Agenda

1) We will send you this assignment one day before the on-site coding test. This will give you some time to think about the requirements and do necessary preps.
2) You will do the coding in our office on November 14, 2017.
3) Our development team will review your code.
4) We will organize a meeting for you to present your work and answer questions.

General Information

1) This coding test is open book and done in our office. You can bring in your books, laptop, etc.
2) A project has been scoped for this test. See the scope below.
3) The duration of the test is one work day in office including presentations. Extra time can be requested if you need. 4) You are expected to write professional program.
5) You can make it a Window Application or Web Application.
6) C# is the preferred language for this test. However, you can use Java or C++ if that makes it easier for you.
7) You can use other technologies such as WPF/XAML, MVVM, DevExpress, Infragistics, ASP.NET, ADO.NET, Entity Framework, NHibernate, REST, Dashboard, etc.
8) You can download a free trial version if you want to use DevExpress/ Infragistics for UI design and/or reporting.
9) Other developers can help develop the UI if you don't feel comfortable with UI work. In this case, you and other developers will work together to hook your work to the UI.
10) You can add unit testing if you want.
11) Our development team will be available during the test to clarify the rules and the requirements.
12) At the end of the test, you need to turn in your work and give a presentation/demo about your design and implementation plan and results.
13) You can expand the original scope and add features that you think useful if you have extra time, but the minimum requirement defined in this assignment must be met.

When and What to Submit

You need to submit the following work at the end of the coding test:

1) Code.
2) A doc explaining your implementation plan, design and architecture, and how to run the project.

Criteria for Evaluation

- Are both the code and doc submitted?
- Does it satisfy the requirements?
- Does it run successfully?
- Is it maintainable?
- Can it scale?
- Is it your code and do you understand it? (You may be asked to discuss and explain any part of it.)
- If you use any code from a third-party source, you must note its source. As with your code, you may be asked to discuss and explain it.

Description

In this test, our expectation is two-fold. You are expected to provide a solution that will (1) manage the project data and their relations (2) read a sample project data, which has been saved in SQLite tables (Sample_Database.s3db), and present the data on the GUI (see sample snapshot in the Attachment). Note, the project data may be saved in other formats. The SQLite data was provided here just for the coding test purpose only.

1) The project may have multiple *mining areas* that represent the portion of the deposits that can be mined independently. There are a few mining area types based on the data sources and type of the deposits, such as Open Pit, Underground or GSM.
2) Each mining area may have a few *levels* that represent the vertical mining sequence.
3) Each level is divided into small mining units based on materials, grades and geometries, etc.
4) Each small mining unit is called a *cut*. A cut is a mining entity that has geometries, reserves and some other attributes.
    a. *Geometries* can be polygons, solids, or polylines. In this project, we assume that all cuts have polygonal geometry, which is a series of points.
    b. *Reserves* are tonnage, volume, and grades of each material. Only tonnage will be used in this test.
    c. Other attributes include *cut ID*, *cut name*, *cut level*, etc. They are all text type of data.

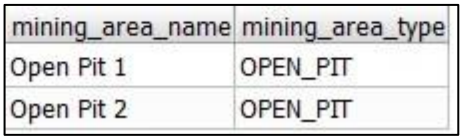
Goal

The goal is to create a solution, which will provide the following functionalities:

1) Create classes to manage the mining areas, levels, and cuts and the associated geometries, reserves and attributers, and the relations of these data.
2) Read the data mentioned above from a sample SQLite database so the data can be accessed by other objects. The SQLite database (Sample_Database.s3db) will be prepared for this test.
3) Provide helper functions to return the following information:
    a. Number of mining areas in a project.
    b. List of mining areas in a project.
    c. Number of levels in given mining area.
    d. List of levels in given mining area.
    e. Number of cuts on each level.
    f. List of cuts on each level.
    g. Cut ID
    h. Cut Name
    i. Cut Tonnage
    j. Cut Level
    k. Cut Geometry
    l. Calculating the area of a cut
4) Design GUI to present the data above.

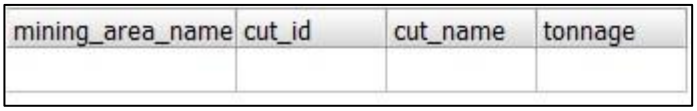
Attachments

t_mining_area



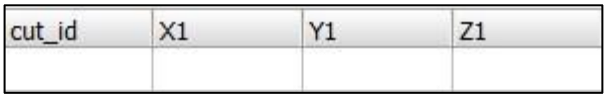
mining_area_name	mining_area_type
Open Pit 1	OPEN_PIT
Open Pit 2	OPEN_PIT

