MMLS R2 – Implementation Evaluation Rubric Section and Team: *team-info*

This is the rubric that will be used for evaluating your Design Project implementation. The instructor will do spot checking of the submitted source code to check these evaluation dimensions.

Dimension	Exceptional Performance 4	Competent Performance 3	Acceptable Performance 2	Developing Performance 1	Beginning Performance 0
Functionality (75%)	Program flawlessly provides all required functionality	Program provides all required functionality with a few small bugs	Program provides most required functionality or has several bugs	Program starts but has little functionality or the functionality is so buggy it is unusable.	Program never starts.
	☐ No lingering problems from R1 seen ☐ Select between on-line MusicBrainz Web Service database OR off-line database ☐ GUI to request and display music information ☐ Select between GUI or command line		☐ Multiple separate, persistent user libraries ☐ Undo/redo of adding/removing songs from library Especially when artist is removed because there are no more songs or releases ☐ Undo/redo of rating a song		
File Header, Method Header and Code Comments (10%)	All header comments are provided, and are short, succinct, and clear descriptions of the class, method, etc., that they describe. All necessary areas are commented. Every comment significant, none is verbose.	All header comments are provided and describe methods, classes, etc., appropriately but some are verbose or confusing. Few comments are missing, unnecessary, obvious, or verbose.	Some header comments are missing, or are incorrect with respect to what a class, method, etc. is responsible for. Several comments missing, unnecessary, obvious, or verbose.	Many missing, incorrect, inappropriate, or misleading header comments. Many comments missing, unnecessary, obvious, or verbose.	No header comments. No method body comments.
Methods (10%)	Clear, cohesive methods with appropriate args and return types. Private methods to reduce complexity and factor out repeated code. No inappropriate choice of statements, expressions and control structures.	Methods have clear purposes and straightforward implementations. Little repeated code. Most choices of statements, expressions, and control structures are appropriate.	Several long methods, or noticeable repetitive code. Several examples of inappropriate statement selection, expressions, or control structures.	Several methods with complex interfaces, compound (incohesive) purposes, or a large amount of repeated code. Examples of poor statement selection, expressions, or control structures.	Many methods with overly complex interfaces, incohesive purposes, complex implementations. Use of unstructured coding techniques.
Indentation and Formatting (5%)	Consistent indentation; judicious use of white space to set off blocks of code.	Consistent indentation. Adequate formatting.	Some inconsistencies in indentation. Some formatting problems.	Gross inconsistencies in indentation; inconsistencies among team members.	No attempt at reasonable indentation or readable formatting