

Team Sirius

NVIZION Application

(Phase II Peachy Galaxy Application)
Collaboration Project with Schulich School of Medicine and Dentistry

Ryan Divigalpitiya

(Project Manager / UI & UX Lead / Graphing Lead)

Byron Roby

(Data Loading Lead / Error Processing)

Gustavo Andres Murcia

(UI & UX / Graphing / Error Processing)

Sam Ali-Mirsalari

Raman Here

(Graphing / Session Saving Lead)

(Session Saving / Technical Writer)

Wen Chang Pu

Zilin Xie

(Liaison / Session Saving)

(Installations Lead / Technical Writer)

Contents

1 — Requirements	1
1.1 — Enhanced Requirements Description	1
1.2 — Software Requirements Specification Sheet	3
2 — System Testing	4
2.1 — Approach	4
3 — System Design	4
3.1 — Use-Cases	4
3.2 — Original Class Diagram	9
3.3 — Enhanced Class Diagram	9
3.4 — Sequence Diagram	9
3.5 — Package Diagram	9
4 — Development Plans	9
4.1 — Deliverable 1 Development Plans	10
4.2 — Deliverable 2 Development Plans	11
4.3 — Deliverable 3 Development Plans	12/13
4.4 — Deliverable 4 Development Plans	14/15
4.5 — Deliverable 5 Development Plans	16/17

5 — Lessons Learnt	18
5.1 — Git	18
5.2 — Qt	18
5.3 — C++	18
5.4 — Team	18
5.5 — Value	19
5 — Singleton Pattern	19
5.1 — Single Pattern	19
5.2 — Prototype Pattern	20
7 — User Guide	21
7.1 — Starting Screen	21
7.2 — New Session: The Dashboard	21
7.3 — Load Files	22
7.4 — Filters & Sorts	24
7.5 — Visualizations	25
7.6 — PDF Export / Print	26
7.7 — Session Saving	27
8 — Collaboration View (Continuation of User Guide)	28
8.1 — Definition	28
8.2 — Possible User Actions within View	29
8.3 — Additional Possible Actions within View	30
8.4 — Warning about Large File Computation	30
8.5 — Demo of Large File Visualization	30

Software Requirements Specification Sheet

Manual Test Matrix

Use-Case Diagram

Class Diagram

Package Diagram

Sequence Diagram

Web Graph Algorithm Description & Explanation

Timelines & Agent-Task Matrices

1 — Requirements

1.1 — Enhanced Requirements Description

Here follows our list of enhanced requirements, both mandatory and stretch, that have been implemented for the Final Submission.

FILE DETECTION & AUTO-PLACEMENT [Optional — Origin: Team Sirius]

REQ ID #1.1 — File Detection

Once the user chooses a file (or multiple files as specified in REQ ID #2.1) to load into the program, the software will detect the type of activity of the file(s). There are four possible file types; Teaching, Publications, Presentations, and Grants and Clinical Funding. The user will not be required to manually navigate to the corresponding activity tab of the file before loading the file into the program.

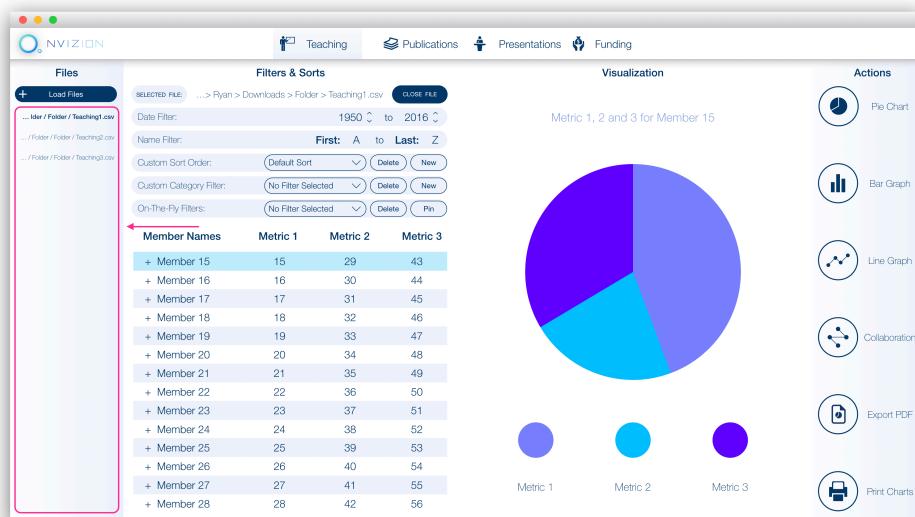
REQ ID #1.2 — Automatic Placement

When the file is loaded and correctly categorized using our file detection feature, it will automatically be sorted into the corresponding activity tab in the dashboard.

MULTI-FILE LOADING [Optional — Origin: Team Sirius]

REQ ID #2.1 — Multi-File Loading

When prompted by the File Explorer to select files to load into the program, user is able to select multiple CSV files. The user can do this by either ctrl + left-clicking (Mac: Cmd + left-click) or simply dragging their mouse over all CSV files that are to be imported. If multiple files are loaded into the software and there are multiple files that are the same activity type, the most recent file that was loaded for that activity type will show up. This is to be changed for the Final System Delivery (Deliverable 5)



where multiple files of the same activity type will be displayed in a left side-bar as illustrated in a mockup below (refer the red arrow/selection):

[Optional — Origin: Team Sirius]

REQ ID #4.1 — Transition Old UI to New UI

Converted the entire UI to much more modern format. The newer UI is better suited to implement and visualize all the new requirements needed.

REQ ID #4.2 — Redesign UI to Accommodate 3.0

The UI redesign was focused to suit the changes made in 3.0. The multi-file loading meant multiple files could have errors so navigating effectively through them was a necessity. Also needed to be informative as to which files are missing entries.

REQ ID #4.3 — Listable Files per Activity Tab

With multi-file loading their needed to be a way to clearly and neatly represent all the files loaded and an easy way to access them and close them. Having a panel on the left with the files loaded to the certain tab make it easy to switch and close any files not needed.

ALL TEXT FIELDS EDITABLE BEFORE IMPORTING [Optional — Origin: Team Sirius]

REQ ID #4.1 — Universal Editable Fields

Before importing the data files the user is able to edit any text fields not just the ones that are empty.

MAC OS DRAG & DROP INSTALL FILE [Optional — Origin: Team Sirius]

REQ ID #10.0 — Mac OS Install File

An installation file of the type DMG to install the application on Mac computers. The file just needs to be opened and app dropped into Applications folder, much like most other Mac applications.

ENHANCED ERROR CORRECTION [Mandatory — Customer Requirement]

REQ ID #3.1 — Error Count Per File

When a new CSV file is loaded, the error dialog box pops up if one or more files have errors. The dialog box has the added functionality of displaying the total number of errors in a single file. Errors decrease when errors are fixed by the user.

REQ ID #3.2 — Error-Filled Multi-File Navigation

When error-filled CSV files are loaded, the error edit dialog box provides a way to navigate between the multiple files in order to edit their errors

REQ ID #3.3 — Erroneous Entries Navigation

Find Next / Find Previous buttons which jump to the next / previous erroneous record are present in the error editing dialog box upon loading erroneous files.

REQ ID #3.4 — Navigation Control Hiding

Once all errors have been fixed, navigation controls for navigating erroneous records will hide themselves.

REQ ID #5.2 — Line Graph

User now has the option to visualize selected data with a Line Graph.

REQ ID #5.4 — Collaborations Web Graph

User now has the option to visualize collaborations between authors in a publication file.

REQ ID #6.1 — Session Resuming

User now has the save session when user attempts to quit the app.

REQ ID #6.2 — Choose between Starting New or Loading Session

User now has the option to a saved session in addition to starting a new session.

REQ ID #5.4 — Collaborations Web Graph

User now has the option to visualize collaborations between authors in a publication file.

REQ ID #8.1 — Collaborations Web Graph

User has the option to sort by a user selected list

REQ ID #9.1 — Universal Editable Fields

User now has the ability to edit all fields before importing them.

Below is our Software Requirements Specification sheet that details all remaining requirements to be implemented in addition to the ones that have been implemented for Stage 1 and Stage 2 thus far. (**A larger, more readable version is available in the appendix**).

1.2 — Software Requirements Specification Sheet

ALL ENHANCED REQUIREMENTS		REQ ID	STATUS	REQ DESCRIPTION	SUPPORTING EXAMPLE	ORIGIN
FILE DETECTION & AUTO-PLACEMENT		1.0	✓			Team Sirius
File Detection		1.1	✓	Software detects the type of activity of file out of the four possible activity types when loading a file		Team Sirius
Automatic Placement		1.2	✓	When file is detected and loaded, the software automatically places it under the right activity tab		Team Sirius
MULTI-FILE LOADING		2.0	✓			Team Sirius
Multi-File Loading		2.1	✓	User is able to select multiple CSV files in the File Explorer and successfully load files	Refer to Updated UI Mockup	Team Sirius
ENHANCED ERROR CORRECTION		3.0	✓			Mixed
Error Count per File		3.1	✓	When the CSV file is loaded, the error dialog box should display total number of errors	Refer to Updated UI Mockup	Customer (Required)
Error-Filled Multi-File Navigation		3.2	✓	Next/Prev button navigates to files with errors	Refer to Updated UI Mockup	Team Sirius
Erroneous Entries Navigation		3.3	✓	"Find Next/Prev" button which jumps on to the next/prev error must be present	Refer to Updated UI Mockup	Customer (Required)
Navigation Control Hiding		3.4	✓	After all the errors are fixed, "Find Next/Prev" should be disabled.	Refer to Updated UI Mockup	Customer (Required)
UI MODIFICATIONS		4.0	✓			
Transition old UI to new UI		4.1	✓	Self-Explanatory	Refer to Updated UI Mockup	Team Sirius
Redesign UI to accommodate 3.0		4.2	✓	Add in controls and text boxes	Refer to Updated UI Mockup	Team Sirius
Listable Files per Activity Tab		4.3	✓	Left Side-bar per Activity Tab listing loaded files. Click file to load to dashboard	Refer to Updated UI Mockup	Team Sirius
ENHANCED VISUALIZATION		5.0	■			Mixed
Sub-Category Graphing		5.1	■	User is able to graph sub-category data		Customer (Required)
Line Graph		5.2	✓	User has the option to visualize data with a Line Graph	Refer to Updated UI Mockup	Customer (Required)
Collaboration Web Graph for Publications		5.4	✓	User has the option to visualize author collaboration data (from publications) via a node-web visualization	Refer to Updated UI Mockup	Team Sirius
SESSION SAVING / RESUMING		6.0	●			Customer (Required)
Session Resuming		6.1	✓	User can resume the previous session and all previous loaded files and filters will be preserved	Refer to Updated UI Mockup	Customer (Required)
Choose Between Starting New or Loading Session		6.2	✓	Upon startup of application, user has choice of starting a new session or loading previous session	Refer to Updated UI Mockup	Customer (Required)
MEMBER DIVISION SORTING		7.0	■			Customer (Required)
New Category Sorting		7.1	■	User has the option to sort by new updated CSV categories		Customer (Required)
USER SELECTED LIST SORTING		8.0	●			Customer (Required)
User Generated List Filtering		8.1	✓	User has the option to sort by a user generated filter list		Customer (Required)
ALL TEXT FIELDS EDITABLE BEFORE IMPORTING		9.0	✓			Customer (Stretch C)
Universal Editable Fields		9.1	✓	User has the option to change any fields before importing them		Customer (Stretch C)
MAC OS DRAG & DROP INSTALL FILE		10.0	✓			Customer (Stretch C)

2 — System Testing

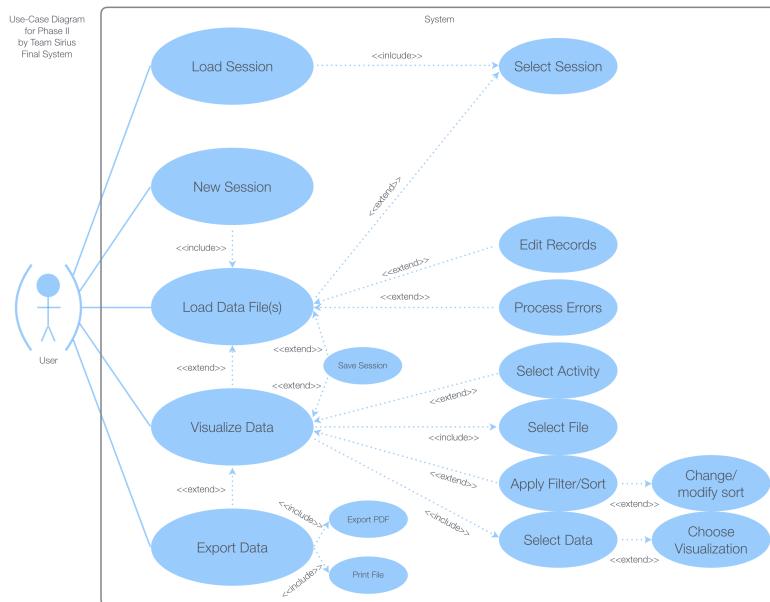
2.1 — Approach

A combination of manual and automated testing was utilized to test our demo system. For the implemented requirements, ID #1.1, 1.2, 2.1, automated tests were created for the last deliverable which tests the backend code that was written for these requirements. Unfortunately, we didn't have time to write new automated tests for the new features created for the final submission — however: Manual tests were conducted and are displayed in our manual test matrix in the appendix.

3 — System Design

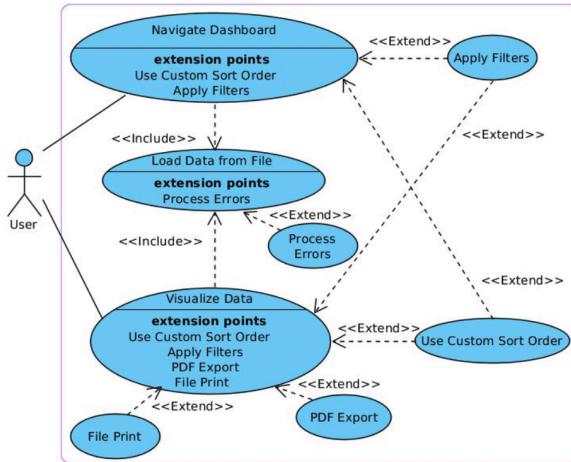
3.1 — Use-Cases

3.1.1 — Use-Case Diagram Comparisons



A larger, more readable version of Team Sirius's Phase II Use-Case diagram (pictured above) can be found in the appendix. A user refers to anyone running the program. For the Final Submission, a user has five use cases, namely "Load Session", "New Session", "Load Data File(s)", "Visualize Data", & "Export Data". Our use-case diagram differs from Team Peach's (pictured below) mainly due to our application's ability to load and navigate between multiple files of the same activity type unlike Team Peach's system. The "Load Data File(s)" and "Visualize Data" use-cases possess the most number of user interaction (as shown in our diagram above where they have the most extended and included use-cases branching off the

two) and, thus, shows us that this is where the user will spend most of their time in the application. Our redesign of Team Peach's use-case diagram illustrates this takeaway with more clarity. Shown below is Team Peach's use-case diagram with a description of their diagram for comparative purposes.



Taken from Team Peach's documentation: "Above is the Peachy Galaxy use case diagram. A user refers to anyone running the program, be it a faculty member, department manager, or anyone authorized to use the program. Each user has two use cases, namely "Navigate Dashboard" and "Visualize Data". Each of these use cases correspond to a customer requirement: Navigate Dashboard addresses the dashboard screen requirement, while Visualize Data addresses the visualizations requirement. Note that both of these use cases include another use case, "Load Data from File", which addresses the CSV file data processing requirement."

3.1.2 — Use-Case Textual Descriptions

3.1.2 — a) Load Data File(s)

Main Success Scenario:

1. The user clicks the "+" button
2. The system displays a File Explorer screen
3. The user selects CSV files (or a single file) of either different activity types and / or multiple files of the same activity type and clicks the Open button.
[Alternate Course A: File(s) is or are not a CSV type] [Alternate Course B: User clicks the Cancel button]
4. The system verifies if the records contain any missing fields [Extension Point: 3.1.2 — b) Process Errors / Records]
5. The system prompts user the ability to edit all records [Extension Point 3.1.2 — b) Process Errors / Records]

Alternate Course A:

1. The system displays an error message
2. The user closes the error message

Alternate Course B:

1. The system closes the file structure screen

3.1.2 – b) Process Errors / Records

Main Success Scenario:

1. The system displays message showing number of invalid records and prompts user to edit or discard them
2. The user clicks the Edit button. [Alternate Course A: User clicks Discard button]
3. The system displays an error processing screen
4. The user fills in all missing entries and clicks the Save button. [Alternate Course B: User clicks Previous/Next button for Errors or Files] [Alternate Course C: User clicks Cancel button]
5. The system includes the newly modified records in the data to be loaded
6. The system prompts user if they wish to edit (or go through) the rest of the remaining records in the entire file
7. User clicks the Yes button .[Alternate Course D: User clicks the No button]
8. User edits remaining records and hits the Save button [Alternate B: User clicks the Cancel button]
9. The system closes the error/records processing screen

Alternate Course A: User clicks discard button

1. The system discards records with missing mandatory entries from the data to be loaded [Return to Main Success Scenario step 6]

Alternate Course B: User clicks Next/Previous button for Errors or Files

1. The system navigates to the next or previous erroneous record

Alternate Course C: User clicks Cancel button

1. The system discards records with missing mandatory entries [Return to Main Success Scenario step 6]

Alternate Course D: User clicks the No button

1. [Return to Main Success Scenario step 9]

3.1.2 – c) Select Activity

Main Success Scenario:

1. The user selects an activity tab
2. The system updates the dashboard view with listed files in the left sidebar belonging to selected activity tab

3.1.2 – d) Select File

Main Success Scenario:

1. The user selects a file from the left file side-bar

2. The system updates the dashboard view with data displayed from the selected file

3.1.2 – e) Visualize Data

Main Success Scenario:

1. After files are loaded and the error/records processing screen is closed, the user is brought to the dashboard [Extension Point 3.1.2 – c) Select Activity] [Extension Point 3.1.2 – f) Apply Filter/Sort] The system displays Filter/Sort controls and the tabular data in the summary view
2. The user expands/collapses elements of the dashboard summary view
3. The system displays the expanded/collapsed elements of the dashboard summary view
4. The user clicks on an element in the dashboard summary view
5. The system displays a visualization (default is Pie Chart) of the selected element. [Extension Point: 3.1.2 – h) Export Data. Applicable to step 9]
6. The user clicks on a different visualization option
7. The system displays selected visualization option

3.1.2 – f) Apply Filter/Sort

Main Success Scenario:

1. The user modifies the values in the start and end date boxes.
2. The system sets its date range according to the values in the start and end date boxes.
3. The user modifies the values in the first and last letter of member last name boxes.
4. The system sets its member name range according to the values in the first and last letter of member last name boxes.
5. The user selects a choice from the Division's drop-down menu
6. The system sets the Division's range according to the value selected
7. The user applies a sort from the drop-down menu [Extension Point 3.1.2 – g) Custom Sort Order]
8. The system applies the sort to the tabular data

3.1.2 – g) Custom Sort Order

Main Success Scenario:

1. The user clicks Create New Sort Order button. [Alternate Course A: User selects existing sort order]
2. The system displays a new sort order screen.
3. The user enters the name of the new sort order, selects the hierarchy of filters to order the sort by, and clicks the Save button. [Alternate Course B: User does not enter name] [Alternate Course C: User clicks Cancel button]

4. The system closes the new sort order screen.
5. The system adds the new sort order to the list of existing sort orders.
6. The user selects the sort order from the list of existing sort orders.
7. The system sets its sort order to the one selected in the list of existing sort orders.

