Specification Document

# System Description

The system, in its entirety, will be a universal remote having the ability to control an infrared (IR) equip devices. It will be able to record IR data sequences (or commands) and store them in a SQL database. The system will also have the ability to create a sequence of commands to accomplish a specific task (Turn everything off, setup living room for a movie, etc…). In the future, I plan to include other protocols, such as Bluetooth, TCPIP, and Zigbee.

For this programming class, I will focus more on the control system and create the framework to hold the rest of the features. The framework will have the ability to interact with hardware as well as the database. This will include the hierarchical structure that includes all interfaces.

The user interface will be graphical, that will have the ability to store delete or send IR codes, create modify execute and destroy sequences. The interface will display stored information in some way that is helpful to the user.

# Test Requirements

Most of the testing will be white box testing because this system will not be fully integrated during the course of this class. This means I will need to rely on the unit testing to unsure there are no defects in the software.

## Framework requirements

* The framework needs to have access to all objects
* The framework needs to have the ability to above store IR codes
* The framework needs to have the ability to delete IR codes from the database.
* The framework needs to have the ability to send IR codes to the hardware to be broadcasted.
* The framework needs to have the ability to create command sequences from stored codes.
* The framework needs to have the ability to modify command sequences.
* The framework needs to have the ability to delete command sequences, but not the IR codes.
* The framework shall inform the user if a delete IR code was included in a command sequence.
* The framework needs to have the ability to relay errors, warnings, and exceptions to the user in a timely fashion.

## GUI Requirements

* The GUI will need to display a log of any errors, warnings, and exceptions during any process
* The GUI will need to have a button to above store IR codes
* The GUI will need to have a button ability to delete IR codes from the database.
* The GUI will need to have a button to send IR codes to the hardware to be broadcasted.
* The GUI will need to have a button to create command sequences from stored codes.
* The GUI will need to have a button to modify command sequences.
* The GUI will need to have a button to delete command sequences, but not the IR codes.
* The GUI will need to inform the user if a delete IR code was included in a command sequence.

# Initial Design Strategy

The initial design will try to keep modularity and extensibility in mind. This is a relatively small piece of software, so speed shouldn’t be an issue. The main problem will be future proofing so when I continue work after this class. This means the hierarchy of my classes will need to be sophisticated enough to add more in the future. Interfaces seems like a good idea.

To make this project realizable, I will need to reduce the complexity a bit. This means the interfaces leading outside of the system will need to be taken out of the current scope. If I have time I can work on them later.