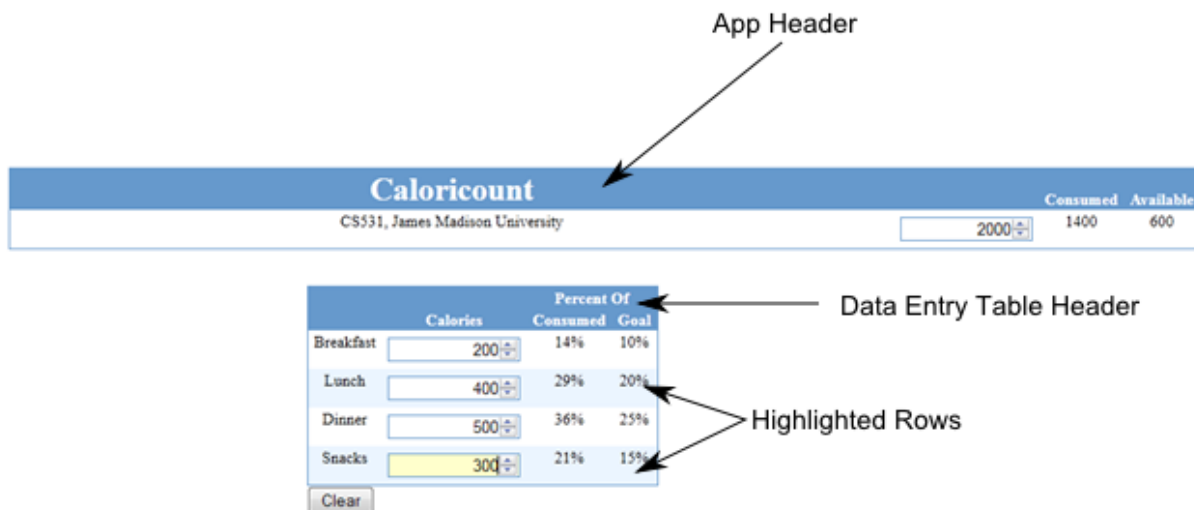# Requirements for Caloricount v2

Caloricount is a simple program that allows a user to calculate the way she/he allocates calories across meals and snacks. Version 2 must satisfy the following requirements (in addition to the requirements for version 1).

1. User Interaction Requirements – Appearance/Presentation of the GUI
    1.1. The product must support different color themes.
        1.1.1. Each theme must include the background color for the app header.
        1.1.2. Each theme must include the background color for the header in the data entry table.
        1.1.3. Each theme must include the background color for the highlighted rows in the data entry table.

    1.2. The product must be brandable.
        1.2.1. The name of the product (in the App Header) must be specifiable by a third-party.
        1.2.2. The text under the product name must be specifiable by a third-party.
    1.3. The product must be internationalized/localized.
        1.3.1. The word "Consumed" must be in the user's language.
        1.3.2. The word "Available" must be in the user's language.
        1.3.3. The word "Calories" must be in the user's language.
        1.3.4. The word "Goal" must be in the user's language.
        1.3.5. The phrase "Percent Of" must be in the user's language.
        1.3.6. The word "Breakfast" must be in the user's language.
        1.3.7. The word "Lunch" must be in the user's language.
        1.3.8. The word "Dinner" must be in the user's language.
        1.3.9. The word "Snacks" must be in the user's language.

KitchIntel

2. Theme Requirements
    2.1. The product must have at least two different themes.
        2.1.1. The `ocean` theme must be identical to the theme that was implicitly used in v1.
        2.1.2. The `jmu` theme must use purple and gold for the various background colors.
3. Theme Selection Requirements
    3.1. The theme must be selectable using the query string of the URL.
        3.1.1. The `name` of the pair must be `theme`.
        3.1.2. In the case of an invalid/unknown/unspecified theme, the `ocean` theme must be used.
4. Branding Requirements
    4.1. The brand of the product (i.e., "Caloricount" in v1) must be specified using the query string of the URL.
        4.1.1. The `name` of the pair must be `brand`.
        4.1.2. In the case of an unspecified `brand`, "Caloricount" must be used.
    4.2. The slogan (i.e., the text under the brand) must be specified using the query string of the URL.
        4.2.1. The `name` of the pair must be `slogan`.
        4.2.2. In the case of an unspecified slogan, "CS531, James Madison University" must be used.
5. Language Requirements
    5.1. The product must have at least two languages.
        5.1.1. One of the languages must be English.
6. Language Selection Requirements
    6.1. The language must be selectable using the query string of the URL.
        6.1.1. The `name` of the pair must be `language`.
        6.1.2. In the event of an invalid/unknown/unspecified language, English must be used.
7. Non-Functional Requirements
    7.1. The product must not use an existing localization/internationalization library (i.e., all of the code needed for localization/internationalization must be written in house).
        7.1.1. The design of the in-house localization/internationalization subsystem may be influenced by existing localization/internationalization libraries.