Alternate Course A: User selects existing sort order

1. [Return to Main Success Scenario step 6]

Alternate Course B: User does not enter name

1. The system displays an error message.
2. The user accepts or closes the error message. [Return to Main Success Scenario step 3]

Alternate Course C: User clicks Cancel button

1. The system closes the new sort order screen.

3.1.2 – h) Export Data

Main Success Scenario:

1. The user clicks the Export to PDF button to export the visualization to PDF via the use case 3.1.2 – i) Export PDF
2. The user clicks the Print File button to export the visualization to a printed file via the use case 3.1.2 – j) Print File

3.1.2 – i) Export PDF

Main Success Scenario:

1. The user clicks the Export to PDF button
2. The system displays a File Explorer screen
3. The user selects a file path, enters a file name and clicks the Save button. [Alternate Course A: User clicks Cancel button]
4. The system exports the selected visualization type to a PDF with entered file name at the selected file path

Alternate Course A: User clicks Cancel button

1. The system closes the File Explorer screen

3.1.2 – j) Print File

Main Success Scenario:

1. The user clicks the Print button
2. The system displays the Operating System's print options and list of connected printers
3. User selects Print [Alternate Course A: User clicks Cancel button]

Alternate Course A: User clicks Cancel button

1. The system closes the Print dialog box

3.2 – Original Class Diagram (Found in the appendix)

3.3 – Enhanced Class Diagram (Found in the appendix)

3.4 – Sequence Diagram (Found in the appendix)

3.5 – Package Diagram (Found in the appendix)

4 – Development Plans

4.1 – Deliverable 1 Timeline & Agent-Task Matrix: Planned vs. Actual

4.2 – Deliverable 2 Timeline & Agent-Task Matrix: Planned vs. Actual

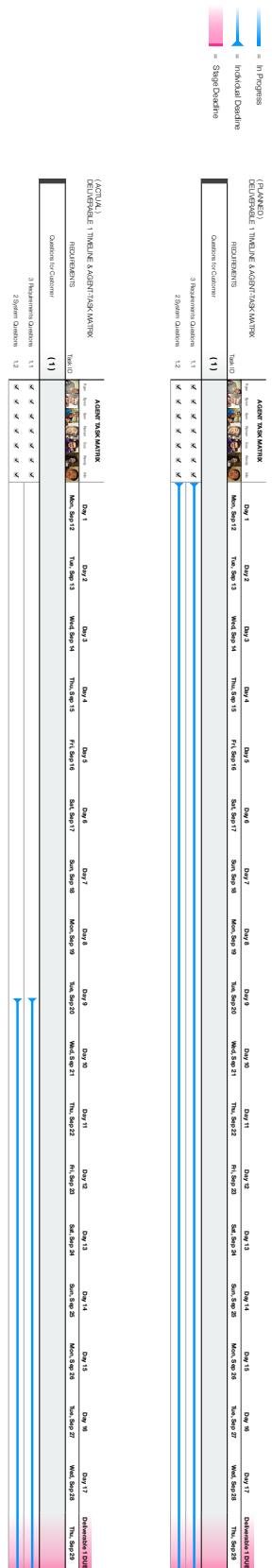
4.3 – Deliverable 3 Timeline & Agent-Task Matrix: Planned vs. Actual

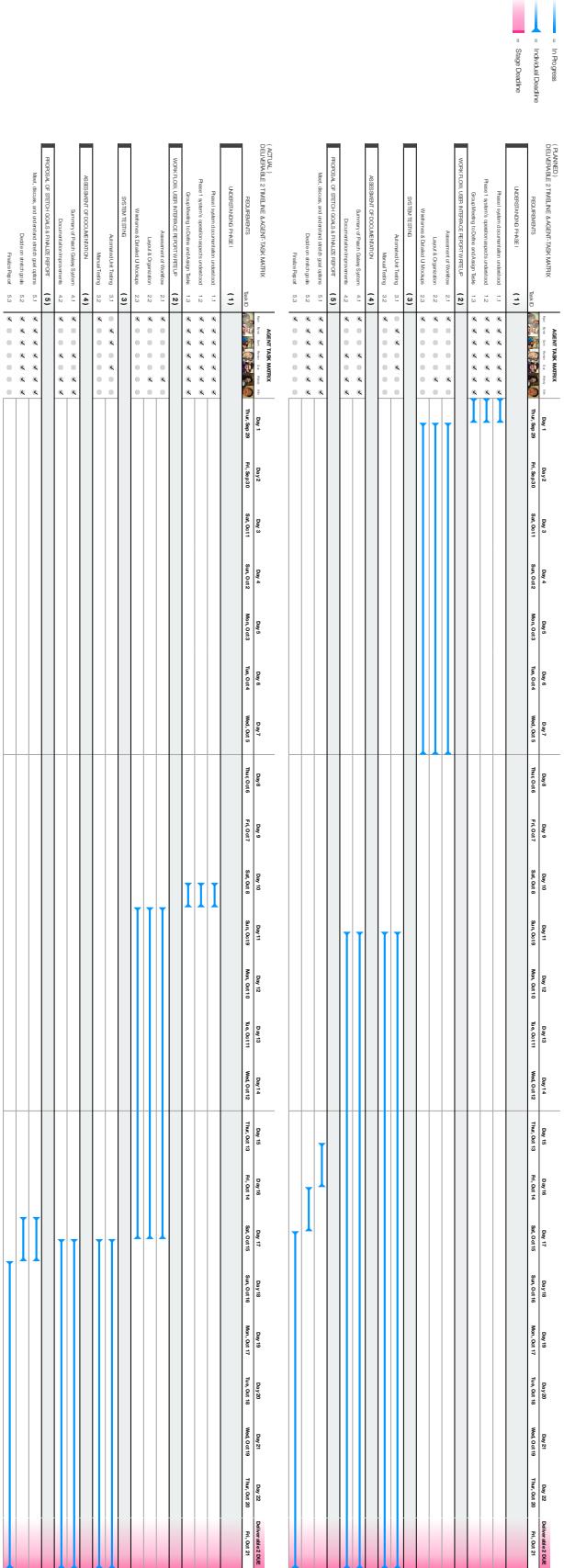
4.4 – Deliverable 4 Timeline & Agent-Task Matrix: Planned vs. Actual

4.5 – Deliverable 5 Timeline & Agent-Task Matrix: Planned vs. Actual

4.1 – 4.5 are in the next following pages

(Larger, more readable versions can be found in the appendix)





(PLANNED)
DELIVERABLE 3: TIMELINE & AGENT-TASK MATRIX

AGENT TASK MATRIX

Req ID / Task ID	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6	Agent 7	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Deliverable 3 DUE
Meeting to Determine Enhanced Req's, Task K	High	Medium	Low	Very Low	Medium	High	Medium	Mon, Oct 24							Sat, Oct 29
Assignment & Development Infrastructure Setup	Medium	High	Very High	Very High	Very High	Very High	Very High								Sun, Oct 30
File Detection & Auto-Categorization	High	Medium	Low	Very Low	Medium	High	Medium								Mon, Oct 31
Multifile Loading	Medium	High	Very High	Very High	Very High	Very High	Very High	2.1	Medium	Medium	Medium	Medium	Medium	Medium	
Auto-Categorization	Medium	High	Very High	Very High	Very High	Very High	Very High	1.2	Medium	Medium	Medium	Medium	Medium	Medium	
Multi-File Loading	Medium	High	Very High	Very High	Very High	Very High	Very High		Medium	Medium	Medium	Medium	Medium	Medium	
UI Modifications	Medium	High	Very High	Very High	Very High	Very High	Very High		Medium	Medium	Medium	Medium	Medium	Medium	
Enhanced Error Correction	Medium	High	Very High	Very High	Very High	Very High	Very High	3.0	Medium	Medium	Medium	Medium	Medium	Medium	
Individual File Error Count (UI)	Medium	High	Very High	Very High	Very High	Very High	Very High	3.1	Medium	Medium	Medium	Medium	Medium	Medium	
Start Transition of Old UI to New UI	Medium	High	Very High	Very High	Very High	Very High	Very High	4.1	Medium	Medium	Medium	Medium	Medium	Medium	
Redesign UI to Accommodate 3.1	Medium	High	Very High	Very High	Very High	Very High	Very High	4.2	Medium	Medium	Medium	Medium	Medium	Medium	
Create Automated Unit Tests	Medium	High	Very High	Very High	Very High	Very High	Very High	2.1	Medium	Medium	Medium	Medium	Medium	Medium	
Unresolved Heuristic Tests	Medium	High	Very High	Very High	Very High	Very High	Very High	2.2	Medium	Medium	Medium	Medium	Medium	Medium	
Test Matrix (2)	Medium	High	Very High	Very High	Very High	Very High	Very High	2.3	Medium	Medium	Medium	Medium	Medium	Medium	
Enhanced Req Description Writing	Medium	High	Very High	Very High	Very High	Very High	Very High	3.1	Medium	Medium	Medium	Medium	Medium	Medium	
System Testing	Medium	High	Very High	Very High	Very High	Very High	Very High	2.1	Medium	Medium	Medium	Medium	Medium	Medium	
Whitepaper	Medium	High	Very High	Very High	Very High	Very High	Very High	3.1	Medium	Medium	Medium	Medium	Medium	Medium	
Finish Off Requirements Table	Medium	High	Very High	Very High	Very High	Very High	Very High	3.2	Medium	Medium	Medium	Medium	Medium	Medium	
System Design	Medium	High	Very High	Very High	Very High	Very High	Very High	4.1	Medium	Medium	Medium	Medium	Medium	Medium	
User-Case Diagram	Medium	High	Very High	Very High	Very High	Very High	Very High	4.1	Medium	Medium	Medium	Medium	Medium	Medium	
Original Class Diagram	Medium	High	Very High	Very High	Very High	Very High	Very High	4.2	Medium	Medium	Medium	Medium	Medium	Medium	
Or Enhanced Class Diagram	Medium	High	Very High	Very High	Very High	Very High	Very High	4.3	Medium	Medium	Medium	Medium	Medium	Medium	
Development Plans	Medium	High	Very High	Very High	Very High	Very High	Very High	5.1	Medium	Medium	Medium	Medium	Medium	Medium	
Timeline	Medium	High	Very High	Very High	Very High	Very High	Very High		Medium	Medium	Medium	Medium	Medium	Medium	
Stage 1 Agent-Task Matrix	Medium	High	Very High	Very High	Very High	Very High	Very High	5.2	Medium	Medium	Medium	Medium	Medium	Medium	
Stage 1 to 2 Agent-Task Matrix	Medium	High	Very High	Very High	Very High	Very High	Very High	5.3	Medium	Medium	Medium	Medium	Medium	Medium	
Put Together Final Report & Submit All Deliverables 3 Items	Medium	High	Very High	Very High	Very High	Very High	Very High	5.4	Medium	Medium	Medium	Medium	Medium	Medium	

(ACTUAL) DELIVERABLE 3 TIMELINE & AGENT-TASK MATRIX

REQUIREMENTS

Req ID	Task ID	Mon, Oct 24	Tue, Oct 25	Wed, Oct 26	Thu, Oct 27	Fri, Oct 28	Sat, Oct 29	Sun, Oct 30	Mon, Oct 31
MEETING TO DETERMINE ENHANCED REQS.	-								
ASSIGNMENT & DEVELOPMENT INFRASTRUCTURE SETUP	-								

FILE DETECTION & AUTO-CATEGORIZATION **1.0**

File Detection	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Deliverable 3 Due
Auto-Categorization	1.1							
Auto-Categorization	1.2							

MULTI-FILE LOADING **2.0**

Multi-File Loading	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Deliverable 3 Due
Multi-File Loading	2.1							

ENHANCED ERROR CORRECTION **3.0**

Individual File Error Count (0..1)	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Deliverable 3 Due
Individual File Error Count (0..1)	3.1							

UI MODIFICATIONS **4.0**

Start transition of old UI to new UI	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Deliverable 3 Due
Start transition of old UI to new UI	4.1							
Redesign UI to accommodate 3.1	4.2							

SYSTEM TESTING **(2)**

Create Automated Unit Tests	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Deliverable 3 Due
Create Automated Unit Tests	2.1							
Unresolved Faded Tests	2.2							
Test Matrix (2)	2.3							

ENHANCED REQ DESCRIPTION WRITERUP **(3)**

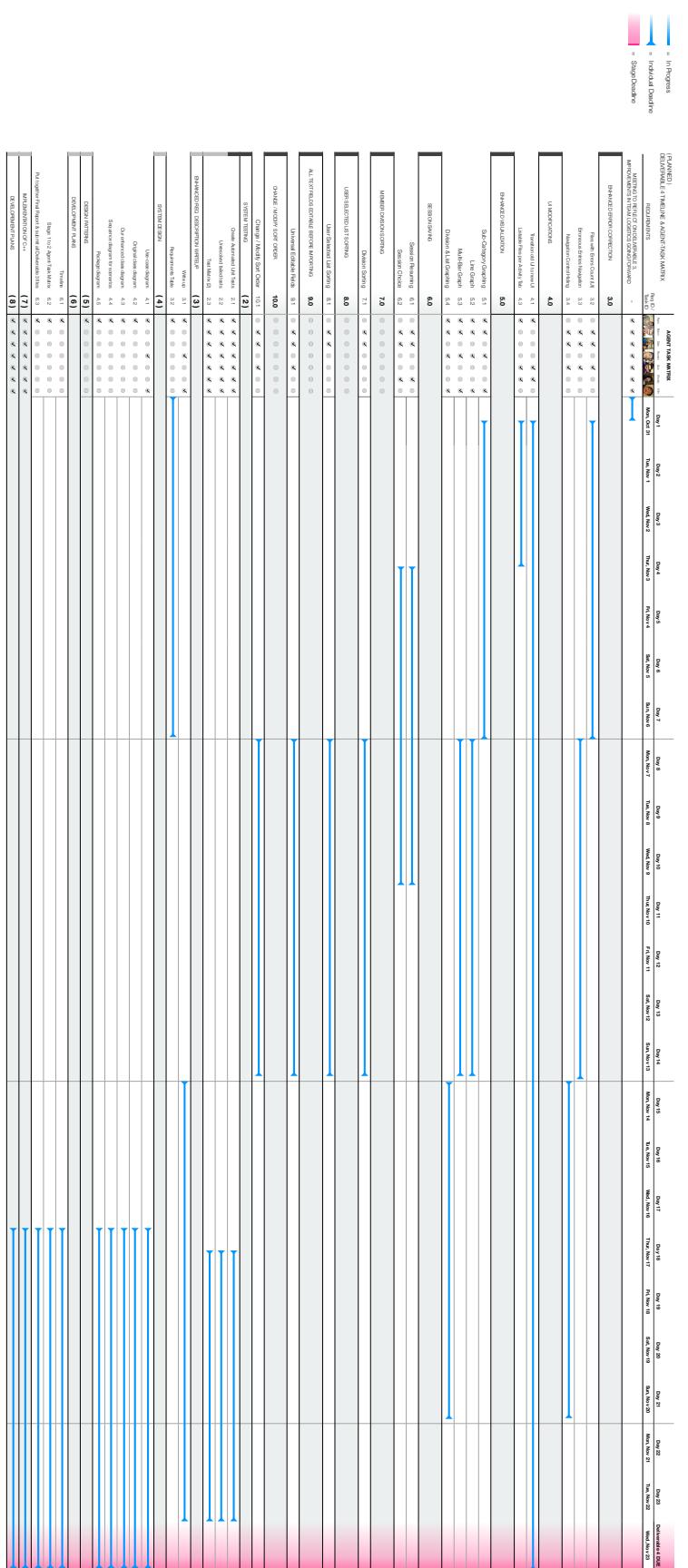
Write Up	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Deliverable 3 Due
Write Up	3.1							
Finish off Requirements Table	3.2							

