Architectural Analysis

The architecture used for this project was a strict layered architecture. There were a total of three layers: the database layer, the backend layer and the frontend layer. This is how this architecture meets each illitiy.

Attribute	**Importance**	**Why? How?**
Usability	2	Access to the system can only be done for an end-user through the frontend, which might condense the backend's features too much.
Reliability	2	Although there is nothing that makes this inherently unreliable, the system requirements sometimes have to be bent to make it fit this architecture. This means there could be design flaws built in just due to lack of diligence when designing.
Performance	3	This should perform well so long as each layer is well-written.
Supportability	3	Modular design of layers means this would be easy to maintain.
Security	3	In theory, there is nothing inherently insecure about this architecture, assuming that everything is built on the proper technology.
Safety	3	Reasonably reliable and performs well.
Availability	2	More layers means more potential for something to go wrong.
Maintainability	2	Layers should be able to be debugged and upgraded individually as needed.
Scalability	3	Layers should be able to be added as needed and the requirement that layers are not repeated means that more data does not mean more calls.
Portability	3	Layers should not be tied to the previous layer.

Architecture with Group Illities Table

Attribute	Group Priorities	Strict Layered
Usability	3	3
Reliability	3	3
Performance	1	1
Supportability	3	3
Security	3	3
Safety	1	1
Availability	2	2
Maintainability	2	2
Scalability	3	1
Portability	3	2