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 uses a mishmash of spreadsheets and macros to manage food production, including inventory, logistics, sales, etc. They would like a unified system to replace these highly manual procedures. This system will serve the following use cases:

- The system will track ingredients, including vendors which can provide a given ingredient, cost information, and storage information (frozen, refrigerated, etc.). These details are provided by the administrator.

- The system will allow authorized users to document orders for ingredients and to monitor how storage of ingredient stock compares to on-site storage capacity.

- The system will allow the administrator to note when ingredients are consumed by production using the popular “shopping cart” metaphor.

- The system will provide reports on purchasing and ingredient use.

- Users will be able to bulk-import new ingredients from a simple text format.

2 Definitions

- **Ingredient**: A food product purchased by the company for use in production. Available from at least one vendor, typically more. A given ingredient will come in a specific package and temperature state.
• **Ingredient package**: Ingredients may come in a sack or pail (around 50 lbs), a drum (for liquids, around 500 lbs), a “supersack” (a large square sack that sits on a shipping pallet; around 2,000 lbs), or a truckload or railcar (for liquids, 50,000+ lbs).

• **Temperature state**: The kind of storage a given ingredient requires to maintain food safety and/or freshness: either frozen, refrigerated, or room temperature. Ingredients should never be stored at a different temperature state than they come in.

• **Vendor**: A partner company that sells one or more ingredients.

• **Storage capacity**: The total amount the company can store for a given temperature state. Storage capacity is measured in pounds.

• **Ingredient stock**: The total amount of a given ingredient owned and stored for use by the company. Measured in pounds.

• **Production run**: The act of consuming one or more ingredients to produce a product.

## 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)
   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.
      
      *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). Other UI solutions are likely also possible.*
2. Vendor information management

2.1. The administrator will be able to add, edit, or remove vendors. A vendor will be defined by a unique name, a free-form field for contact information, and a unique case-insensitive alphanumeric freight carrier code.

3. Inventory tracking functionality

3.1. The administrator will be able to add or edit ingredients. An ingredient is defined as:

- A unique name
- The package the ingredient comes in, which is one of:
  - Sack (50 lbs)
  - Pail (50 lbs)
  - Drum (500 lbs)
  - Supersack (2000 lbs)
  - Truckload (50000 lbs)
  - Railcar (280000 lbs)
- The temperature state of the ingredient: frozen, refrigerated, or room temperature
- The vendor(s) which can provide the ingredient (see req 2) and the price each charges per package

3.2. The administrator will be able to remove ingredients.

3.3. The administrator will be able to set the storage capacity for ingredients in each of the three temperature zones: freezer, refrigerator, and warehouse (room temperature), except for truckload/railcar ingredients, for which there is no tracked storage limit.

**Clarification:** Truckload/railcar ingredients are either stored in special-purpose tanks or left in their original vessel until use, so maximum capacity need not be tracked by this software.

3.3.1. Storage capacity per temperature state (freezer/refrigerator/warehouse) will be tracked in pounds.

3.3.2. The system must enforce storage limitations when processing orders (see req 3.4).

3.3.3. Temperature states must always be respected; an ingredient may never be stored at another temperature state.

3.3.4. If the user tries to edit a capacity to a value below the current total inventory storage in that temperature state, an appropriate error will be shown and the operation will be prevented.

3.4. Users shall be able to log orders of ingredients from vendors.

3.4.1. The user navigates to a given ingredient, identifies a vendor, and requests an order for a given integer number of packages.

3.4.2. Orders which would exceed storage capacity must be denied with an appropriate error message.

3.4.3. Successful orders shall immediately log the ingredient into the inventory.

3.4.4. The system shall track money spent on ingredients for use in reporting (see req 4).

3.5. Users shall be able to browse the inventory.
3.5.1. All ingredients shall be summarized in an efficient manner.
3.5.2. The user should be able to search by ingredient name and filter based on temperature state and/or package.
3.5.3. The quantity of each ingredient will be tracked in pounds.
3.5.4. The user shall be able to view a detail page for any given ingredient showing all tracked information for it.
3.5.5. The administrator shall be able to directly edit ingredient stock quantity to correct errors.

3.6. Users shall be able to note the consumption of ingredients as follows.
3.6.1. The user can add/remove ingredients to a “cart” with quantities specified in pounds.
3.6.2. The user can “check out” the cart, thus indicating that the chosen ingredients are removed from inventory and used in production.
3.6.3. The inventory totals for the selected ingredients should be decremented appropriately.
3.6.4. Attempts to add more to cart than inventory stock allows should be disallowed, as the production must not exceed inventory stock. The inventory cannot go negative.

4. Reporting

4.1. **Spending report**: Users shall be able to view a report of ingredient spending which indicates total spending on each ingredient to date, both overall and only for ingredients used in production.

5. Documentation

5.1. **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.
5.2. **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.

6. Bulk import facility

6.1. Administrators shall be able to import new ingredients into the system by means of a text-based import (CSV, JSON, or other simple plaintext format). **The customer is accepting proposals on the format.**
6.2. The import interface shall include documentation as to the import format.
6.3. 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.