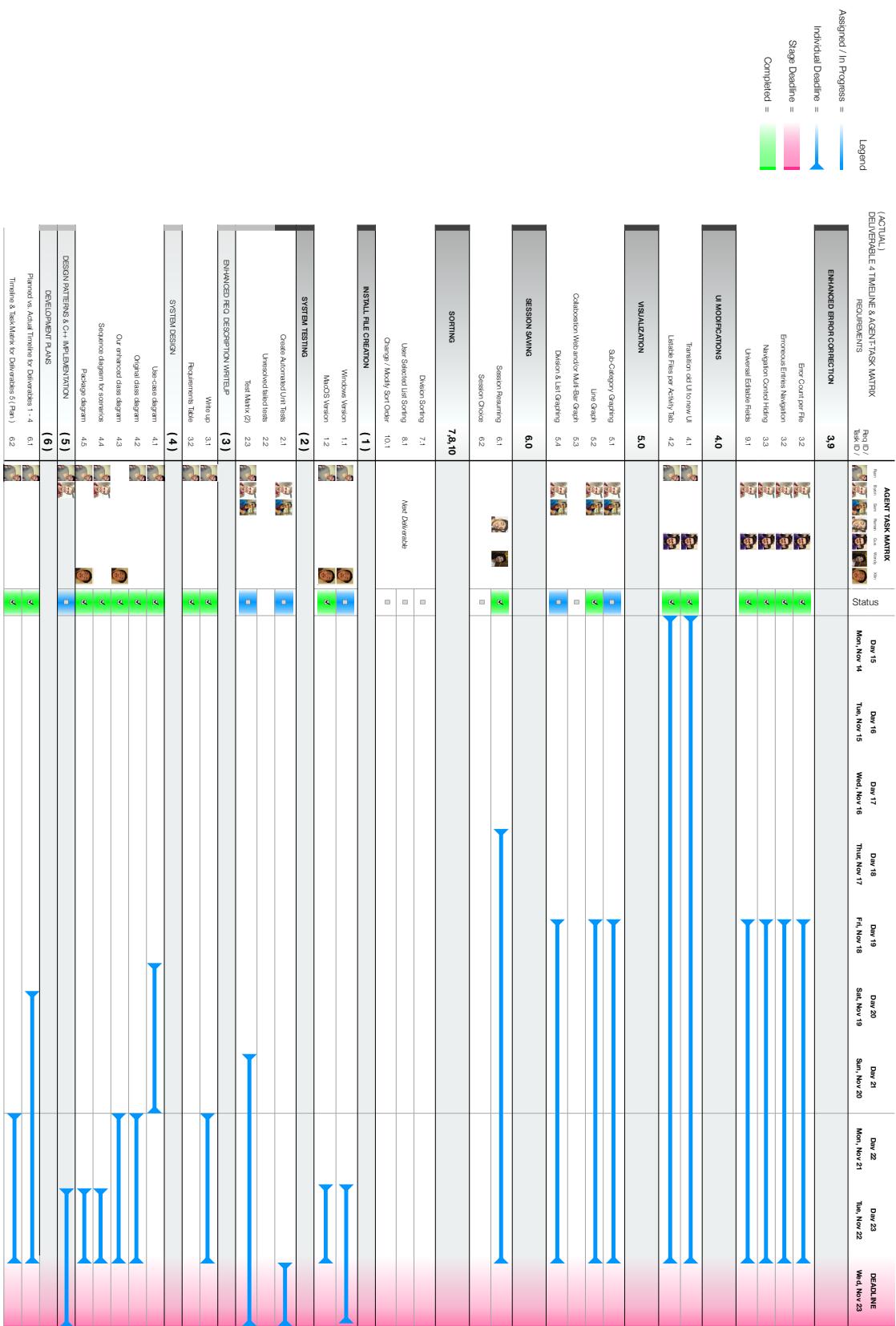
SYSTEM DESIGN **(4)**

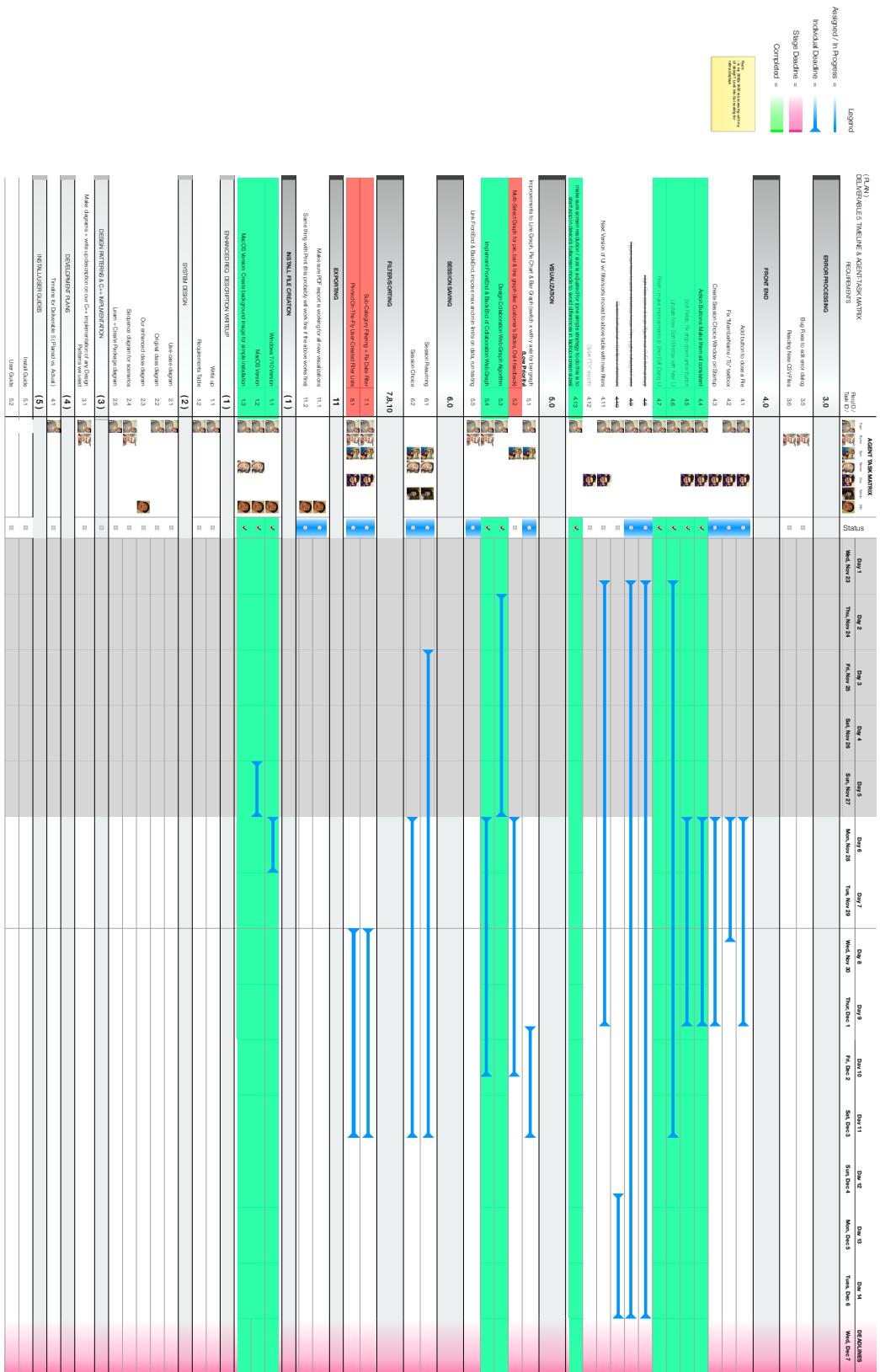
User Case Diagram	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Deliverable 3 Due
User Case Diagram	4.1							
Original class diagram	4.2							
Our enhanced class diagram	4.3							

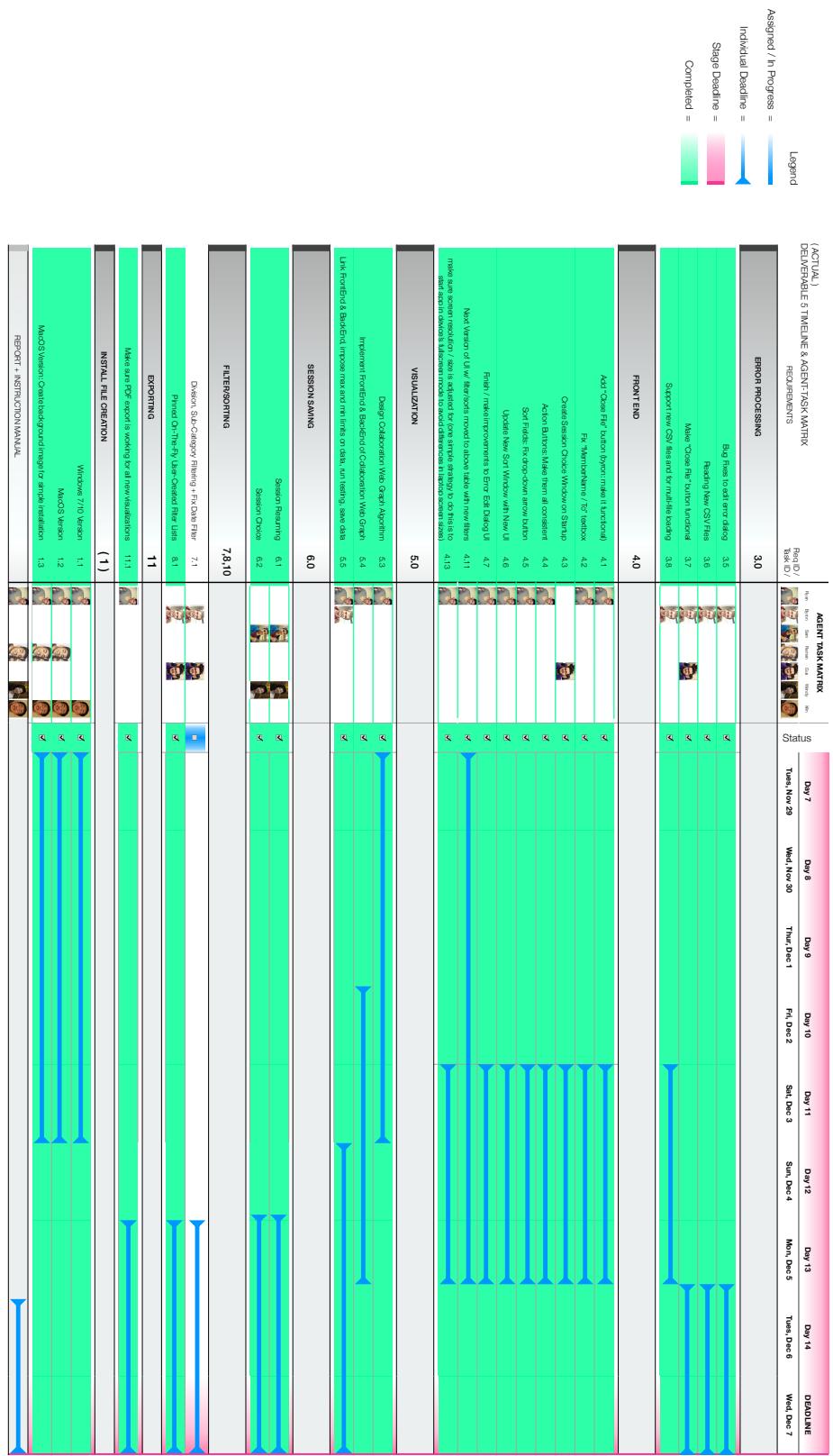
DEVELOPMENT PLANS **(5)**

Timeline	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Deliverable 3 Due
Stage 1 Agent-Task Matrix	5.1							
Stage 1 to 2 Agent-Task Matrix	5.2							
Put Together Final Report & Submit all Deliverable 3 Files	5.3							
Put Together Final Report & Submit all Deliverable 3 Files	5.4							









5 — Lessons Learnt

5.1 — Git

Git is a very useful tool and is easy to use. A few times we did get careless and have merge conflicts and issues when a few of us were working on the same branch. If we maintained a little more order it would be much more effective. One of your branches was also worked on much more than others and had effectively taken over as our Master branch, which is bad practice and was a hassle to revert. The effectiveness of Git still outweighed its tiny intricacies but require a good deal of order.

5.2 — Qt

Qt was new to all of us and came with a bit of a learning curve. Many of us were not even champions of C++ so we were taken aback with this great change to our environment. Qt was very effective for GUI and worked nicely with Github. Qt had a much more intuitive and large library which made it easier to create and make aesthetically pleasing aspects. Compared to Swing Library in Java it is overwhelming the amount of libraries offered and how easy they are to use.

5.3 — C++

C++ was new to many of us and was much less straight-forward than other programming languages we had used before. Thankfully we did have members who were strong and helped us throughout it was very encouraging having members who were very familiar and can solve little nuances we had that they have already dealt with in their careers. The manual memory-management was very unfamiliar, such a powerful tool required great focus to use well. Making sure files and their headers and dependencies were all in order was a key aspect as well. Creating useful functions to have modular code was a very important element as well since many functions were called over and over. Functions helped members use code that they hadn't worked on easily. Commenting throughout was very helpful in helping members work on your code without much instruction, this made for seamless transitions and saved time from members understanding code. Team Peach's old code was not very maintainable and to change the UI to the new UI required a complete overhaul of the code. Moving much of the code from the front-end to the back-end, to create a much cleaner code

5.4 — Team

Working in a team with members skills varying greatly is a huge challenge. Assigning tasks was hard since some members were stronger at coding C++ and needed to be reserved for the more important tasks. Some members also underestimated their abilities and had to be reassigned since some tasks had become too daunting, and switches had to be made on tasks. We had a team leader that kept everyone organized and assigned tasks to the best of their abilities but since you could not gauge members skill level it was hard to designate each member to a task that was reasonable.

A very valuable change in hindsight would be to start earlier, just to look at the task ahead. Conceptually some tasks seemed easier than they actually were. Since a lot of code had to be read and understand and not done from scratch. With this in mind a member could see his upcoming work and try to start and see if it is within his skill set and get help sooner rather than later.

5.5 — Value

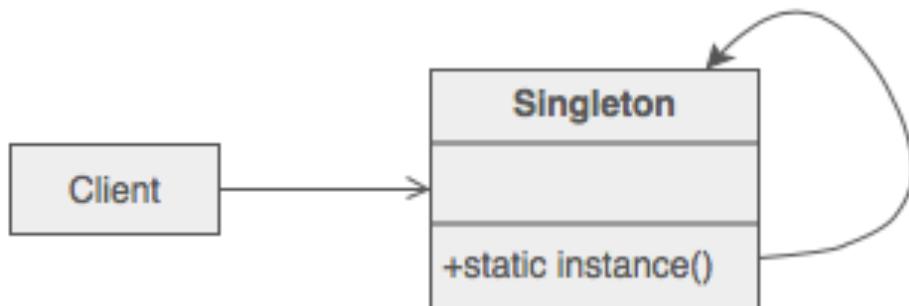
- Huge opportunity to work with a real customer
- Chance to make a program someone will actually use
 - Provides more encouragement to do the best work possible
- C++ is a great language to learn since it is such a popular language and it is tougher to use which helps in other aspects of programming

6 — Design Patterns

6.1 — Singleton Pattern

Most of our code was inherited from the TeamPeach Project, so much of our added code was built upon their design patterns.

A singleton design pattern is a design pattern where a class will only have one instance running, and will have a global access point. To make a class globally accessible, a

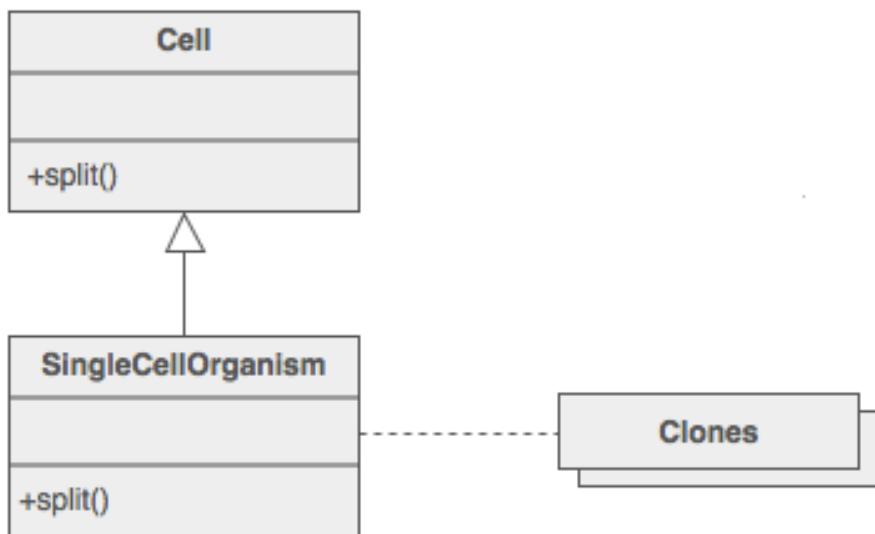


global variable is created. However, this does not solve the problem of multiple instances of a class. So, another solution would be for the class to keep track of whether an instance of it has been created or not.

The Singleton Pattern is used in the MainWindow class. Only one MainWindow will be running at any point in time.

6.2 — Prototype Pattern

The prototype design pattern is another pattern inherited from the TeamPeach source. In the prototype pattern, the objects created are modelled after a prototypical instance. If many classes have similar functions, it may be easier to have a prototype pattern, where an



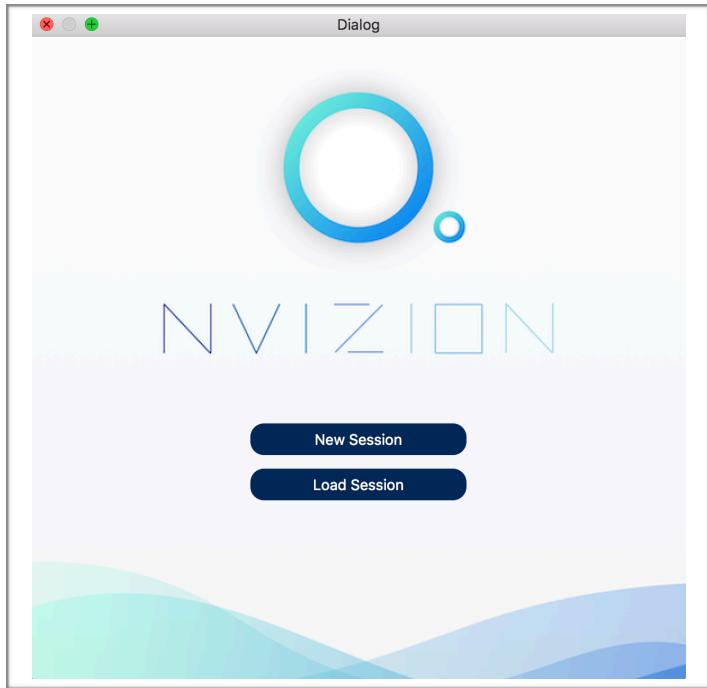
instance is cloned instead of created different instances. One of the main benefits of such a design is it bypasses the need to create a "new" instance, therefore reducing the load of an application.

The prototype pattern is used in TreeModel, which is instantiated when files are loaded. The MainWindow, PieChartWidget, and CustomSort instances are also cloned when a new state is required for each instance. This is much easier than creating a new instance from scratch.

7 — User Guide

7.1 — Starting Screen

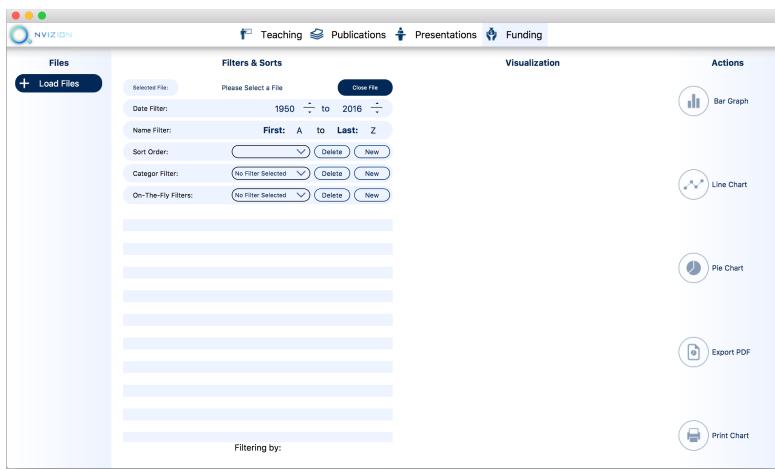
Starting Screen



This is what the Starting Screen looks like. Here you'll be able to select a new session, or select load session to continue working on a previous saved section. Clicking Load Session opens a File Browser in which case you can select a previously saved file which ends in **.dat format**.

