**University of Houston Clear Lake**

Spring ‘22

CSCI 4388.02

Senior Project in Computer Science

**Design Document**



The Data Extractors

03/09/2022

1. **Upper Level Class Diagrams**

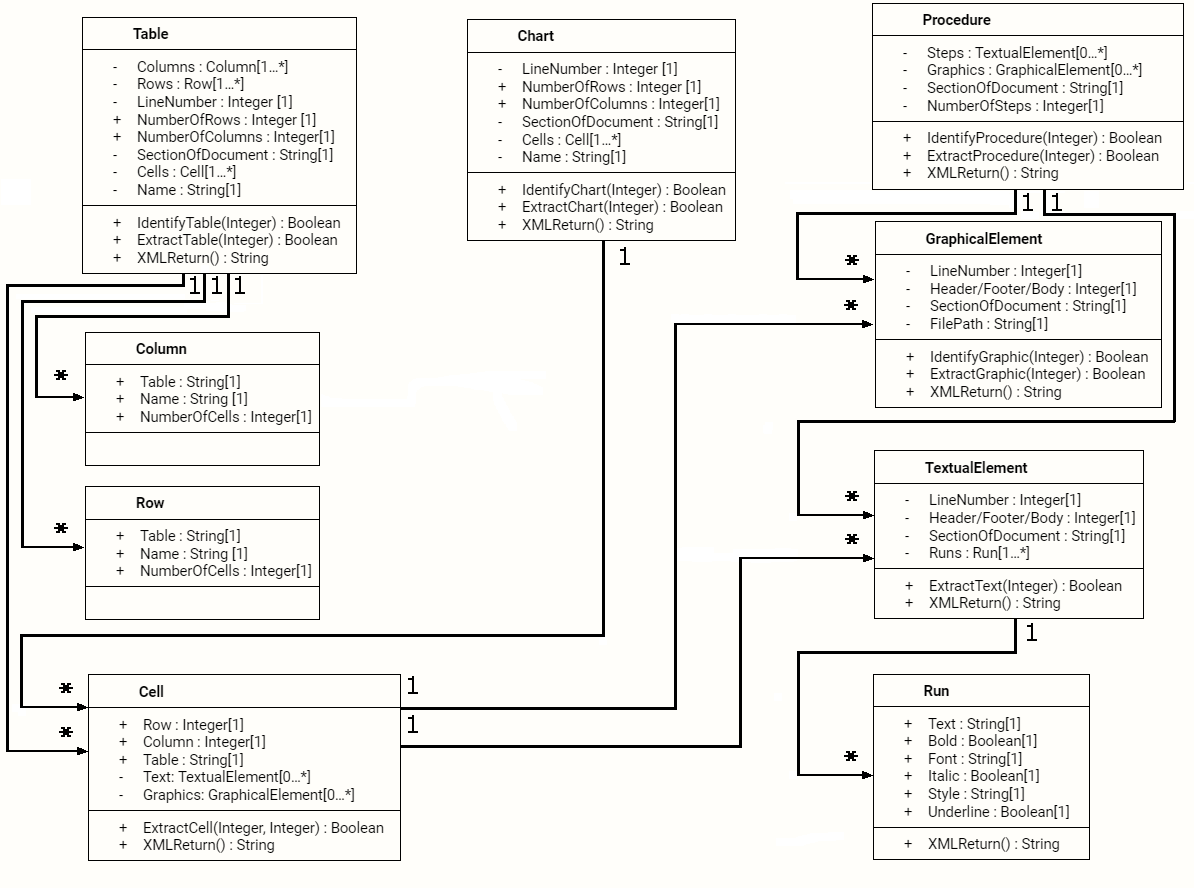
| **VariableUtility** |
| --- |
| * OutputFileLocation : String[1] * OutputFileName : String[1] * InputFilePath : String[1] |
| * OutputCheckValidity() : Boolean * InputCheckValidity() : Boolean * EndProgram() : Void |

| **Parser/Labeler** |
| --- |
| * XMLStrings : String[1…\*] |
| * ParseLabelFile(String) : Boolean * SaveXML() : Boolean |

