Exam PG5200 Tools Programming
Duration: 72 hours

Delivery info

- Individual delivery
- The code must compile and run in Visual Studio 2017.
- Documentation must be in doc, docx or pdf format.

Make a card-creator

Implement an editor for collectable playing cards such as Magic the Gathering or Hearthstone. The purpose of this editor is to help game designers effectively mass-produce cards. The game can be an existing game, or a game you design for the occasion. You are free to include as many features as you like. If you have a specific game in mind, which would deviate somewhat from these requirements, you may do this if you explain in detail why and how you have done this, and the complexity is at least as high as these requirements.

Functional requirements, must have:

1. One free-text field for name
2. A drop-down menu for «type» (such as colour in Magic, or minion/spell in Hearthstone)
3. An additional window where you can define a new «type». Types should at a minimum have a name, maximum and minimum stats.
4. At least three numerical stats inputs which limit input to numbers, and limits maximum and minimum value based on type. (Or similar rules)
5. A field that brings up a file selector to select an image for the card. It should verify that the file is an image, and the image should be displayed in the editor.
6. The editor should have a button for storing the card to a database, empty all fields and be ready for next card.
7. The editor should have a browser that lets you look through existing cards and open them.
8. It should be possible to delete cards from the database.
9. Multiple users should be able to add or update cards in parallel without the possibility of conflicts.
10. The editor should have ability to export cards to JSON-files.
11. The editor should be able to import cards from JSON-files.

Functional requirements, nice to have:

1. Ability to search for cards based on names, types and stats
2. Instead of simply showing the image for the card, the editor may show a compete card, with background graphics and the stats in the right place
3. The editors should understand some basic rules for the game. (For example: The editor knos and enforces that for 3 mana the sum of attack and toughness may only be 6.)
4. An importer allowing the user to load the card in Unity3d. The card should then be shown as it would in an actual game.

Documentation requirements, must have:

1. A thorough description of how to set up, compile and run the project.
2. An overview of the software design of the project.
3. A description of problems encountered during development and solutions to these.
4. Part of the solution you are especially satisfied with.
5. Elements you wanted to include but did not get time for.

Technical requirements:

1. Programmed in C#.

2. Use WPF.
3. Use an SQL database and LINQ to SQL
4. Database should be Microsoft SQL server.
5. Use a JSON serialiser.

End of assignment