CMD/CTRL + N → New Session
CMD/CTRL + L → Load Session

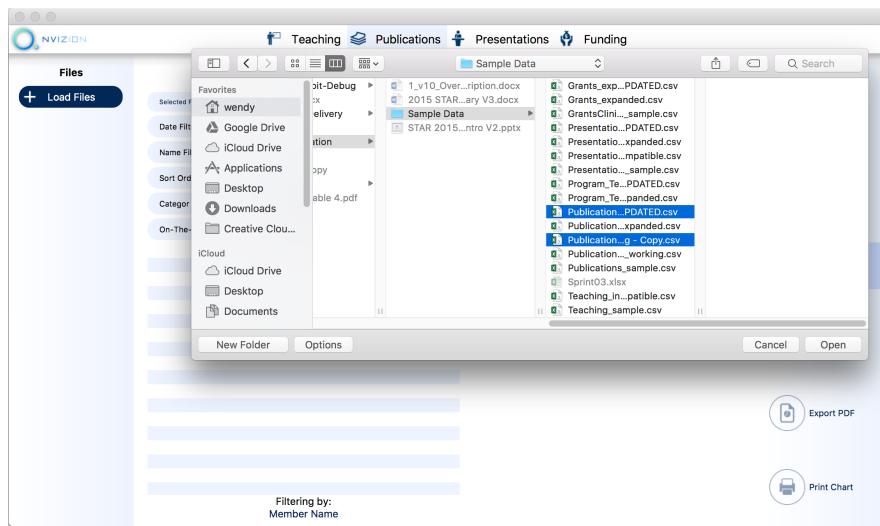
7.2 — New Session: The Dashboard



This is the New Session screen, where you'll see 4 tabs at the top indicating which type of file you're working with (Teaching, Publications, Presentations, Funding). You can Load Files (6.3 Load Files), select Filters & Sorts (6.4 Filters & Sorts), visualize your data (6.5 Visualize), as well as additional actions to change the data visualization (6.6 Actions).

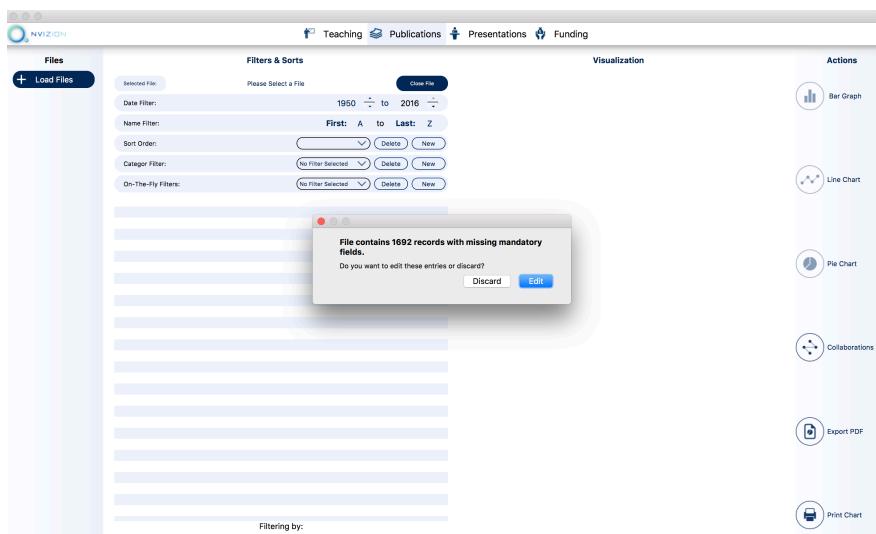
CMD/CTRL + L → Load Files

7.3 – Load Files



Publication: Load File

Clicking the **“+ Load Files”** button will bring up your file explorer, where you can choose which files to load. You are now able to select multiple files of any type you wish and click “Open”. All files will be opened and placed under each appropriate tab for you and listed in the left **“Files Sidebar”** for you to click on to view in the Dashboard.



Publication: Load File Errors

Occasionally, there may be missing fields in the CSV files that are being opened, in which case you can choose to edit the files, or discard rows with missing data and continue.

Publications_expanded_UPDATED.csv (Users/wendy/CS3307/CS3307Sirius/Project Information/Sample Data/Publications_expanded_UPDATED.csv)

ID	Member Name	Department	Division	Publication Status	ubmed Article ID	Type	Area	Status Date	Role	Peer Reviewed?	Published In (I)	Presentation?	Significant Publ	* Articles in ...	
														Principal Aut...	Collaborator
1	23084	Mootz, Dav...	Medicine	Allergy	Published		Published A...	Research	Principal Aut...	FALSE	FALSE	FALSE	FALSE	FALSE	
2	23097	Mootz, Dav...	Medicine	Allergy	Published		Published A...	Research	Principal Aut...	FALSE	FALSE	FALSE	FALSE	FALSE	
3	23115	Mootz, Dav...	Medicine	Allergy	Submitted		Published A...	Research	Collaborator	FALSE	TRUE	FALSE	FALSE	FALSE	
4	23116	Mootz, Dav...	Medicine	Allergy	Published		Published A...	Research	Collaborator	FALSE	TRUE	FALSE	FALSE	FALSE	
5	23118	Mootz, Dav...	Medicine	Allergy	Published		Published A...	Research	Collaborator	FALSE	FALSE	FALSE	FALSE	FALSE	
6	10108	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Nov-77	TRUE	TRUE	TRUE	FALSE	FALSE	
7	10109	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Nov-77	TRUE	TRUE	TRUE	FALSE	FALSE	
8	10110	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	1977	TRUE	TRUE	TRUE	FALSE	FALSE	
9	10111	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	1978	TRUE	TRUE	TRUE	FALSE	FALSE	
10	10113	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Mar-78	TRUE	TRUE	TRUE	FALSE	FALSE	
11	10114	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	1978	TRUE	TRUE	TRUE	FALSE	FALSE	
12	10115	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	1978	TRUE	TRUE	TRUE	FALSE	FALSE	
13	10116	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	1978	TRUE	TRUE	TRUE	FALSE	FALSE	
14	10118	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Oct-78	TRUE	TRUE	TRUE	FALSE	FALSE	
15	10120	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Oct-78	TRUE	TRUE	TRUE	FALSE	FALSE	
16	10121	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	1978	TRUE	TRUE	TRUE	FALSE	FALSE	
17	10122	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	1979	TRUE	TRUE	TRUE	FALSE	FALSE	
18	10123	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	1979	TRUE	TRUE	TRUE	FALSE	FALSE	
19	10125	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Feb-79	TRUE	TRUE	TRUE	FALSE	FALSE	
20	10126	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Feb-79	TRUE	TRUE	TRUE	FALSE	FALSE	
21	10129	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Jun-79	TRUE	TRUE	TRUE	FALSE	FALSE	
22	10131	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Oct-79	TRUE	TRUE	TRUE	FALSE	FALSE	
23	10132	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Jun-80	TRUE	TRUE	TRUE	FALSE	FALSE	
24	10133	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	Jun-80	TRUE	TRUE	TRUE	FALSE	FALSE	
25	10134	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	1980	TRUE	TRUE	TRUE	FALSE	FALSE	
26	10135	Kostuk, Will...	Medicine	Cardiology	Published		Published A...	Research	1980-02-25	TRUE	TRUE	TRUE	FALSE	FALSE	

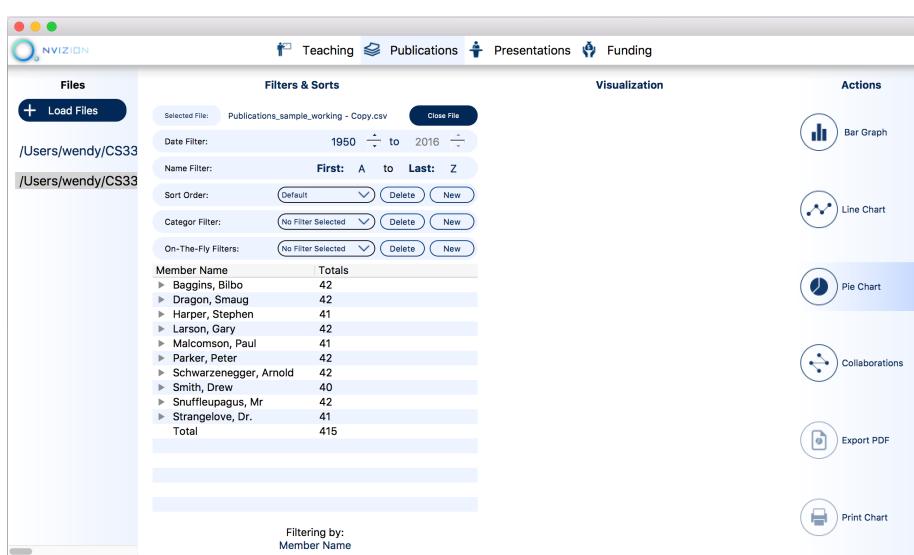
Previous File Next File File Contains 1651 Errors Cancel Save

Publications: Edit Load Files

If you clicked “Edit”, you will see this full screen error-editing view, where the missing fields are highlighted in pink. You can navigate through the errors and files, as well as cancel or save. Current file being viewed is displayed at the top of the window along with its file path.

ESC → Cancel

CMD/CTR + S → Save

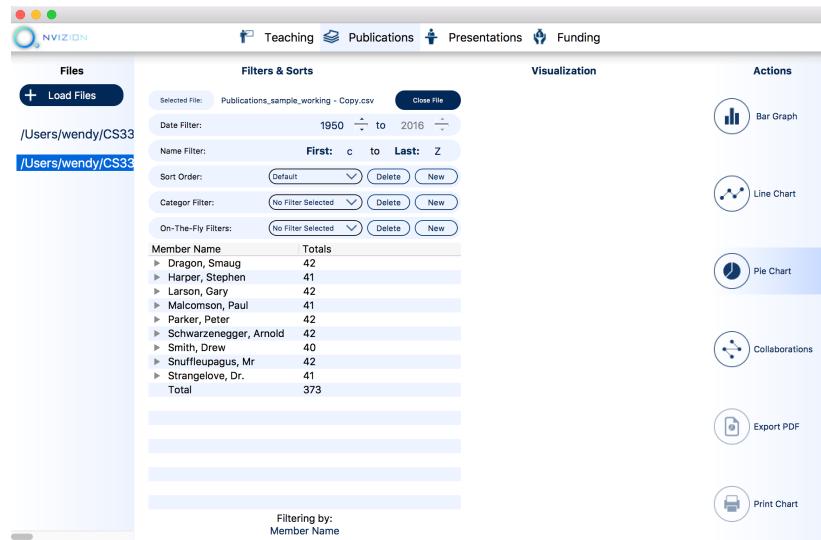


Example: Loading Two Publication Files

After loading the two files, you can choose which one to work with. After selecting the second file in the Files column, you can see the data populated under the Filters & Sorts. The Filenname and Filepath is displayed above the Filters & Sorts in between the **Close File button** and “Selected File” label. Close File button unloads the dashboard’s table and removes the file from the current session.

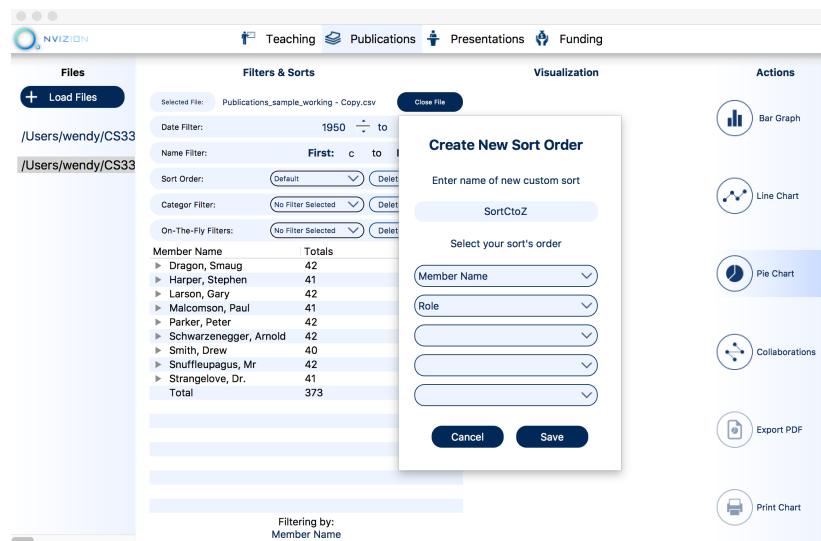
7.4 – Filters & Sorts

There are various filters that you can work with, ranging from name filter, date filter, sort order, and more. The following are some examples of the Name and Sort filters.



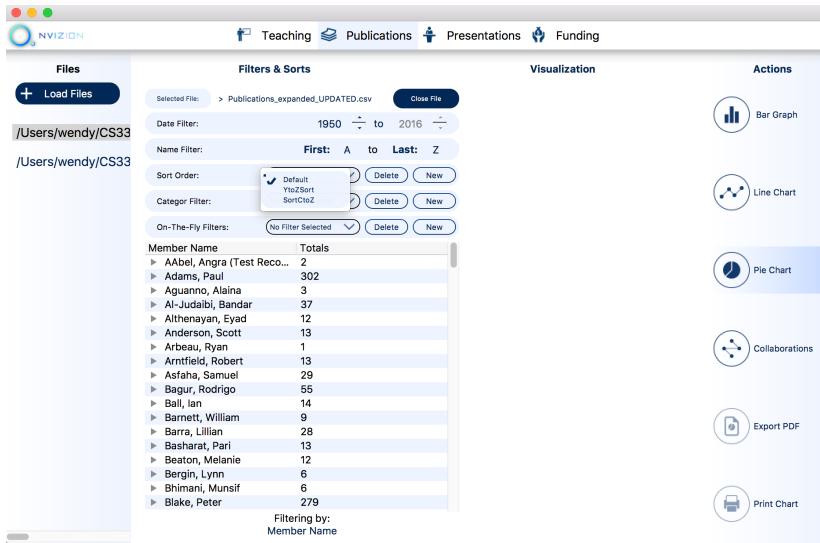
Filter & Sort: Name Filter

By typing C into “First.” in the Name Filter, you can filter all the members by last name from C to Z.



Filter & Sort: Save Sort Order

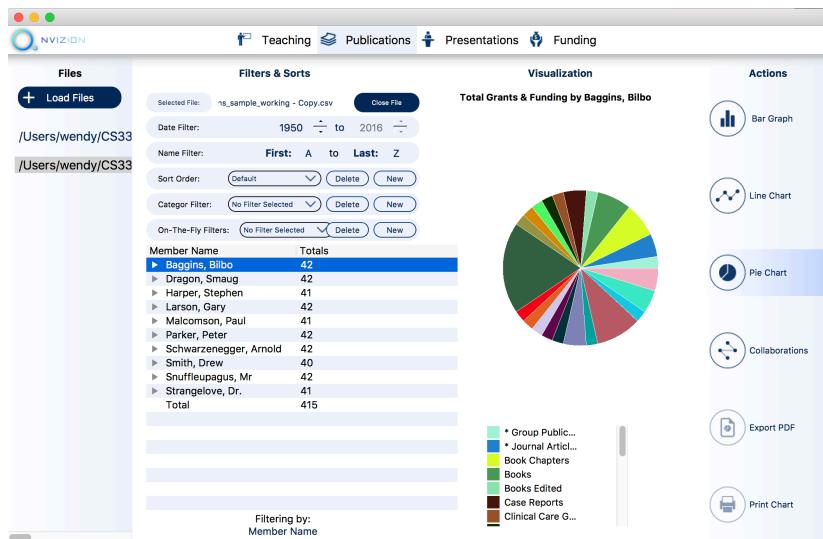
To save a sort order, you can click New under Sort Order. Here, we've named the new sort order “SortCtoZ”, and sorted by Member Name and Role.



Filter & Sort: See Saved Sort Orders

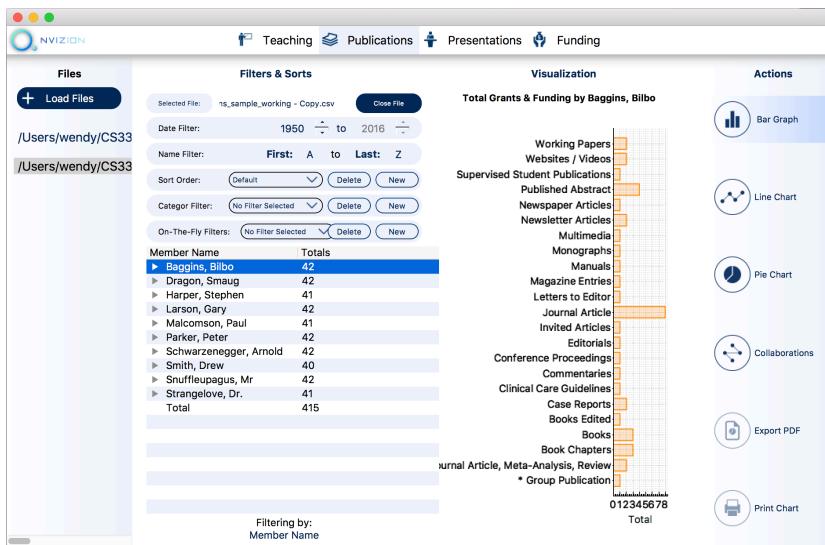
If you select the Sort Order again, you can see the previously saved sort orders and apply selected sorts.

