format
  - Could lead to errors if not correct
- Make it automatic but only allow the user to import certain types of files
  - Focus on most widely used types
  - Minimal setting up for client
- Have specific outputs for each data type
  - Puts importation work onto graphing component
  - Breaks our concept of isolated components
- Standardized data output
  - Cleaner interaction between two components

Final Technical Solution

- Create individual classes for each different data type/format
- Have one superclass that each class will inherit from
- The other modules will call the superclass method
- Each individual import for the different data types will return a unified data structure to pass to the graphing module

Ramifications

- Every time we want to import a new data type we will have to create a new class specific to that data type
  - Requires maintenance
  - Not very scalable
- Makes our system more modular
  - Easier to test and find bugs
  - Easier for a large group to work on
  - Easier to understand