Dummy Project

The dummy project is your major programming project. It is a general project where you get to choose what application to create. You only need to fulfill the requirements. Each component needs to be incorporated into your project with a relevant role. It is up to you which component(s) are more heavily utilized. You can create an application that is light on GUI and heavy on non-GUI and vice versa.

Design

Your dummy project needs to be designed before the bulk of the coding begins. Feel free to do a little prototype coding during the design phase but save the actual coding until after the design is done.

Design Steps

1. Come up with an idea for your project.
2. Come up with features (functionality) for your application.
3. Split your feature set up into 2 sets; required features and optional features.
4. Create a high level diagram of your application. This includes how you want to structure your application (big boxes with arrows) as well as sketches of how you want your GUI/web to look.

After the design is complete, begin full out coding! Be sure to update your design as you are coding.

Repository

Your design and application needs to be in a git repository. Create a repository on the server **lind2802** under the directory **`/repositories/<USERNAME>`** as you've done in the past.

Components

- GUI - Something using Qt
- Non-GUI - Something that is not graphical
- Database - Sqlite is fine or a full fledged DB like MySQL/Postgres is OK too
- Web - Anything from a static client side only webpage to a full stack setup

Component Requirements

GUI

1. Uses the graphics library/framework Qt
2. Must have a QMainWindow
3. Must have at least 2 menus in the QMenuBar
4. Must have at least 1 QDialog or QMessageBox
5. Must be interactive

Non-GUI

1. Must be usable without a GUI. This means, as an example, if the GUI was replaced with a CLI (command line interface) then this component would still work.
2. Must be object oriented (uses classes)

Database

1. Must have at least 1 table
2. Must have a class that represents it

Web

1. Must use HTML
2. Must use CSS
3. May use a web server (CherryPy, Flask, Apache, Tomcat, Glassfish, etc)
4. Must have an interactive component