7.5 –Visualizations

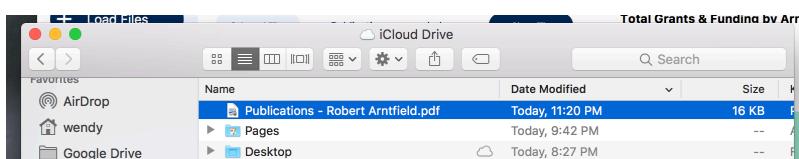
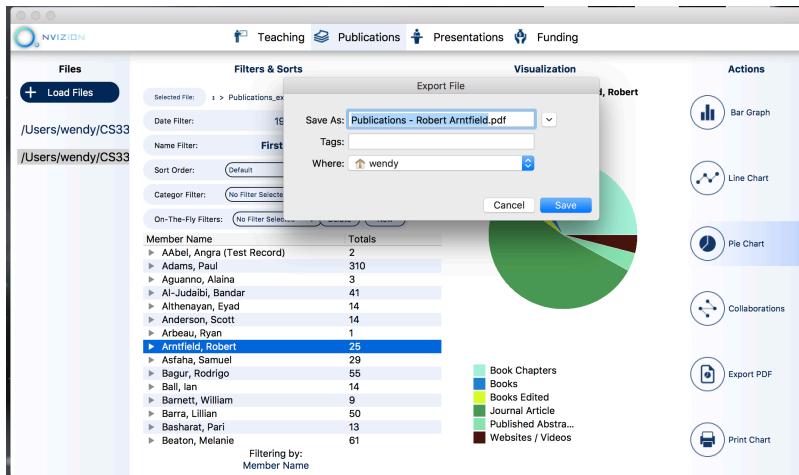


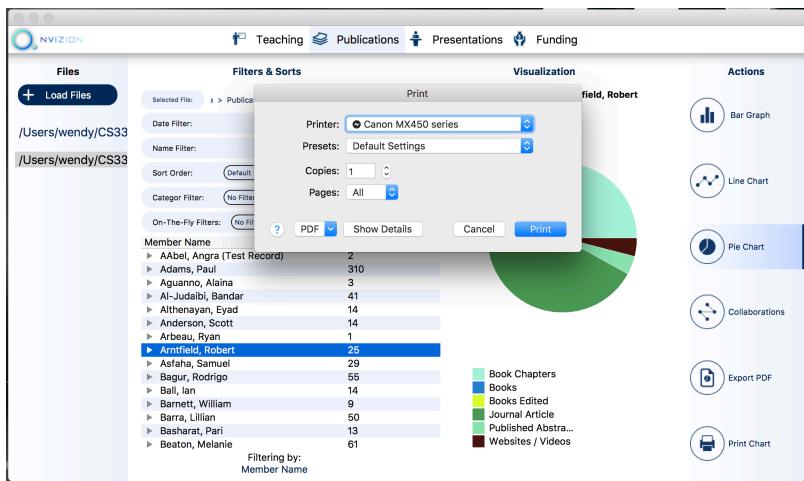
Visualization: Pie Chart

By clicking Pie Chart while selecting a member name, we can see the details for that member. The legend below the pie chart becomes scrollable when it is too large to display in the window as shown.



7.6 – Export PDF / Print





Print Visualization

On clicking "Print Chart", you can also select a printer to print the visualization to.

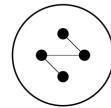
CMD/CTR + P → Print

7.7 – Session Saving

To save a session, simply attempt to close the window by clicking the "x" button in your respective OS-dependent close button. You will be prompted to save the session or discard your work upon button press.



Collaboration View user guide follows on the next page.



Collaboration View

User clicks on Collaboration View () in Publications Tab

8.1 — Definition

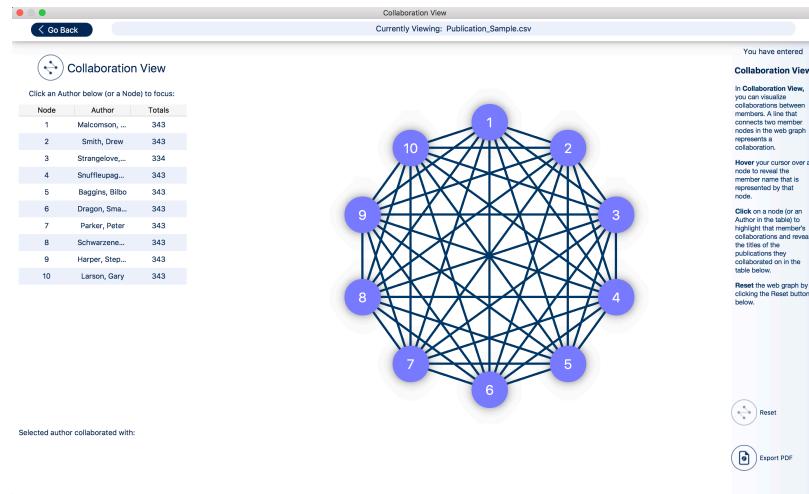
In Collaboration View, you can visualize collaborations between members in a Publication CSV file. Here, we demonstrate this feature by loading 2 files: 1) Publication_Sample.csv and 2) Publication_Expanded_UPDATED

Member Name	Totals
Baggins, Bilbo	42
Dragon, Smaug	42
Harper, Stephen	41
Larson, Gary	42
Malcomson, Paul	41
Parker, Peter	42
Schwarzenegger, Arnold	42
Smith, Drew	40
Snuffleupagus, Mr.	42
Strangelove, Dr.	41
Total	415

Here, we've loaded Publication_Sample.CSV. We have navigated to the Publications Tab. Note the "Collaborations" button located in the right "Actions Side-bar" which is unique only to the Publications Tab.

Click the "Collaborations" button.

A new maximized window pop ups with a web graph visualization as shown below:



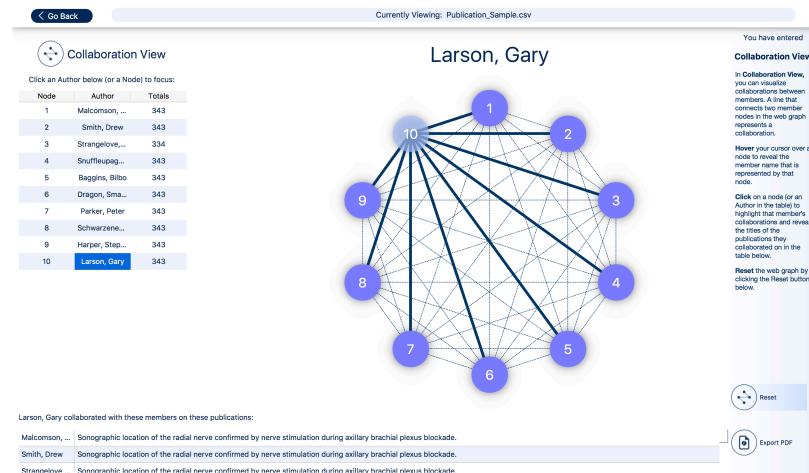
Each node in the web graph represents an author (ie. a member) and a line between two nodes represents a collaboration between those two members. **The main criteria we used to determine if a collaboration has occurred is:** if the title of a publication is the same between two or more members, then it was assumed those two or more members collaborated with one another.

8.2 — Possible User Actions within View

You can simply click on a name in the table on the left (or the node number in the Node column) and NVIZION will automatically:

- ▶ highlight the corresponding node (focus on selected Author's node)
- ▶ bolden only it's connecting lines (focus on selected Author's collaboration lines)
- ▶ fade away all other connecting lines
- ▶ Set the title of the Collaboration View's web graph to the selected Author's name.
- ▶ Reveal the name of all authors the selected author has collaborated with along with the title of all publications they have collaborated on, in the table positioned at the bottom below the web graph "Selected Author Collaborated With: "

The following screenshot shows the above in action where Larson Gary was selected:



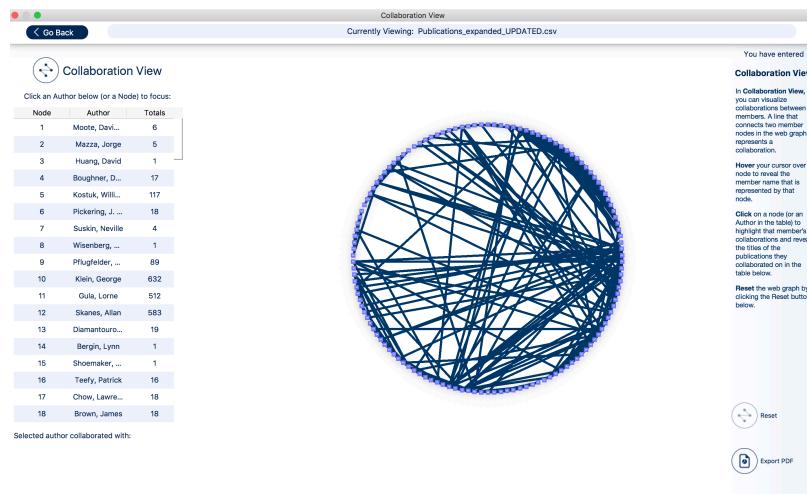
8.3 — Additional actions the user is able to take in Collaboration View:

- ▶ You can click the actual nodes themselves to focus on one node and its connecting lines
- ▶ You can hover your cursor over each node to reveal the name of the author each node represents.
- ▶ You can click the “Reset” button in the right “Actions Bar” to “unselect” a node and reset the node and line focus (reset the web graph).
- ▶ In the table on the left, you can sort each columns by clicking each header. The Node and Totals column will be sorted from smallest to largest and Author will be sorted alphabetically.
- ▶ You can export two things from this view: the web graph with its title (if a node / author has been selected) and the left legend table which will be rendered on the second page in the PDF document.
 - ▶ Note: the render of the web graph *might* be very small in the PDF file that is exported — but because PDF is a vector-based format, you can zoom in as much as you want into where the web graph has been placed on the page while retaining perfect render quality of the web graph. In hindsight, a fully-fledged PDF Exporting open-source C++ library should have been used but because the customer did not specify that they needed the PDF exporting functionality to be improved upon, we re-used the previous team's (Team Peach's) manual PDF exporting functionality.

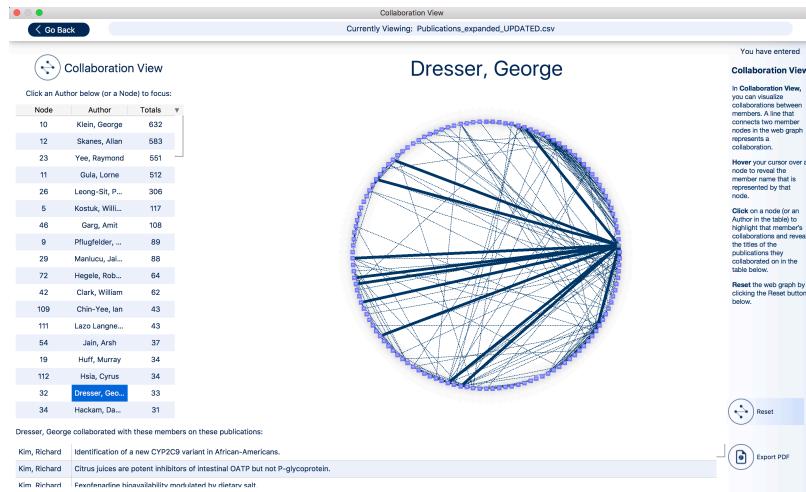
8.4 — Warning

When loading very large files, the computations required to determine collaborations can take awhile. On a Core i7 with 16 GB RAM, a solid-state drive running MacOS Sierra, it took us ~2 minutes to render the web graph for **“Publications_expanded_UPDATED.csv”**. However, once a large file has been loaded and computed once, it **does not** need to re-compute the collaborations every time the Collaborations button is pressed. Performance may vary on different machines and different file sizes. The “sample” files loaded instantly in our tests.

8.5 — Demo of Large File Visualization: A screenshot of Publications_expanded_UPDATED.csv being loaded and rendered:



The following below has “Dresser, George” selected. Note the **“Totals” column** has been sorted to find out who has collaborated the most.



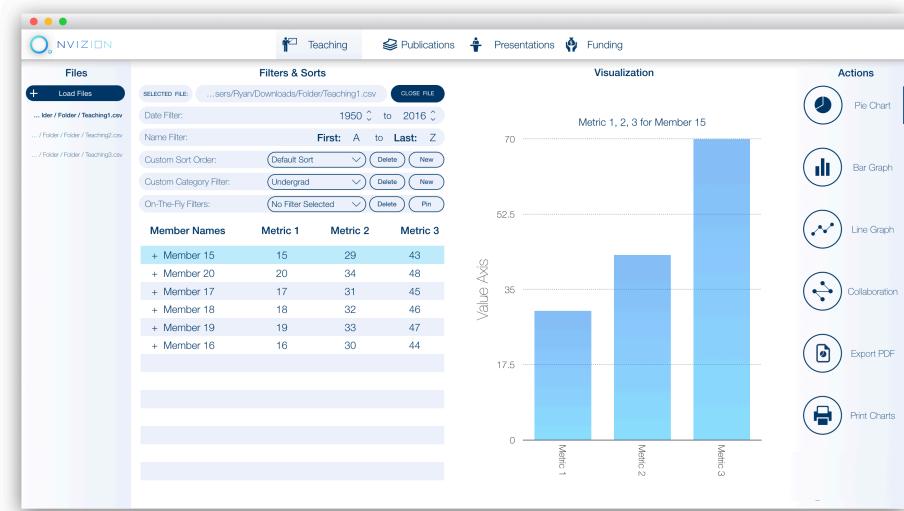
Note that tips and instructions on what a user can do in the Collaboration View will always be present in the right Actions Side-Bar.

Explanations of the algorithms that generate this visualization, (which were created from scratch), are given in the Appendix.

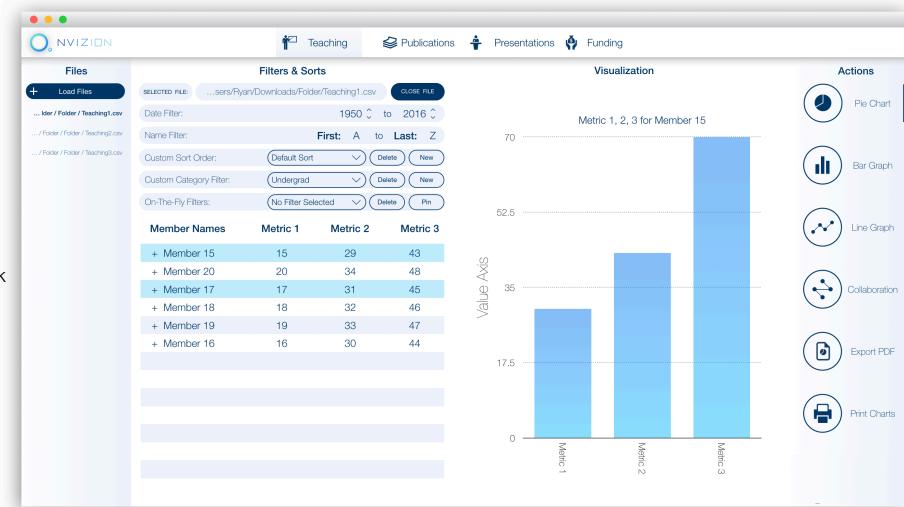
On-The-Fly Filters (User-Selected Lists Filtering) Feature is walked-through on the next set of pages.

**User Selected List Filtering
(On The Fly Filters)**

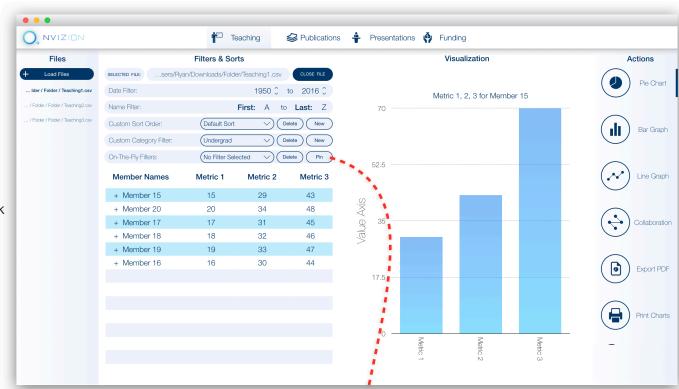
Only members that have taught undergrad now show up:
(note that we have a default sort order here)



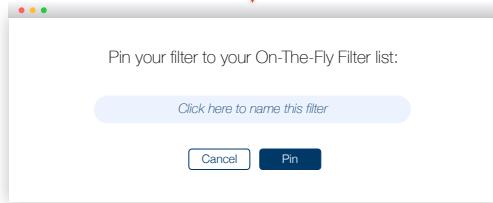
User has selected 2 rows by holding down Ctrl/Cmd + left click



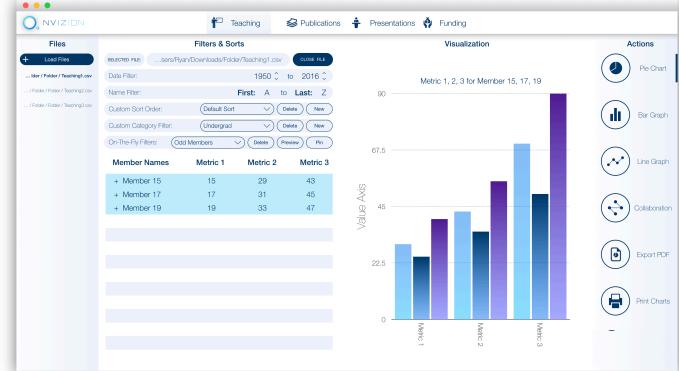
User has selected 3 rows by holding down Ctrl/Cmd + left click



User can pin this "On the fly" filter to their saved filtered lists so they may instantly re-apply this user-created filter at another time.



Pinned filter is applied and only filtered items are now visible. Only these items will be inputted into the graph.



