1 Introduction and Use Cases

Your client, Hypothetical Meals, is a large food company that produces a significant portion of the world’s food. They currently use a mishmash of spreadsheets and macros to manage food manufacturing scheduling. They would like a unified system to replace these highly manual procedures. Further, they would like to gain a better understanding of the costs and revenues involved to improve business decision-making. This system will serve the following use cases:

- The system will track SKUs (Stock-Keeping Units) for the food items produced by the company. These details are provided by the administrator.

- The system will track the ingredients that are used to manufacture each SKU.

- Manufacturing staff will use the system to calculate ingredient needs based on manufacturing goals.

- Product managers will use the system to produce reports showing information about ingredients and the SKUs they are used to make.

- Users will be able to bulk-import and bulk-export ingredients and SKU information from a simple text format.
2 Definitions

- **Ingredient**: A food product purchased by the company for use in manufacturing. Ingredients come in packages of varying size (e.g. a 55 gallon drum, 20 lb. sack, etc.).

- **SKU**: Stock-Keeping Unit. Refers to a product made by the company. The term “SKU” may also be used to refer to the SKU number, commonly written as SKU#.

- **Customer**: A food retailer or distributor, such as a grocery store, convenience store, food service company, etc.

- **Consumer**: The individuals who purchase SKU units from our customers.

- **Product line**: A group of related SKUs, such as “rich and hearty soup”. Every SKU belongs to a product line.

- **Unit**: An individually purchasable food item (e.g. one can of soup). Units are what are sold to consumers.

- **Case**: A carton or box of units (e.g. a box of 24 cans of soup). Cases are what get sold to customers.

- **UPC**: Universal Product Code. A standard for assigning numbers to items purchased or traded in the worldwide economy. Hypothetical Meals products use the UPC-A standard.

- **Manufacturing goal**: A list of SKUs and case quantities that the company may want to manufacture.

- **Input assistance**: A user input is said to be **assisted** if it is a user-selected reference to an existing record (ingredient, SKU, etc.) where the UI provides a listing, inline search, autocomplete, and/or other means to allow easy and efficient selection.

3 Requirements

_A note on requirements: No set of requirements is perfect, and that is certainly true here. I'm sure that contradictions, under-specified behavior, and unintended consequences will be revealed. Your overriding goal should be to produce a quality system; if you believe that goal would be better served if a requirement were altered or interpreted a certain way, ask about it, and get the conclusion in writing. The result may be a variance in a requirement for a specific team, or even modification of this requirements document for all teams. In short, if unsure, ask._

Some requirements have attached an informal tip, clarification, or example – these do not alter the requirements themselves, but are meant to answer likely questions about a requirement.

1. **Server**

   1.1. Your software must have a server that supports an arbitrary number of users.

   1.2. During the install/setup process, a special user named “admin” is configured.

   1.3. Users must have their accounts created by the admin user before being able to use the system.
1.4. Any stored passwords must be kept in a secure manner (i.e., salted + hashed at minimum)

1.5. All communication between the clients and server must be encrypted.
   
   Tip: For web-based solutions, this means using HTTPS.

1.6. The server must maintain state in a persistent fashion.

1.7. For all views which show a potentially unbounded number of records, the response time of the interface shall not depend on the quantity of records unless a full listing is explicitly requested by the user.

   Tip: This implies some form of pagination so that only a finite number of records are retrieved at a time. Pagination can be explicit (page 1 of N) or implicit (infinite scrolling). The latter part of the requirement (“unless a full listing is requested”) implies a “show all” button or similar. Other UI solutions are likely also possible.

2. SKU Data Management

2.1. The administrator shall be able to add, modify, review, and remove ingredients within the system.

2.1.1. An ingredient consists of:
   
   2.1.1.1. Ingredient name (required, unique)
   2.1.1.2. Ingredient number (required, unique, auto-generated unless supplied by user)
   2.1.1.3. Vendor information (optional free-form multi-line text field)
   2.1.1.4. Package size (required, free-form short text field describing how much comes in a single package of the ingredient, e.g. “60lb”, “55 gallon drum”, etc.)
   2.1.1.5. Cost per package in USD (required – represents the cost of the ingredient per “package size” as defined above)
   2.1.1.6. Comment (optional free-form multi-line text field)

2.1.2. Users may review ingredients with the following view options.
   
   2.1.2.1. The view should be filterable by any combination of:
       - Keyword search.
       - SKU(s) produced with the ingredient. SKU selection shall be assisted.
   2.1.2.2. The interface should show the number of SKUs that use a given an ingredient and provide some kind of detail view allowing the user to review a list of the SKUs that use an ingredient.
   2.1.2.3. The view should be sortable by all shown fields.

2.2. The administrator shall be able to add, modify, review, and remove product lines within the system to help organize SKUs.

