Daniah Mokhtar ***Solution Manual***

3esi Project

Scenario

3esi-Enersight is building a new tool to help import entities into our applications. The design calls for a system that can import a comma separated value (CSV) file containing the relevant entity information. A summary of the import should be created, informing the user of any successes and failures during the process. The import should not terminate if a business rule is violated.

Functional Requirements

Write a program that:

Parses inputted csv file:

- Comma separated
- Has 2 Types: Well and Group
- Order of parsing does not matter
- Names are unique across both wells and groups in the processed file
- Locations are unique in each type
- Groups cannot overlap
- A well is considered a child of a group if the well's top hole location falls within the area of the group defined by its radius
- The well type is determined by the distance between the top and bottom hole locations with the following tolerances:
  - Vertical: 0 – 1
  - Slanted: 1 – 5
  - Horizontal: >=5
- Log errors

Design Decisions & Assumptions

- Used **FileHelpers** (https://www.filehelpers.net/download/) C# Nuget package to parse the files (proven to be fast and reliable)
- Reading only 1 file at each run (for simplicity and demoing the functionality needed)
- csv file is included in Daniah_ConsoleApp/UploadedFiles/3esi.csv
- use the provided csv titles to create the record modeling classes
- Because this is a multi-record csv file, it is not read as an asynchronous file so the performance might be affected for large csv files
- Assumption for the well type that "slanted" includes 1 but vertical does not based on the last condition "horizontal" which includes 5
- Assumption for overlapping group we will keep the first group and all others that overlap with it will be added to error list
- Display output
  - Wells that do not belong to a group
  - Groups and their children wells
  - Parsing Errors
  - Business Rules Errors

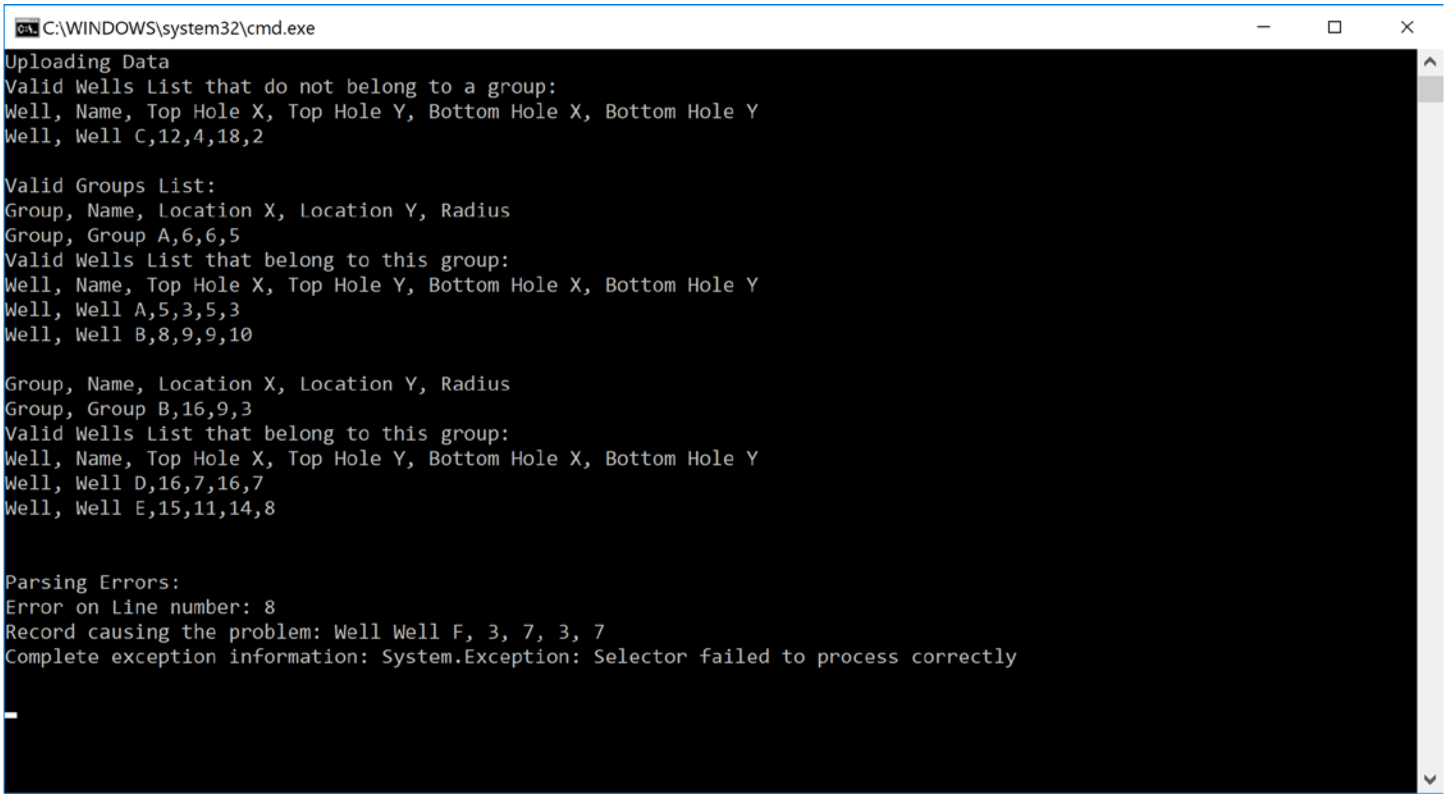
Solution Structure

- Create a class library named **3esi_BusinessLayer** that parses and applies the business rules.
- Create a console application named **3esi_ConsoleApp** that takes the file name from the user and prints out the results.
- Create a unit testing project to test the functionalities for 3esi_BusinessLayer project named **3esi.Tests**

To build and execute the code project

- Please open **Daniah_3esi** main project folder
- To run the project from visual studio 2017, open the project solution **3esi.sln**
- Ensure that **3esi_ConsoleApp** is set as the StartUp project. If not, right click on that project and set it as the start-up project by selecting “Set as StartUp Project”
- CTRL+F5 to run the project or F5 to start a debugging session
- To run the test project, right-click on **3esi.Tests** project and select “Run Unit Tests”

Screen Shot of Output



```
Uploading Data
Valid Wells List that do not belong to a group:
Well, Name, Top Hole X, Top Hole Y, Bottom Hole X, Bottom Hole Y
Well, Well C,12,4,18,2

Valid Groups List:
Group, Name, Location X, Location Y, Radius
Group, Group A,6,6,5
Valid Wells List that belong to this group:
Well, Name, Top Hole X, Top Hole Y, Bottom Hole X, Bottom Hole Y
Well, Well A,5,3,5,3
Well, Well B,8,9,9,10

Group, Name, Location X, Location Y, Radius
Group, Group B,16,9,3
Valid Wells List that belong to this group:
Well, Name, Top Hole X, Top Hole Y, Bottom Hole X, Bottom Hole Y
Well, Well D,16,7,16,7
Well, Well E,15,11,14,8


Parsing Errors:
Error on Line number: 8
Record causing the problem: Well Well F, 3, 7, 3, 7
Complete exception information: System.Exception: Selector failed to process correctly
```