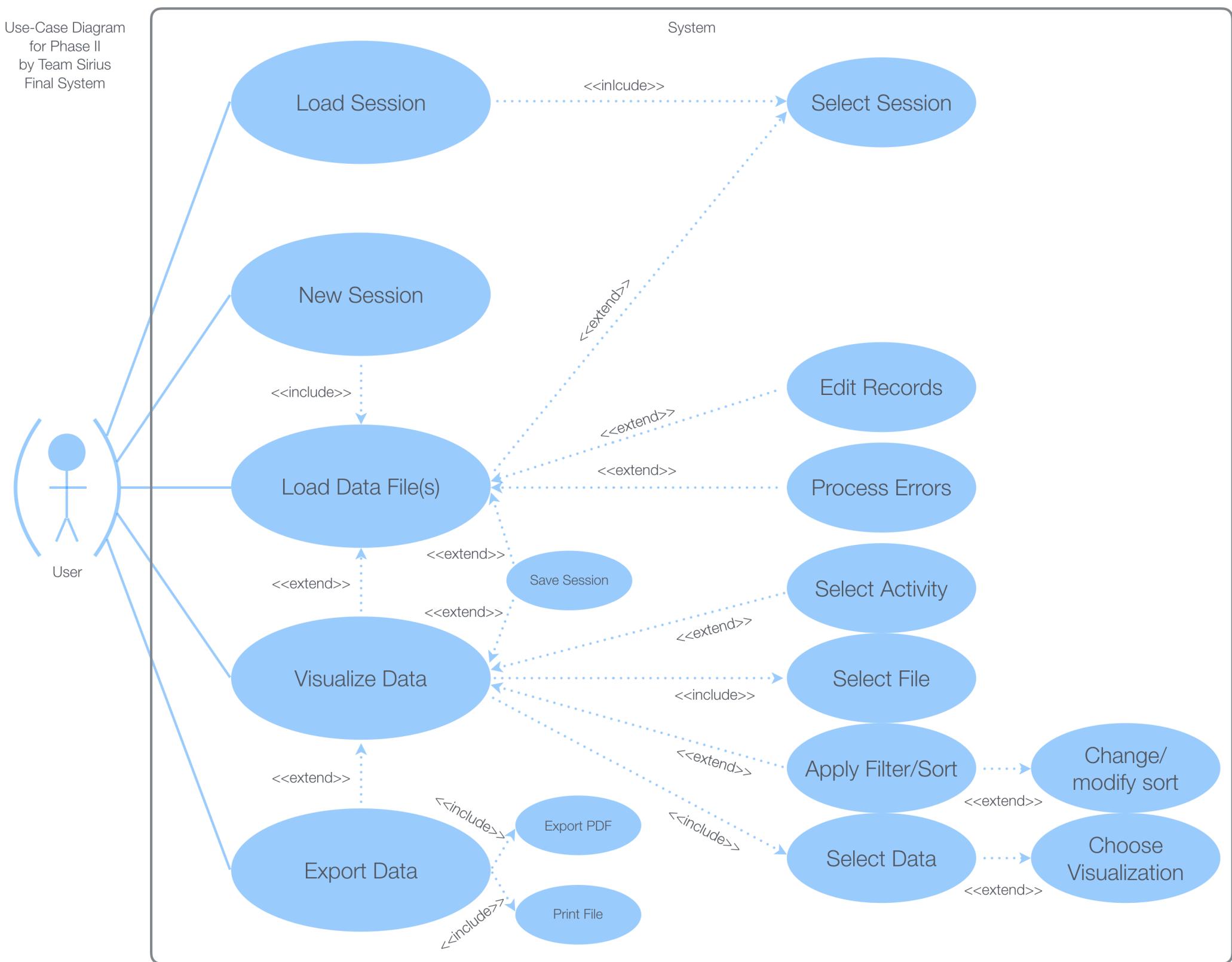
Appendix

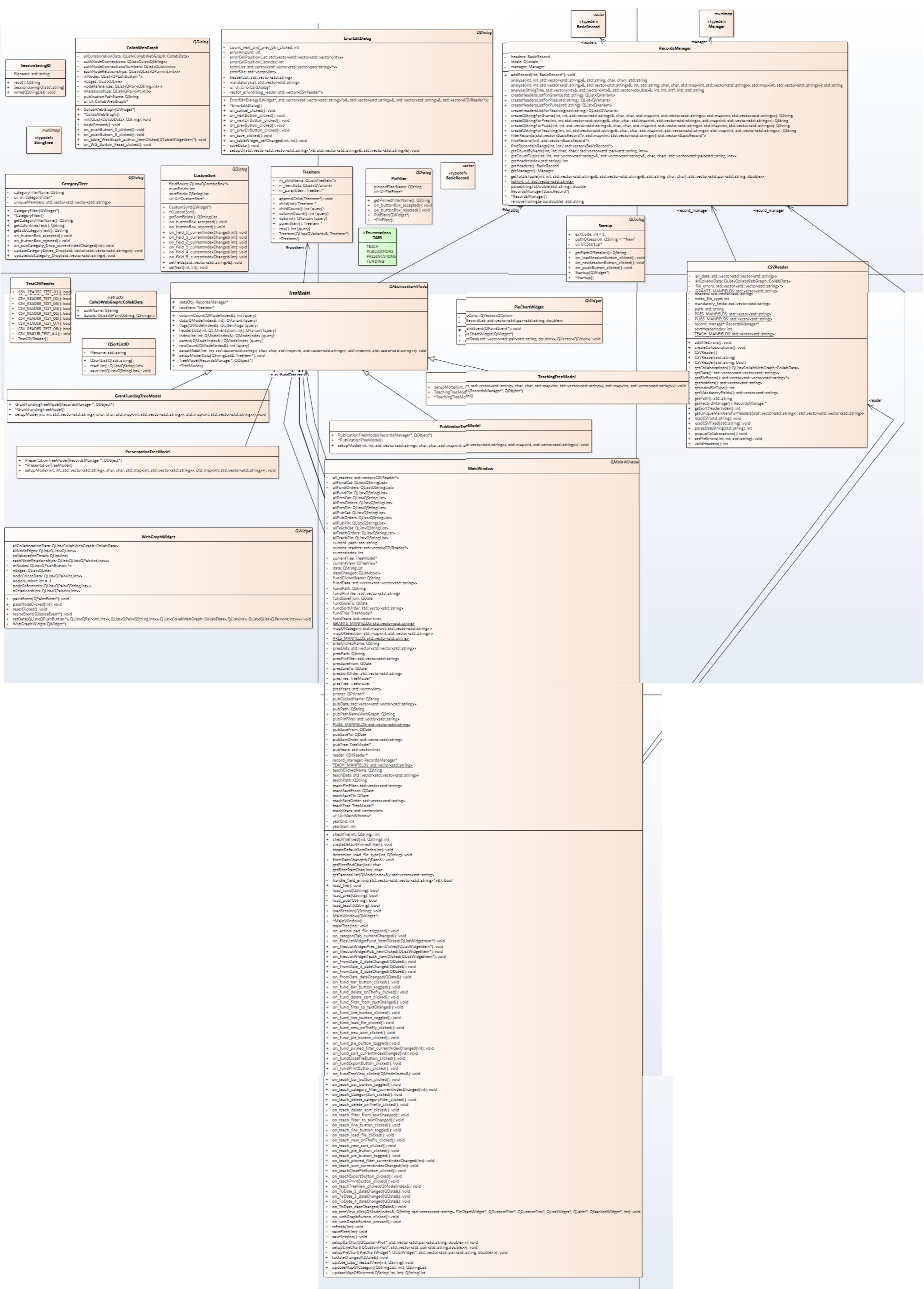
Software Requirements Specification Sheet
Manual Test Matrix
Use Case Diagram
Class Diagram, Package Diagram, &
Sequence Diagram
Web Graph Algorithm Description &
Explanation
Timelines & Agent-Task Matrices

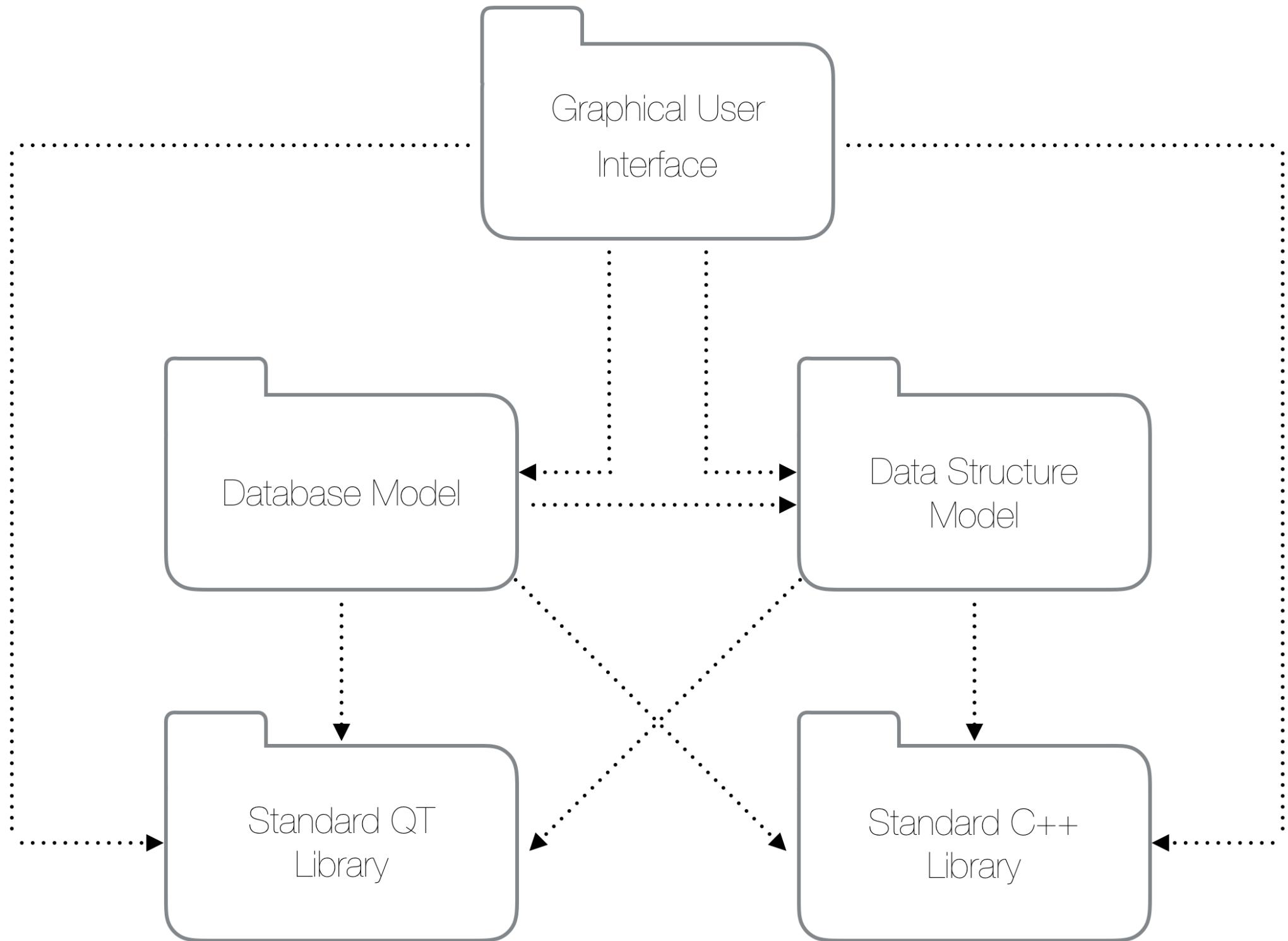
Completed = ✓ In Progress = ■

ALL ENHANCED REQUIREMENTS		REQ ID	STATUS	REQ DESCRIPTION	SUPPORTING EXAMPLE	ORIGIN
FILE DETECTION & AUTO-PLACEMENT		1.0	✓			Team Sirius
Stretch Goals	File Detection	1.1	✓	Software detects the type of activity of file out of the four possible activity types when loading a file		Team Sirius
	Automatic Placement	1.2	✓	When file is detected and loaded, the software automatically places it under the right activity tab		Team Sirius
	MULTI-FILE LOADING		2.0	✓		
	Multi-File Loading	2.1	✓	User is able to select multiple CSV files in the File Explorer and successfully load files	Refer to Updated UI Mockup	Team Sirius
	ENHANCED ERROR CORRECTION		3.0	✓		
Mandatory Goals		3.1	✓	When the CSV file is loaded, the error dialog box should display total number of errors	Refer to Updated UI Mockup	Customer (Required Goal)
Mandatory Goals	Error-Filled Multi-File Navigation	3.2	✓	Next/Prev button navigates to files with errors	Refer to Updated UI Mockup	Team Sirius
	Erroneous Entries Navigation	3.3	✓	"Find Next/Prev" button which jumps on to the next/prev error must be present	Refer to Updated UI Mockup	Customer (Required Goal)
	Navigation Control Hiding	3.4	✓	After all the errors are fixed, "Find Next/Prev" should be disabled.	Refer to Updated UI Mockup	Customer (Required Goal)
	UI MODIFICATIONS		4.0	✓		
Stretch Goals		4.1	✓	Self-Explanatory	Refer to Updated UI Mockup	Team Sirius
Mandatory Goals	Redesign UI to accommodate 3.0	4.2	✓	Add in controls and text boxes	Refer to Updated UI Mockup	Team Sirius
	Listable Files per Activity Tab	4.3	✓	Left Side-bar per Activity Tab listing loaded files. Click file to load to dashboard	Refer to Updated UI Mockup	Team Sirius
	ENHANCED VISUALIZATION		5.0	■		
Mandatory Goals		5.1	■	User is able to graph sub-category data		Customer (Required Goal)
Mandatory Goals	Line Graph	5.2	✓	User has the option to visualize data with a Line Graph	Refer to Updated UI Mockup	Customer (Required Goal)
	Collaboration Web Graph for Publications	5.4	✓	User has the option to visualize author collaboration data (from publications) via a node-web visualization	Refer to Updated UI Mockup	Team Sirius
	SESSION SAVING / RESUMING		6.0	■		
Stretch Goals		6.1	✓	User can resume the previous session and all previous loaded files and filters will be preserved	Refer to Updated UI Mockup	Customer (Required Goal)
Mandatory Goals		6.2	✓	Upon startup of application, user has choice of starting a new session or loading previous session	Refer to Updated UI Mockup	Customer (Required Goal)
Stretch Goals		MEMBER DIVISION SORTING		7.0	■	Customer (Required Goal)
Mandatory Goals		7.1	■	User has the option to sort by new updated CSV categories		Customer (Required Goal)
Stretch Goals		USER SELECTED LIST SORTING		8.0	■	Customer (Required Goal)
Mandatory Goals		8.1	✓	User has the option to sort by a user generated filter list		Customer (Required Goal)
Stretch Goals		ALL TEXT FIELDS EDITABLE BEFORE IMPORTING		9.0	✓	Customer (Stretch Goal)
Mandatory Goals		9.1	✓	User has the option to change any fields before importing them		Customer (Stretch Goal)
Stretch Goals		MAC OS DRAG & DROP INSTALL FILE		10.0	✓	Customer (Stretch Goal)

