1. Choose excel file
2. Parse template excel file
   1. Parse excel file
   2. Must look for tags which say what info we need from the source xml file
   3. Must collect color and sort information as well.
3. Use information from template to Parse source xml file
   1. Parse xml source file from Altium.
   2. Collect data associated with tags from template file
4. Sort source file by several listed factors in template file
   1. Sort body information by several layers of sorting by priority
   2. Must be able to sort by ascending, descending, and by specific characters.
   3. System counts identical parts and updates a column for quantity.
      1. For identical parts, the top row will contain the quantity, all bottom rows will have empty space in the quantity section.
5. Create excel file and populate with sorted info by template settings
   1. Create new excel EBOM file using original template settings.
   2. Populate excel with new source info
   3. Modify cells with body colors
   4. Verify the amount of rows created matches amount of components found in the source xml
   5. Create footer floating note at bottom of body
   6. Resize all columns
6. Provide ability for program to be started by double clicking source xml.

Stereotype Classes:

Actor: User

Interface: Launch screen

Report: EBOM Excel File

Business:

Excel file handler

Xml file handler

Excel parser

Xml parser

Sort

Count parts

Template attributes

Text Use Case: choose source .xml file

Primary actor: user

Description:

1. User has started the program
2. User selects button to choose source xml
3. System opens file browser window
4. User selects source xml
5. System populates text box with selected file path.

Text Use Case: Parse template excel file

Primary actor: user

Description:

1. User has chosen excel file path.
2. User selects start button
3. System opens excel file.
4. System reads excel cells info one at a time.
5. System looks for specific tags which indicate:
   1. Which data to look for in the xml source file.
      1. First by title block
      2. Second by header row
   2. How the data is to be sorted in the body of the excel file.
      1. Collect info on which column is to be sorted by priority
         1. Sorting methods to include are ascending, descending, custom character list.
   3. How far to search in the excel file.
6. System looks for certain cell attributes such as color and border if necessary.
7. System populates filter settings and sort settings for parsing source xml.

Text Use Case: Parse source xml file

Primary actor: parse template excel file

Description:

1. User has chosen source file path and the sorting and filter setting have been found.
2. System opens xml file.
3. System read xml by tag.
4. System collects data for the information that the template requires.
   1. Collect the info for the title block.
5. System collects info which the header row states will be required in the body of the EBOM excel file.

Text Use Case: Sort source file

Primary actor: parse source xml

Description:

1. The source file information has been collected and the sort settings are found.
2. System sort collected body information by sort settings
3. System counts identical parts and updates a column for quantity.
   1. The top row will contain the quantity, all bottom rows will have empty space in the quantity section.

Text Use Case: Create excel EBOM file

Primary actor: Sort source file

Description:

1. The source xml info is sorted and ready for input into an excel file
2. System copy’s source template and give it the file name of the source xml.
3. System replaces all header tags with the require information in the title block.
4. System removes unnecessary tags in the header row and the elsewhere.
5. System modifies all body cells with necessary color pattern.
6. System populates body rows with sorted list of components from source xml.
7. Verify the amount of rows created matches amount of components found in the source xml
8. System creates floating note with specific color and text recorded from source template and places it at the end of the body.
9. Resize all columns.

0. User selects start button

2. System opens excel file.

3. System reads excel cells info one at a time.

4. System looks for specific tags which indicate:

a. Which data to look for in the xml source file.

i. First by title block

ii. Second by header row

b. How the data is to be sorted in the body of the excel file.

i. Collect info on which column is to be sorted by priority

1. Sorting methods to include are ascending, descending,custom character list.

c. How far to search in the excel file.

5. System looks for certain cell attributes such as color and border if

necessary.

6. System populates filter settings and sort settings for parsing source xml.