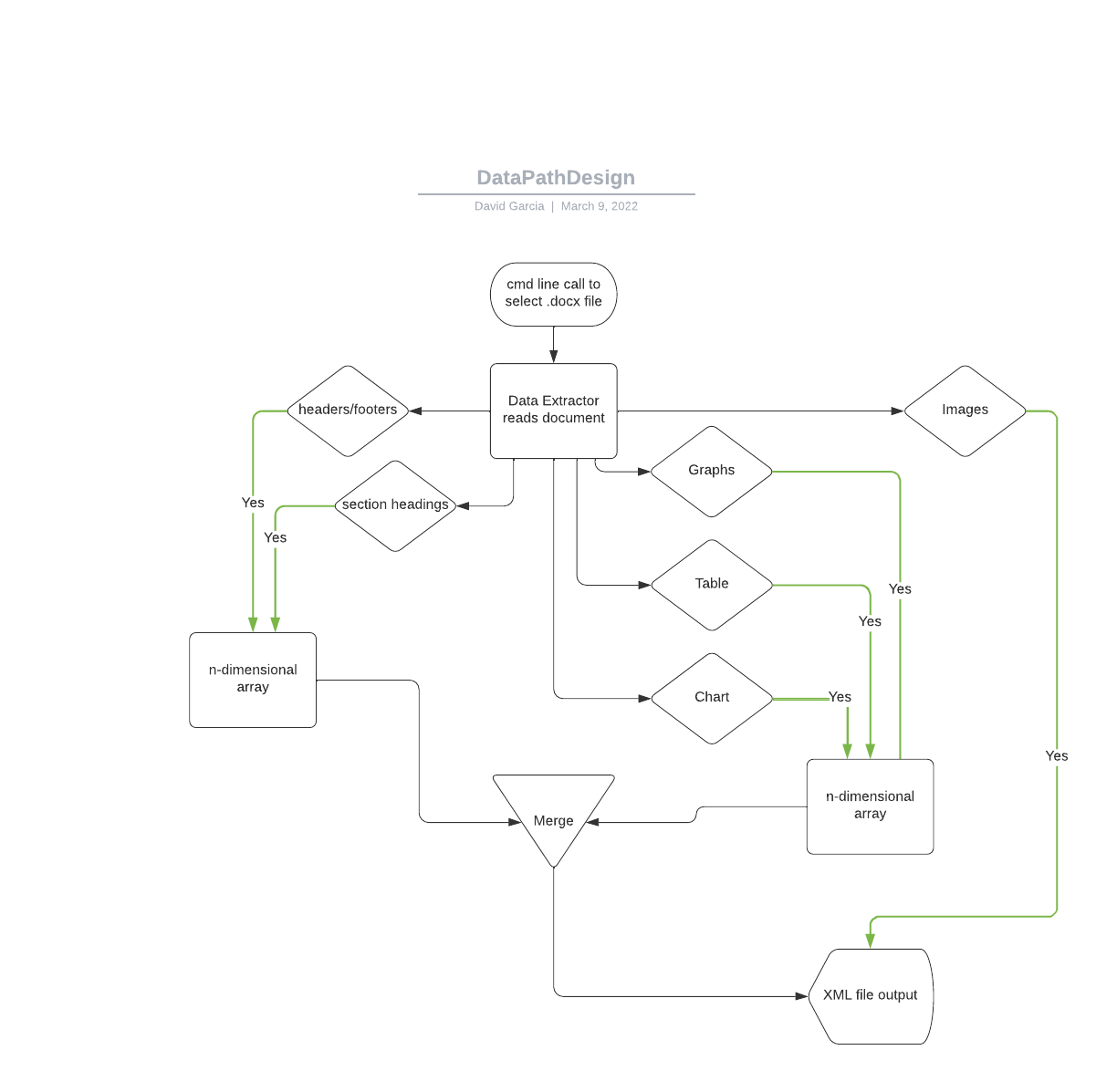
| **Interpreter** |
| --- |
| * CommandList : String[\*] |
| * InterpretCommand(String) : Boolean * CheckCommandValidity() : Boolean |

| **Help** |
| --- |
| * HelpList : String[\*] |
| * HelpDisplay(Integer) : String |