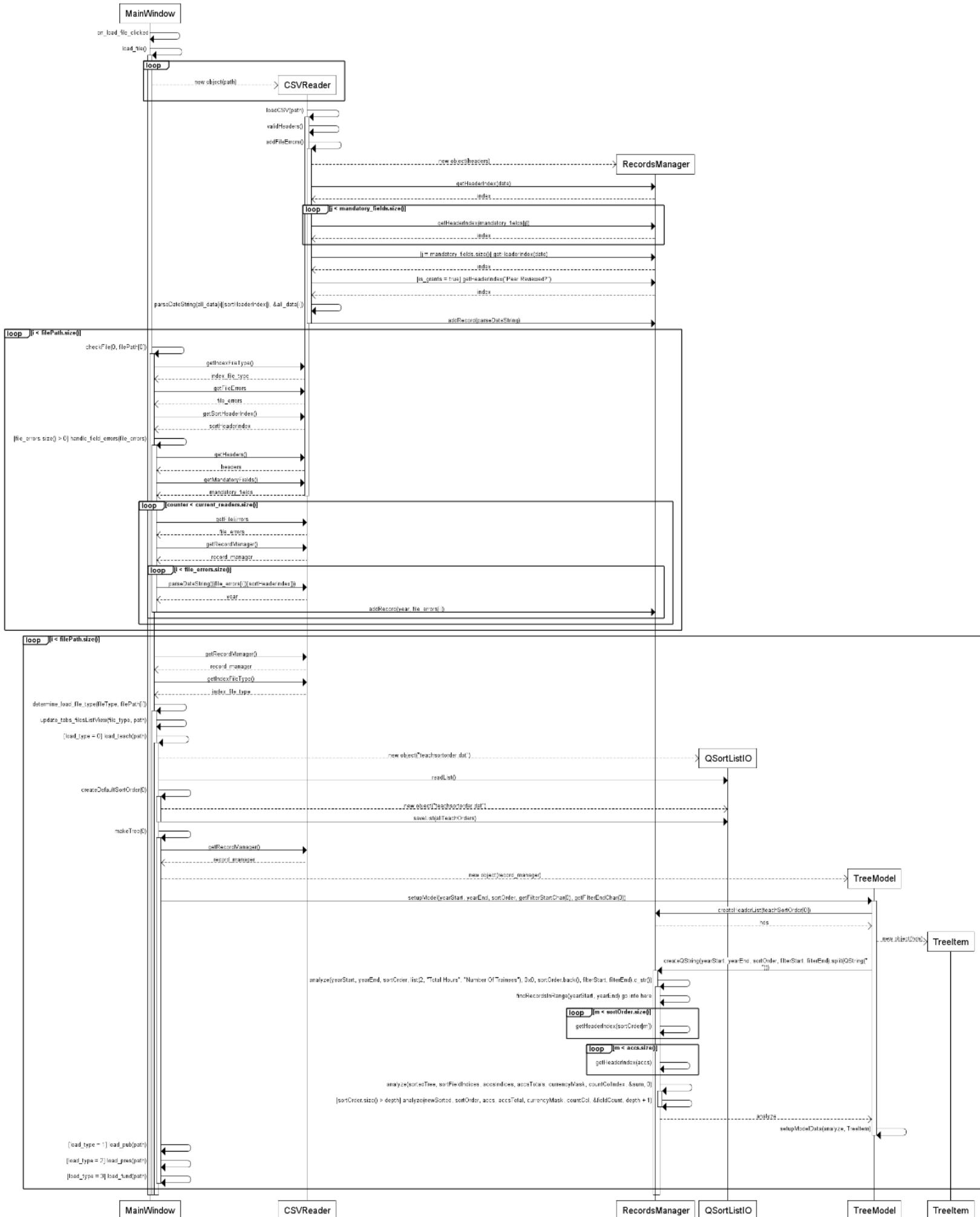
Manual Test Matrix								
Requirements Tested	Test Case ID	Test Case	Assumptions	Steps to Administer Test	Test Data	Expected Results	Actual Results	Pass or Fail
CHECK FILE LOADING FUNCTIONALITY	1.0							P
	1.1	User attempts to load a compatible CSV file	CSV file is compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects a CSV file and clicks the Open button.	[Teaching_compat_errors.csv]	The system loads the records or the system prompts user to edit or discard missing records	System loaded file / prompted user to edit or discard missing records	P
	1.2	User attempts to load an incompatible file	CSV file is incompatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects a CSV file and clicks the Open button.	[Teaching_incompatible.csv]	The system displays an error message.	The system displays an error message.	P
	1.3	User loads multiple CSV files	CSV file is compatible and error free	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects multiple CSV file that contains invalid records and clicks the Open button.	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system loads the multiple files and displays the correct activity-related files under the correct activity tab in the left sidebar	The system loaded the multiple files and displayed the correct activity-related files under the correct activity tab in the left side-bar	P
CHECK ERROR PROCESSING FUNCTIONALITY	2.0							P
	2.1	User loads a CSV file that contains invalid records	CSV file is compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects multiple CSV file that contains invalid records and clicks the Open button.	[CSV file containing invalid records]	The system displays message showing number of invalid records and prompts user to edit or discard them. After selecting edit, the number of errors are displayed along with the editable fields and error count.	System displayed message showing number of invalid records and prompted user to edit or discard them. After selecting edit, the number of errors were correctly displayed.	P
	2.2	User loads a CSV file that contains all valid records	CSV file is compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects multiple CSV file that contains invalid records and clicks the Open button.	[CSV file containing all valid records]	The system loads the records without prompting user to edit invalid records.	System loaded the records with no prompt asking the user to edit the data	P
	2.3	User edits / includes newly modified records in the data to be loaded after errors are detected	CSV file is compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects multiple CSV file that contains invalid records and clicks the Open button. 4. The user fills in all missing entries and clicks the Save button.	[CSV file containing invalid records]	The system hides the controls and the error count displays 0. The system then loads the records with newly modified data.	The system hid the controls and the error count displayed 0. The system then loaded the records with newly modified data.	P
	2.4	User cancels error processing or discards changes to detected invalid records	CSV file is compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects a CSV file that contains invalid records and clicks the Open button. 4. The user clicks Discard or Cancel button	[CSV file containing invalid records]	System discards records with missing mandatory entries from the data to be loaded	System loaded the records with omitted records	P
	2.5	User loads multiple CSV files with errors	CSV files are compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects multiple CSV files that contains invalid records and clicks the Open button.	[CSV file containing valid records]	The system displays message showing number of invalid records and prompts user to edit or discard them for each file in sequential order. After selecting edit, the number of errors are displayed along with the editable fields for each file.	The system displayed a message showing number of invalid records. Prompted user to edit or discard them for each file in a sequential order. After selecting edit, the number of errors were displayed along with the editable fields for each file.	P
	2.6	User navigates erroneous entries	CSV files are compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects multiple CSV files that contains invalid records and clicks the Open button. 4. User clicks the Next/Previous Error buttons in the Error Processing window.	[CSV file containing invalid records]	The system navigates to the next or previous erroneous field	The system navigated to the next or previous erroneous field	P
	2.7	User navigates erroneous entries after a field is fixed	CSV files are compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects multiple CSV files that contains invalid records and clicks the Open button. 4. User fixes the first erroneous field 5. User clicks the Next and Previous Error button in the Error Processing window in that order	[CSV file containing invalid records]	The system saves the fixed / new value, navigates to the next error field, but does not navigate back to the previously fixed field when user click on the Previous Error button.	The system saved the fixed / new value, navigated to the next error field, but did not navigate back to the previously fixed field when user clicks on the Previous Error button.	P
	2.8	Navigation controls hide after user fixes all errors	CSV files are compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects multiple CSV files that contains invalid records and clicks the Open button. 4. User fixes all erroneous fields present in file.	[CSV file containing all valid records]	The system hides the navigation controls once all errors are fixed for that file	The system hid the navigation controls once all errors were fixed for that file	P
CHECK DASHBOARD NAVIGATION FUNCTIONALITY	3.0							P
	3.1	User attempts to view Teaching data in the dashboard	CSV file is compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects a CSV file and clicks the Open button.	[Teaching_compat_errors.csv]	The system displays the updated dashboard summary view for Teaching Activity.	System displayed the updated dashboard summary view for Teaching Activity.	P
	3.2	User attempts to view Publication data in the dashboard	CSV file is compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects a CSV file and clicks the Open button.	[Publications_compat_errors.csv]	The system displays the updated dashboard summary view for Publications Activity.	System displayed the updated dashboard summary view for Teaching Activity.	P
	3.3	User attempts to view Presentation data in the dashboard	CSV file is compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects a CSV file and clicks the Open button.	[Presentations_compat_errors.csv]	The system displays the updated dashboard summary view for Presentations Activity.	System displayed the updated dashboard summary view for Teaching Activity.	P
	3.4	User attempts to view Grant & Funding data in the dashboard	CSV file is compatible	1. The user clicks the Load button. 2. The system displays a File Explorer screen. 3. The user selects a CSV file and clicks the Open button.	[GrantsClinicalFunding_compat_errors.csv]	The system displays the updated dashboard summary view for Grants + Funding Activity.	System displayed the updated dashboard summary view for Teaching Activity.	P
	3.5	User expands/collapses sections of the dashboard summary view.	User has loaded file	1. The user clicks on an expand/collapse arrow	[CSV file containing all valid records]	The system displays the expanded/collapsed elements of the dashboard summary view.	System displayed the expanded/collapsed elements of the dashboard summary view.	P
	3.6	User navigates to different Activity tabs	User has loaded all files	1. The user clicks on one of four activity tabs	[CSV file containing all valid records]	The system displays the updated dashboard summary view for the selected Activity.	System displayed the updated dashboard summary view for the selected Activity.	P
CHECK FILTER FUNCTIONALITY	4.0							P
	4.1	User filters start and end dates with desired date range	CSV file is already loaded	1. The user clicks on start date / end date text field and enters in desired values 2. The user hits enter or clicks elsewhere	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system sets its date range according to the values in the start and end date boxes.	System set its date range according to the values in the start and end date boxes.	P
	4.2	User filters first and last letter of member name with desired letters	CSV file is already loaded	1. The user clicks on member name text fields and enters in desired values 2. The user hits enter or clicks elsewhere	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system sets its member name range according to the values in the first and last letter of member last name boxes.	System set its member name range according to the values in the first and last letter of member last name boxes.	P
CHECK CUSTOM SORT ORDER FUNCTIONALITY	5.0							P
	5.1	User specifies a new sort order and saves it under a specified name	CSV file is already loaded	1. The user clicks Create New Sort Order button. 2. The system displays a new sort order screen. The user enters the name of the new sort order, selects the hierarchy of filters to order the sort by, and clicks the Save button.	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system closes the new sort order screen. The system adds the new sort order to the list of existing sort orders which is then visible from the list of existing sort orders.	System closed the new sort order screen. The system added the new sort order to the list of existing sort orders which was then visible from the list of existing sort orders.	P
	5.2	User specifies a new sort order and attempts to save it without specifying a name	CSV file is already loaded	1. The user clicks Create New Sort Order button. 2. The system displays a new sort order screen. The user does not enter the name of the new sort order, selects the hierarchy of filters to order the sort by, and clicks the Save button.	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system displays an error message.	System displayed an error message.	P
	5.3	User chooses a sort order from the list of existing sort orders (which includes previously saved ones)	CSV file is already loaded	1. User selects the sort order from the list of existing sort orders.	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system sets its sort order to the one selected in the list of existing sort orders.	System sets its sort order to the one selected in the list of existing sort orders.	P
CHECK VISUALIZATION OF DATA FUNCTIONALITY	6.0							P
	6.1	User selects an element (e.g. a member name) in the dashboard summary view to visualize its related data	CSV file is already loaded	1. The user clicks on an element in the dashboard summary view	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system displays a visualization (default is Pie Chart) of the selected element	System displayed a visualization (default is Pie Chart) of the selected element	P
	6.2	User wishes to change visualization type to a bar graph or pie chart	CSV file is already loaded	1. The user clicks on an element in the dashboard summary view 2. The user clicks on desired visualization option (pie chart or bar chart)	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system displays chosen visualization of the selected element(s).	System displayed chosen visualization of the selected element(s)	P
	6.3	User wishes to visualize data using a line graph	CSV file is already loaded	1. The user clicks on an element in the dashboard summary view 2. The user clicks on line graph option	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system displays the selected element(s) using a line graph	The system displayed the selected element(s) using a line graph	P
CHECK PDF EXPORT FUNCTIONALITY	7.0							P
	7.1	User attempts to save bar chart visualization to a PDF and specifies a name of the PDF	CSV file is already loaded, member(s) selected, bar chart is selected	1. The user clicks the Export PDF button 2. The user selects a file path, enters a file name and clicks the Save button	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system displays a File Explorer screen. The system exports the selected visualization type to a PDF with the entered file name at the selected file path	System displayed a File Explorer screen. The system exported the selected visualization type to a PDF with the entered file name at the selected file path	P
	7.2	User attempts to save pie chart visualization to a PDF and specifies a name of the PDF	CSV file is already loaded, member(s) selected, pie chart is selected	1. The user clicks the Export PDF button 2. The user selects a file path, enters a file name and clicks the Save button	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system displays a File Explorer screen. The system exports the selected visualization type to a PDF with the entered file name at the selected file path	System displayed a File Explorer screen. The system exported the selected visualization type to a PDF with the entered file name at the selected file path	P
	7.3	User attempts to save visualizations to a PDF but does not specify a name for the PDF	CSV file is already loaded, member(s) selected	1. The user clicks the Export PDF button 2. The user selects a file path, does not enter a file name and clicks the Save button	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system displays an error message.	The Save Button is unavailable if the user does not specify a file name (tested on Mac OS)	P
CHECK PRINT FUNCTIONALITY	8.0							P
	8.1	User attempts to print bar chart visualization	CSV file is already loaded, member(s) selected, bar chart is selected	1. The user clicks the Print button 2. The user selects a file path (default is current working directory), enters a file name (default is "print") and clicks the Print button 3. The user clicks print	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system displays the OS's print screen. The system prints the selected visualization type.	System displayed the OS's print screen. The system sent the file to OS's print handler.	P
	8.2	User attempts to print pie chart visualization	CSV file is already loaded, member(s) selected, pie chart is selected	1. The user clicks the Print button 2. The user selects a file path (default is current working directory), enters a file name (default is "print") and click the Print button 3. The user clicks print	[Teaching_compat_errors.csv; Publications_compat_errors.cs; Presentations_compat_errors.csv; GrantsClinicalFunding_compat_errors.csv]	The system displays the OS's print screen. The system prints the selected visualization type.	System displayed the OS's print screen. The system sent the file to OS's print handler.	P