2.2.1. A product line consists of a name (required, unique).

2.2.2. All SKUs are a member of exactly one product line.

2.3. The administrator shall be able to add, modify, review, and remove SKUs within the system.

2.3.1. A SKU consists of:
2.3.1.1. Name (required, max 32 characters)

Note: When A SKU is shown elsewhere in the system, its display name should be shown as “<NAME>: <SIZE_PER_UNIT> * <CASE-COUNT>”, e.g. “Tomato soup: 28oz * 128”.

2.3.1.2. SKU# (required, unique, numeric, auto-generated unless supplied by user)

2.3.1.3. Case UPC# (required, conforming to the UPC-A standard (12-digit) for consumer products (starts with 0-1 or 6-9) with a valid check digit, unique)

2.3.1.4. Unit UPC# (required, confirming to same UPC standards as Case UPC#, not necessarily unique (as two SKUs could be for different case-sizes of the same consumer item))

2.3.1.5. Unit size (required, free-form short text field describing how much comes in a single package of the item, e.g. “28oz.”, “1qt.”, etc.)

2.3.1.6. Count per case (required, the integer number of units in one case)

2.3.1.7. Product line (required, a choice of exactly one product line to which this SKU belongs)

2.3.1.8. Zero or more \{ingredient,quantity\} tuples, where ingredient refers to an existing ingredient record (with assisted selection) and quantity is a decimal fraction of a package used.

2.3.1.9. Comment (optional free-form multi-line text field)

2.3.2. Users may review SKUs with the following view options.

2.3.2.1. The view should be filterable by any combination of:

- Keyword search.
- Ingredient(s) used. Ingredient selection shall be assisted.
- Selected product line(s). Product line selection shall be assisted.

2.3.2.2. SKUs may be sorted by any shown field.

2.3.2.3. SKUs may be optionally grouped by product line.

3. Bulk import/export facility

3.1. The administrator shall be able to import new ingredients, SKUs, and product lines into the system by means of an import compatible with modern spreadsheet software (CSV, XLSX, or similar). The customer is accepting proposals on the format.

3.2. The import interface shall include documentation as to the import format.

3.3. The import action shall only occur if the entire input is free of name conflicts or otherwise problematic issues; if such issues arise, the precise nature of the error should be presented to the administrator in enough detail that it can be corrected.

3.4. If an import contains identical record(s) to those already in the system, such records should be ignored.

3.5. If an import contains record(s) that match on name or a unique numeric identifier, the user should be warned about all such records in detail, and if the user approves, the records should be modified to match the imported data.

3.6. After a successful import, a count and list of records that were added, updated, and ignored should be provided.
3.7. The system shall be able to export any of the above data in a format compatible with import. The specific records exported should be filterable by the same means defined in the “view options” described for each record in the requirements above (reqs 2.1.2.1, 2.3.2.1).

Note: This allows for an export/modify/import workflow when large-scale changes are needed.

4. Manufacturing

4.1. Users shall be able to input a manufacturing goal as follows:

4.1.1. Users shall be able to input a set of SKUs, each with a desired case quantity.

4.1.2. The selection of SKUs shall be assisted, further, selection will be aided by the ability to navigate using search and filtering by product line.

4.1.3. Users shall be able to save such manufacturing goals by name for future loading and reference. Manufacturing goals are private to each user.

4.1.4. Users shall be able to export a manufacturing goal to CSV format.

4.2. Users shall be able to access a manufacturing calculator:

4.2.1. Users shall be able to select a saved manufacturing goal and produce a list showing all ingredients required and their quantities as measured in decimal packages.

Clarification: this means if a single case of soup uses 0.55 packages of flour and we’re making 10 cases, that’s 10 * 0.55 = 5.5 packages of flour. Note that the system does not care about the package size of ingredients (e.g. if it’s 1 ton of flour per package or 20 lbs.), only the fraction of ingredient package consumed.

4.2.2. Users shall be able to export the calculation result in CSV format.

4.2.3. Users shall be able to produce a well-formatted printed document suitable for sending to a manufacturing plant. This can be achieved either with native “print” capability (e.g. within a web browser) or by an export-to-PDF feature.

5. Reporting

5.1. Ingredient dependency report: Users shall be able to create a tabular report showing a set of ingredients (the selection of which follows the same rules as the “view options” described in req 2.1.2). For each ingredient, all SKUs made with the ingredient shall be shown. This report should be viewed or exported in CSV format.

6. Documentation

6.1. Developer guide: A document shall be provided which orients a new developer to how your system is constructed at a high level, what technologies are in use, how to configure a development/build environment, and how the database schema (or equivalent) is laid out.

6.2. Deployment guide: A document shall be provided which describes how to install your software entirely from scratch. It should start by describing the platform prerequisites (e.g. Linux distro, required packages, etc.), then mechanically describe every step to deploying your system to production readiness.