t_cut_info

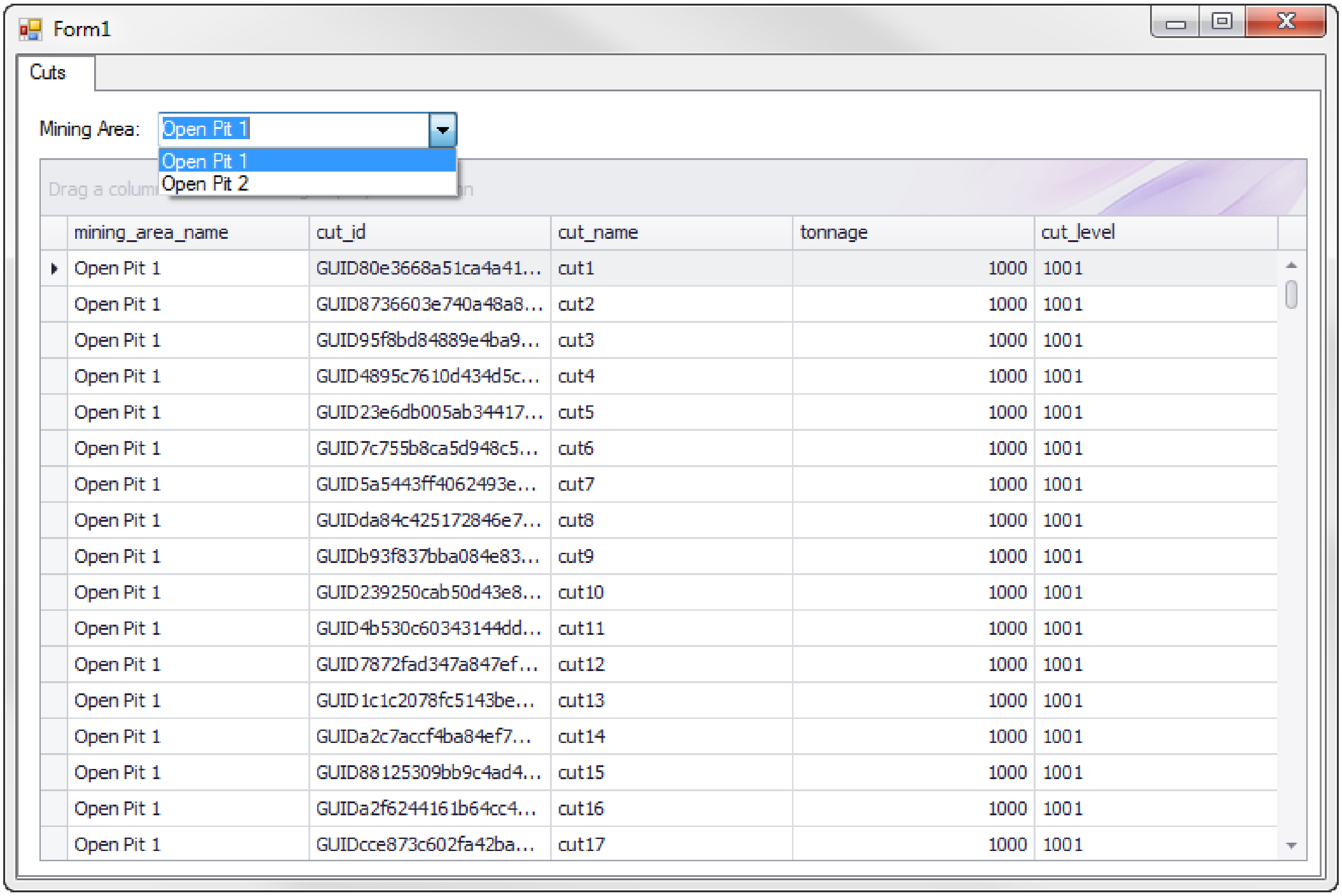


mining_area_name	cut_id	cut_name	tonnage

t_cut_geom



cut_id	X1	Y1	Z1



Mining Area: Open Pit 1

	mining_area_name	cut_id	cut_name	tonnage	cut_level
▸	Open Pit 1	GUID80e3668a51ca4a41...	cut1	1000	1001
	Open Pit 1	GUID8736603e740a48a8...	cut2	1000	1001
	Open Pit 1	GUID95f8bd84889e4ba9...	cut3	1000	1001
	Open Pit 1	GUID4895c7610d434d5c...	cut4	1000	1001
	Open Pit 1	GUID23e6db005ab34417...	cut5	1000	1001
	Open Pit 1	GUID7c755b8ca5d948c5...	cut6	1000	1001
	Open Pit 1	GUID5a5443ff4062493e...	cut7	1000	1001
	Open Pit 1	GUIDda84c425172846e7...	cut8	1000	1001
	Open Pit 1	GUIDb93f837bba084e83...	cut9	1000	1001
	Open Pit 1	GUID239250cab50d43e8...	cut10	1000	1001
	Open Pit 1	GUID4b530c60343144dd...	cut11	1000	1001
	Open Pit 1	GUID7872fad347a847ef...	cut12	1000	1001
	Open Pit 1	GUID1c1c2078fc5143be...	cut13	1000	1001
	Open Pit 1	GUIDa2c7accf4ba84ef7...	cut14	1000	1001
	Open Pit 1	GUID88125309bb9c4ad4...	cut15	1000	1001
	Open Pit 1	GUIDa2f6244161b64cc4...	cut16	1000	1001
	Open Pit 1	GUIDcce873c602fa42ba...	cut17	1000	1001