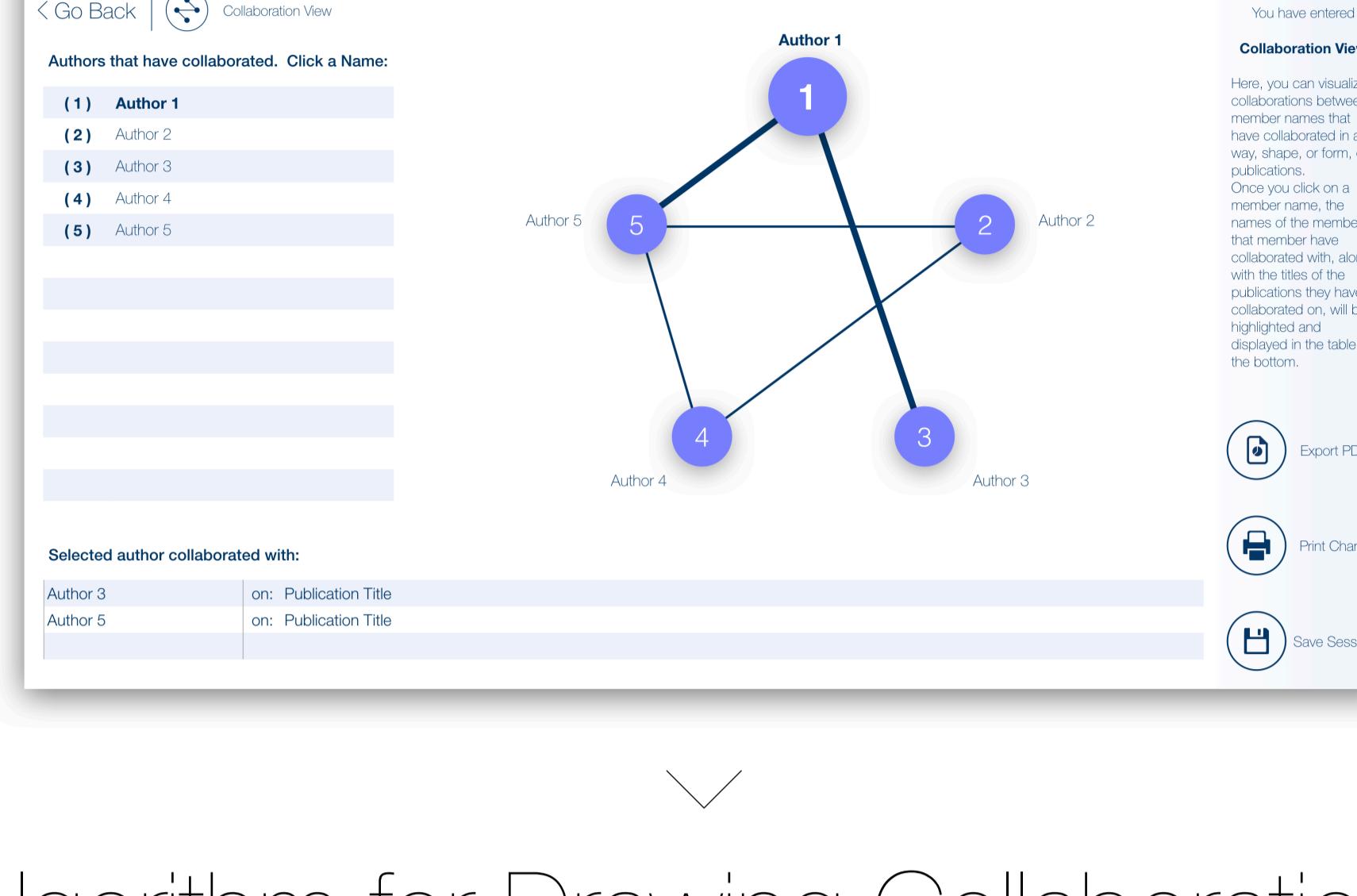




Authentication Sequence

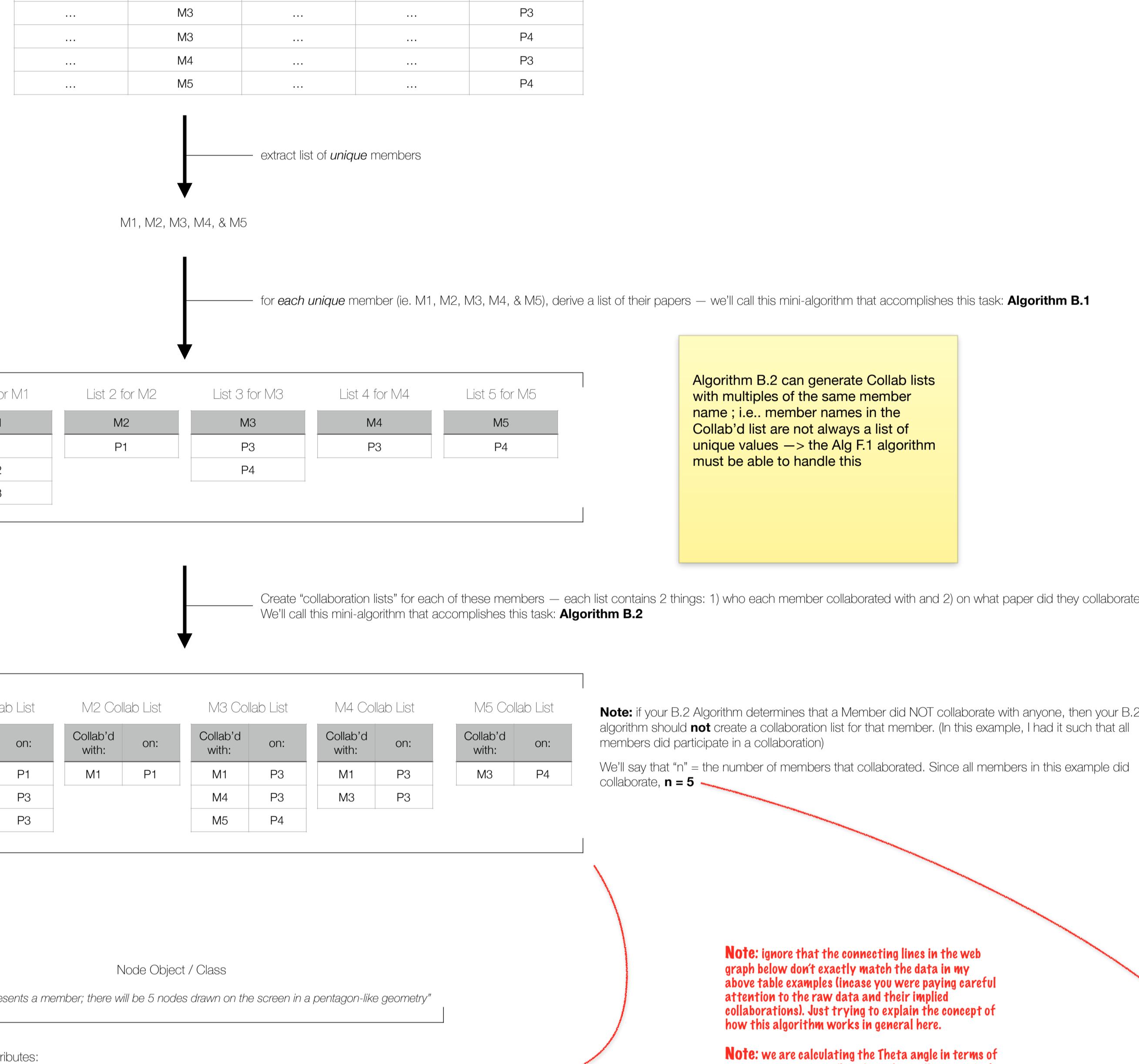


Goal: Draw Collaboration Web Graph

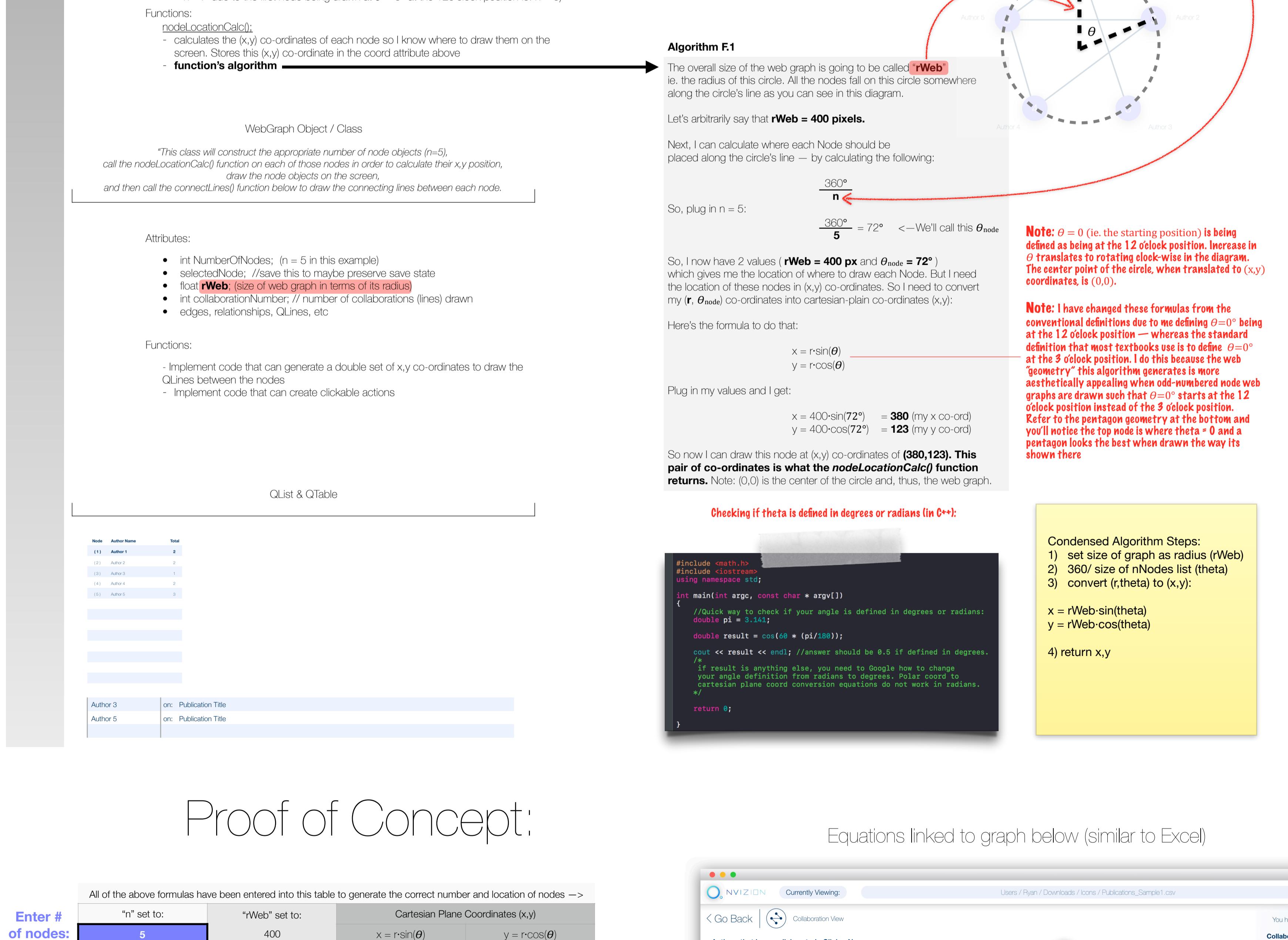


Algorithm for Drawing Collaboration Web Graph

Back End



Front End



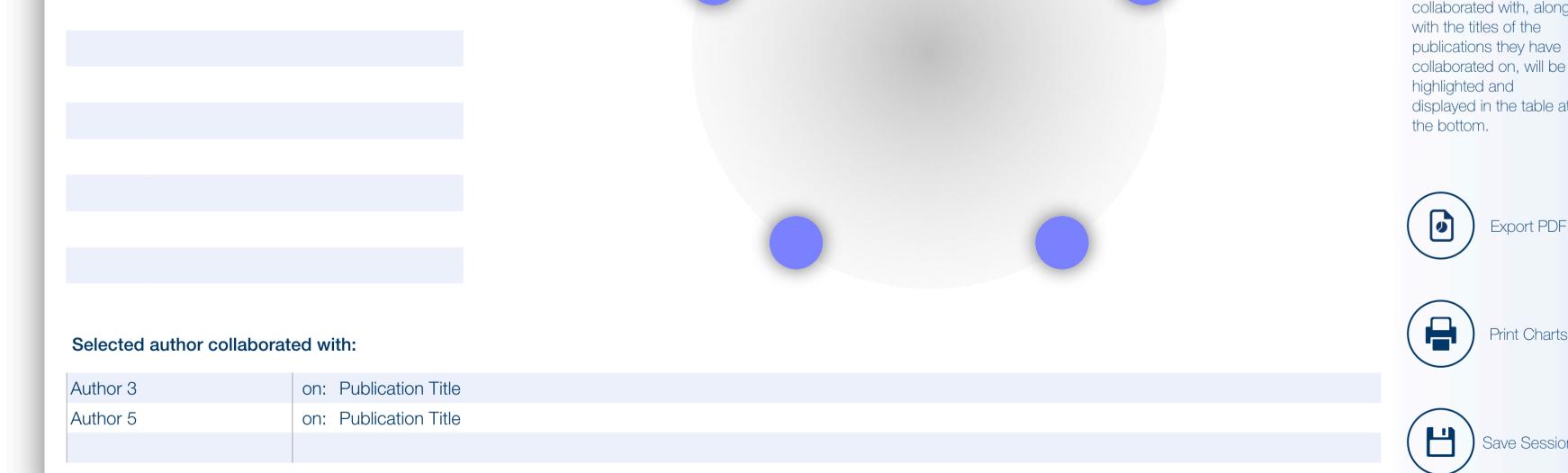
Proof of Concept:

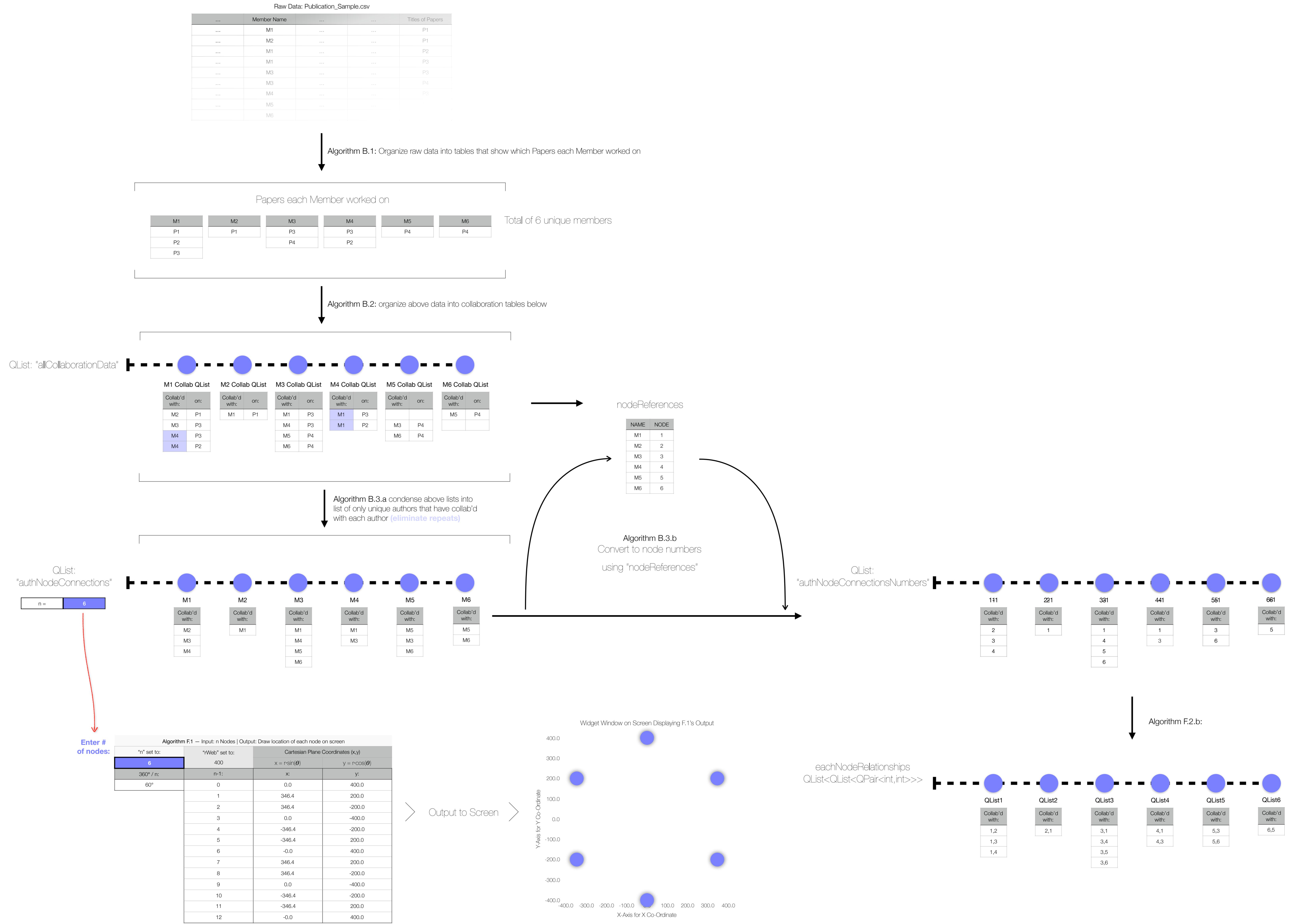
All of the above formulas have been entered into this table to generate the correct number and location of nodes -->		
"n" set to:	"rWeb" set to:	Cartesian Plane Coordinates (x,y)
5	400	x = rsin(theta) y = rcos(theta)
360°/ n:	n-1:	x: y:
72°	0	0.0 400.0
	1	380.4 123.6
	2	235.1 -323.6
	3	-235.1 -323.6
	4	-380.4 123.6
	5	0.0 400.0
	6	380.4 123.6
	7	235.1 -323.6
	8	-235.1 -323.6
	9	-380.4 123.6
	10	0.0 400.0
	11	380.4 123.6
	12	235.1 -323.6

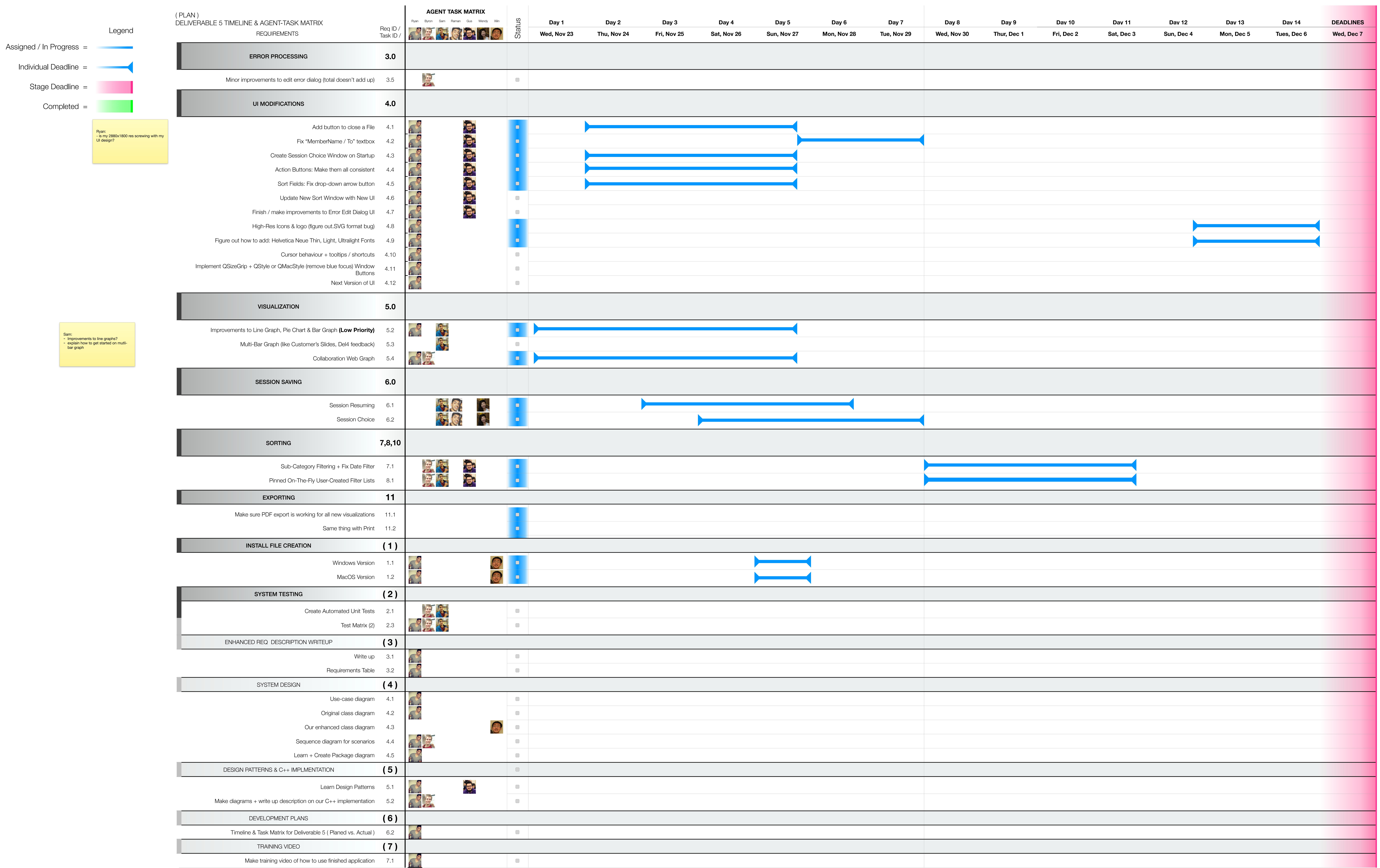
Note: this is all done using Numbers, which is Apple's version of Excel.

Free download on App Store for Mac & iOS

Equations linked to graph below (similar to Excel)







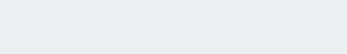
 = In Progress

= Individual Deadline

= Stage Deadline

TIMELINE & AGENT-TASK MATRIX

AGENT TASK MATRIX

REQUIREMENTS	Task ID		Mon, Sep 12
Questions for Customer	(1)		
3 Requirements Questions	1.1	<input checked="" type="checkbox"/>	
2 System Questions	1.2	<input checked="" type="checkbox"/>	

Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13
Tue, Sep 13	Wed, Sep 14	Thu, Sep 15	Fri, Sep 16	Sat, Sep 17	Sun, Sep 18	Mon, Sep 19	Tue, Sep 20	Wed, Sep 21	Thu, Sep 22	Fri, Sep 23	Sat, Sep 24

TIMELINE & AGENT-TASK MATRIX

Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13
Tue, Sep 13	Wed, Sep 14	Thu, Sep 15	Fri, Sep 16	Sat, Sep 17	Sun, Sep 18	Mon, Sep 19	Tue, Sep 20	Wed, Sep 21	Thu, Sep 22	Fri, Sep 23	Sat, Sep 24

 = In Progress

= Individual Deadline

| = Stage Deadline

ANNED)
VERABLE 2 TIMELINE & AGENT-TASK MATRIX

GENT TASK MATRIX

A Gantt chart illustrating a project timeline from Day 1 to Day 22. The chart is divided into several sections:

- Day 1 to Day 7:** Short tasks (1-7) represented by blue bars. Each bar has a black arrowhead at its start.
- Day 8 to Day 10:** Three parallel tasks (8-10) represented by blue bars. Each bar has a black arrowhead at its start.
- Day 11 to Day 14:** A large section containing three parallel tasks (11-14) represented by blue bars. Each bar has a black arrowhead at its start.
- Day 15 to Day 18:** Two tasks (15-18) represented by blue bars. Each bar has a black arrowhead at its start.

The x-axis is labeled with dates corresponding to each day:

- Day 1: Thur, Sep 29
- Day 2: Fri, Sep 30
- Day 3: Sat, Oct 1
- Day 4: Sun, Oct 2
- Day 5: Mon, Oct 3
- Day 6: Tue, Oct 4
- Day 7: Wed, Oct 5
- Day 8: Thur, Oct 6
- Day 9: Fri, Oct 7
- Day 10: Sat, Oct 8
- Day 11: Sun, Oct 9
- Day 12: Mon, Oct 10
- Day 13: Tue, Oct 11
- Day 14: Wed, Oct 12
- Day 15: Thur, Oct 13
- Day 16: Fri, Oct 14
- Day 17: Sat, Oct 15
- Day 18: Sun, Oct 16
- Day 19: Mon, Oct 17
- Day 20: Tue, Oct 18
- Day 21: Wed, Oct 19
- Day 22: Thur, Oct 20
- Deliverable 2 Due: Fri, Oct 21

(ACTUAL) DELIVERABLE 2 TIMELINE & AGENT-TASK MATRIX

AGENT TASK MATRIX

-  = In Progress
-  = Individual Deadline
-  = Stage Deadline

(PLANNED) DELIVERABLE 3 TIMELINE & AGENT-TASK MATRIX

= In Progress

= Individual Deadline

= Stage Deadline

(ACTUAL) DELIVERABLE 3 TIMELINE & AGENT-TASK MATRIX

-  = In Progress
-  = Individual Deadline
-  = Stage Deadline

(PLAN) DELIVERABLE 5 TIMELINE & AGENT-TASK MATRIX		Req ID / Task ID /	AGENT TASK MATRIX	Status	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	DEADLINES
REQUIREMENTS					Wed, Nov 23	Thu, Nov 24	Fri, Nov 25	Sat, Nov 26	Sun, Nov 27	Mon, Nov 28	Tue, Nov 29	Wed, Nov 30	Thur, Dec 1	Fri, Dec 2	Sat, Dec 3	Sun, Dec 4	Mon, Dec 5	Tues, Dec 6	Wed, Dec 7
Assigned / In Progress =																			
Individual Deadline =																			
Stage Deadline =																			
Completed =																			
ERROR PROCESSING		3.0																	
Bug Fixes to edit error dialog		3.5																	
Reading New CSV Files		3.6																	
FRONT END		4.0																	
Add button to close a File		4.1																	
Fix "MemberName / To" textbox		4.2																	
Create Session Choice Window on Startup		4.3																	
Action Buttons: Make them all consistent		4.4																	
Sort Fields: Fix drop-down arrow button		4.5																	
Update New Sort Window with New UI		4.6																	
Finish / make improvements to Error Edit Dialog UI		4.7																	
High Res Icons & logo (figure out SVG format bug)		4.8																	
Figure out how to add Helvetica Neue Thin, Light, Ultralight Fonts		4.9																	
Cursor behaviour + tooltips / shortcuts		4.10																	
Next Version of UI w/ filter/sorts moved to above table with new filters		4.11																	
Style PDF exports		4.12																	
make sure screen resolution / size is adjusted for (one simple strategy to do this is to start app in device's fullscreen mode to avoid differences in laptop screen sizes)		4.13																	
VISUALIZATION		5.0																	
Improvements to Line Graph, Pie Chart & Bar Graph (switch x with y axis for bar graph (Low Priority))		5.1																	
Multi-Select Graph for pie, bar & line graph (like Customer's Slides, Del4 feedback)		5.2																	
Design Collaboration Web Graph Algorithm		5.3																	
Implement FrontEnd & BackEnd of Collaboration Web Graph		5.4																	
Link FrontEnd & BackEnd, impose max and min limits on data, run testing		5.5																	
SESSION SAVING		6.0																	
Session Resuming		6.1																	
Session Choice		6.2																	
FILTER/SORTING		7,8,10																	
Sub-Category Filtering + Fix Date Filter		7.1																	
Pinned On-The-Fly User-Created Filter Lists		8.1																	
EXPORTING		11																	
Make sure PDF export is working for all new visualizations		11.1																	
Same thing with Print (this probably will work fine if the above works fine)		11.2													<img alt="User				

