1 Design

The capstone project will be a site designed around a central theme, which must be a viable business model. A viable business model does not necessarily require the model be profitable. But it does require that the site have a real world application and be viable for immediate deployment. Some example viable business models are:

1. Online store
2. Social networking site for a specific group
3. Site selling a service
4. Non profit organization
5. Government site

Any theme that can be applied to viable business model can be used. The models listed above are examples and are by no means exhaustive.

2 Components

There will be at least four components to the site. Each component will have a backend and frontend sub-component. For instance, a username/password component will have a database backend, a frontend for users to register and edit their own profile, and PHP/JavaScript code that bridges the backend and frontend sub-components. Components include, but are not limited to:

1. Connecting to a Social Media site
2. Using an external API to get data
3. Using/administrating a database to hold internal data
4. Using sessions
5. Creating a username login/logout subsystem

3 Technologies

The following is an example technology list for the project. This list is not exhaustive, and the chosen theme will drive what technologies are used. That is, the needs of the site will give rise to the technology used rather than the technologies used giving rise to the site.

1. HTML5
    - No use of deprecated tags
    - Grid layout

- Includes audio/video
- Includes `<canvas>` element
- Includes a form

2. CSS

    - All styling done in CSS
    - Appropriate use of selectors
    - Uses animation
    - Consistent use of fonts
    - Adaptive site: one look for mobile, one look for desktop using the same HTML

3. PHP

    - Includes/requires library files
    - Uses external API using PHP's JSON/SOAP APIs[1]
    - Uses functions
    - Uses arrays with loops
    - Fully unit tested with `SimpleTest`
    - Uses exception handling
    - Uses object oriented classes
    - Uses-Model-View Controller (MVC) design pattern
    - Uses Data Access Object (DAO) design pattern

4. JavaScript

    - Links to jQuery or other libraries
    - Uses JavaScript functions
    - Uses arrays with loops
    - Uses conditionals

5. mySQL

    - Uses all major relations (1-to-1, 1-to-n, and n-to-m)
    - Full Entity Relationship Diagram (ERD)
    - Uses basic mySQL queries (`INSERT`, `SELECT`, `DELETE`, `UPDATE`)

[1] JSON preferred, unless SOAP is the only available method.