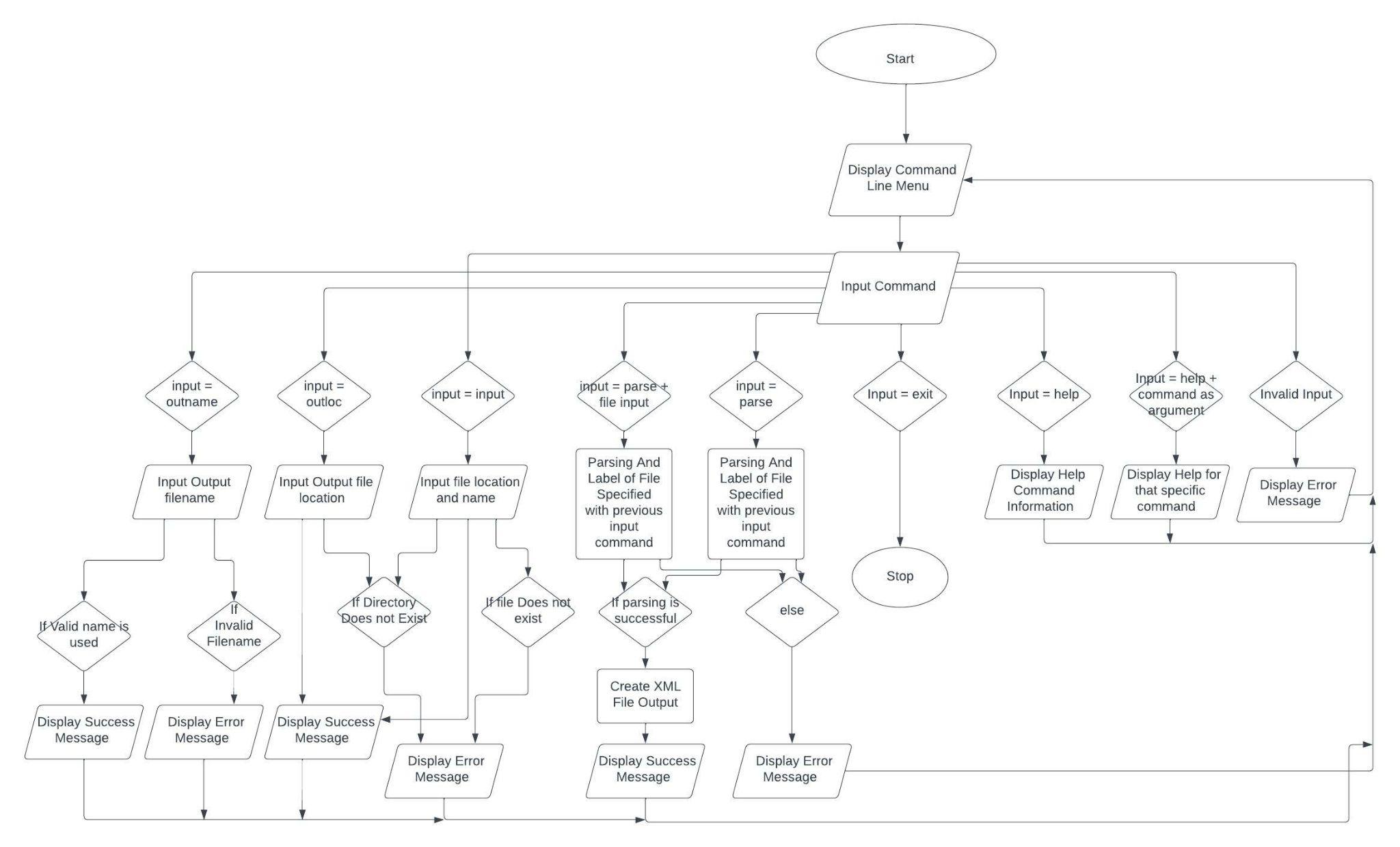
1. **Lower Level Class Diagrams**



1. **Dataflow Communication Diagram**

[](https://lucid.app/documents/edit/de33159a-650d-4cba-8f95-890e5090632a/0?callback=close&name=docs&callback_type=back&v=945&s=612)

1. **Flowchart Sequence Diagram**



1. **XML Schema Layout**

This section gives a general overview of the various types of XML elements we will need to create and use in the Word Text Data Extractor’s output XML file.

Examples of an ‘element’ in XML:

<Employee>

<Name>John Smith</Name>

<Salary>75000</Salary>

<IDNum>12345</IDNum>

</Employee>

In this example ‘Employee’, ‘Name’, ‘Salary’, and ‘IDNum’ are all elements, with the ‘Name’, ‘Salary’, and ‘IDNum’ elements being used to store data about an instance of an ‘Employee’ element.

Here are the types of XML elements we will likely need to use in our output XML file:

**Text – Elements relating to the text from the Word documents passed through the Data Extractor and output in the XML schema**

* Text - String; The actual string of text itself
* Font – String; The type of font that the text is in
* Style – String; The style that the text is in
* Bold – Boolean; Whether or not the text is bolded
* Italics – Boolean; Whether or not the text is italicized
* Underline – Boolean; Whether or not the text is underlined

**Tables & Charts – Elements relating to how the XML schema will output the data from a table or chart found in a Word document passed through the data extractor**

* Name/Label – String; Name/Label of the table
* SectionOfDocument – String; What section of the Word document the table is in
* Location – Int; What line the table starts on in the Word document
* Rows
  + Table – String; What table the row belongs to
  + Name - String; The name/label of the row
  + NumberOfCells - Integer
* Columns
