**Analysis of Phase 1**

**Presented by: Team Orion**



**Table of Contents**

[**Summary & Functionality:**](#_pcn1bokf4p3p) **2**

[**Language used (How it was designed/created)**](#_6r1hwlao6zxe) **3**

[**GUI (Graphical User Interface)**](#_m86ey252bdf9) **4**

[**Documentation**](#_cn95ge71f043) **5**

[**Error handling**](#_26mjcw76rxxp) **6**

[**Conclusion**](#_a3ze97we86mr) **7**

# 

# 

# Summary & Functionality:

The software of “Peachy Galaxy” was created to assist the Schulich school of medicine and dentistry of choreographing the viewing and editing of the organization’s data files in a fast and easy manner. Specific data files, pertaining to the school, are accessed through the program from their respective fields and generated to the main page for viewing. Using this software, filters can be applied to display the data differently as needed by the user for convenience. An aesthetic this software offers is the continuous utilization of pie/bar charts to represent any portion of focused data. It is recognized that this program will only be used by authorized members of the Shulich school, and will therefore not require any security to enter the program. Third-party software utilized in this program was that of “Acuity Star”.

# Design Analysis

**Information/data hiding:** It seems that reasonable efforts have been made to keep functions & attributes private when possible.

**Encapsulation/Abstraction:** Most of the public members have a use in other parts of the program.

**Modularity:** The obvious way to model the data would have been to create class for each of the 4 record types (teach, pub, pres & grant) and use each of the columns of the spreadsheet as an attribute of the class. Specific attributes could then have easily been found and retrieved with ease. No classes for Department, Employee, Course etc. were implemented. Team Peach’s implementation of using trees of vectors of strings is overly contrived and unintuitive.

Additionally, RecordsManager and MainWindow are “megaclasses” which could potentially be trimmed down or split up.

**Cohesion/Decoupling:**  The source code is highly coupled. Most classes store and exchange data in highly specialized, elaborate forms. The entire program is highly tailored to work with the specific input format and their chosen UI structure (with expandable and collapsable trees). Changing either the input or the output will cause major problems throughout the program.

**Efficiency:** The program performs its processes quickly enough that this is not an area of significant concern.

**Reusability:** The class structure does not serve as a model of real world concepts but is highly tied to the input format (.csv files). In the event that the customer wanted to use another method to provide the input (e.g. get the data through use of an API or directly from a database system), the entire datamodel and database sections of the program would have to be scrapped and restarted from scratch.

**Readability/Maintainability:** Comments in the source code are few and far between. Most methods and steps are not explained. Variable and function identifiers are not descriptive. The program has implemented an unnecessarily complex data model for an input that was relatively simple. This is a major obstacle to debugging, modifying and maintaining the source code. If Schulich decides there are additional features they would like to implement, this could become highly problematic.

# GUI (Graphical User Interface)

Peachy Galaxy has a user friendly GUI. There are four subject areas that are shown in the system: teaching, publications, presentations, and grants and clinical funding. When users choose subject areas, users need to load a .csv file then the data information will be shown at the bottom part of the window. The left side will show identity name and their attribute values, you can select an item in this window. And the right side will show charts such as pie or bar chart. In the middle of the window, you can customize your sort option, and create new sort order. The sort order you created will be saved automatically by the system, which is, if you restart the system, the sort order you created will still be there.

Although the system has been successful in implementing functions to help users to read data, there are some weaknesses in the system. After the user loads a .csv file, it may show a warning message that the file is missing mandatory fields, user can choose discard or edit. If users choose edit; however, after the edit erroneous fields window show up, the user cannot actually change the data in the field. Another problem is that, when we choose an item in the left part of the window, the right part can show chart, you are able to expand the item in the left part of window, it will show specific field of data information; however, if you choose specific fields data, then the right part of window, which should show chart, will show nothing.

Pie/Bar Chart: These charts accurately represent the data, however, data types are not clearly shown to the user. These data types include courses taught, number of presentations, grants etc. Data on the bar charts are aesthetically pleasing and accurate whereas the pie charts can be messy at times.

# Documentation

On a conceptual level, the contributors on the Peach Galaxy project did a good overview of the software in their “Final System Delivery” retrospective. They discuss some of the difficulties they faced so that future developers will have a better idea of what they are dealing with. They also give a detailed analysis of what the classes, in the project, do and also of how they interact with each other using diagrams. There are detailed examples in the form of use-cases of how the software works from the perspective of the Schulich employees who will be using the software.  
  
 Ultimately these examples are rather more technical than a user would need to do their work so higher level user oriented documentation would have been more useful.  
  
 Additionally, from the perspective of future maintainers of the Peach Galaxy project who will actually dig into the code to add features, the documentation found within is lacking in comments and explanation of certain methods and classes. The commenting of the code is very inconsistent, with some classes appearing to follow some kind of standard, while others are completely transparent in any useful information. Some of the comments are also rather unprofessional and suggest a lack of polish and testing of the code.  
  
 Before adding the required features of Phase II, Team Orion will go over the code from Phase I and comment it and format it in such a way that those developers in future phases will have an easier time understanding how the code works. We will also define a standard of commenting for the project for the sake of consistency. We will also provide user friendly documentation so that Schulich administration can be more productive more quickly.

# Error handling

When the CSV file, to be imported into the Peach Galaxy program for analysis, has incomplete fields, the software prompts the user to either ignore these or edit the fields. However, when a user chooses to edit the fields, they are required to edit every single offending one or none at all. Should the user only want to edit certain values, they would be at a loss.

In the attempt of a file being accessed by the wrong field, in the program, a pop-up message will be displayed to the user that the file is not a valid “area specific” file. The “Publications” field was most notably susceptible to crashing the program when importing a file, more specifically the release version.

When it came to exporting data, users were allowed to export selected data that had no information. The benefit to this program is that if a file is not inputted for use, the “print” and “export” functions will not be allowed to be utilized by the user.

# Conclusion

Overall, the “Peachy Galaxy” program met its foundation requirements the school needed but lacks in certain areas. Areas that were found to be positive were the functionality and the GUI of the program. Error handling and Documentation were not up to the standard the school was hoping for. Phase 2 of this project hopes to improve drastically current issues that need attention to keep the customer of Schulich content.