  + Table – String; What table the column belongs to
  + Name - String; The name/label of the column
  + NumberOfCells – Integer
* Cells
  + Row - Integer; Which row of the table the cell is in
  + Column - Integer; Which column of the table the cell is in
  + Table - String; The table the cell belongs to
  + Text - String; Text data contained in the cell
  + Graphics - String; Filepath of the graphic after it has been taken out the Word document and saved as its own file by the Data Extractor
* NumberOfColumns - Integer
* NumberOfRows - Integer

**Procedures/Lists – Elements relating to how the XML schema will output a set of procedural instructions (Bulleted list, Numbered list, Checklist, etc.) found in a Word document when passed through the Data Extractor**

* Steps - String; The actual steps of the procedure, written out as a string
* Graphics - String; A reference with the filepath of where the graphic is saved
* NumOfSteps - Integer; Number of steps in the procedure
* SectionOfDocument - String; What section of the document the procedure is in

**Graphics – Elements relating to how the XML schema will output references to graphics found in Word documents passed through the Data Extractor**

* LineNumber – Integer; What line of the Word document the graphic begins on
* Header/Footer/Body – Integer; Number indicates which section of the document the graphic is in (0 for body, 1 for header, 2 for footer)
* SectionOfDocument – String; What section the graphic is in
* FilePath – String; Filepath of the graphic after it has been taken out the Word document and saved as its own file by the Data Extractor

1. **Version Control**

We chose to use GitHub as our version control system. We’ll be using the GitHub repository located at the URL <https://github.com/AustinFoxtrot/Word-Data-Extractor> for this project.