# Specification Document

## Data that Management Team Receives from Kitchen Team

To ensure that we (the management team) can perform all of our tasks, we require the kitchen team to send us the following data.

An individual table implies a logical grouping of data. Each entry in the table has the proposed data name, the proposed data type and the details of the data. Any information required to interpret the table has been specified directly before the table. In addition, the source (the document or lecture recording from which we have identified this requirement) and the source details (any relevant notes from the source) have been specified directly after the table.

The management team requires that the kitchen team send us a menu every week. The menu will be an object of class Menu. The menu object must contain a date (Object) field and a courseSet(HashSet<Course>) field. In order for us to use the menu object as intended, we require the menu object to conform to the requirements detailed below. However, when we request data, we will only request the menu object itself, not the individual instances of data it contains.

| Data Name | Data Type | Data Details |
|---|---|---|
| **menu** | **Object** | Each menu is an object of class Menu.<br>The menu object must contain a date (Object) field and a courseSet (HashSet<Course>) field. |
| courseSet | HashSet<Course> | A HashSet of Course objects (no duplicate values). |
| course | Object | Each course is an object of type Course. Each Course object must have a courseID (int) field and a dishSet (HashSet<Dish>) field. |
| courseID | int | A unique identifier for a course. This will be the primary key for a dish. |
| dishSet | HashSet<Dish> | A HashSet of Dish objects (no duplicate values). |
| date | Object | Each date is an object of class LocalDate (from the java.time package). Each LocalDate object uses the YYYY-MM-DD format.<br>Each menu is used for a single week. Date will specify the date of the first day of the week (Monday) that the menu will be used for. Date will be the primary key for a menu (two menus cannot have the same data, i.e. two menus cannot be used in the same week). |
| dish | Object | Each dish is an object of class Dish. Each Dish object must have a dishID (int) field, name (String) field and an ingredientsList (HashMap<Ingredient, Integer>) field. |

| | | |
|---|---|---|
| dishID | int | A unique identifier for a dish. This will be the primary key for a dish. |
| name | String | The name of a given dish. |
| ingredientsList | HashMap<Ingredient, Integer> | A HashMap with an Ingredient object as the key and the corresponding quantity as the value. |
| ingredient | Object | Each ingredient is an object of class Ingredient. Each Ingredient object must have a name (String) field stockLevel (int) field and a lowStockThreshold (int) field (please see next table for further details). |
| ingredientID | int | A unique identifier for an ingredient. |
| name | String | The name of a given ingredient. |
| stockLevel | int | The current stock level (i.e. quantity available for use) for a given ingredient. |
| lowStockThreshold | int | When the stockLevel is less than this number, a stock level update is sent to the management team. |

Source(s): Case Study -> Kitchen -> Menu Compilation, Lecture Recording (Feb 12 2024 3PM) -> 32:35 - 37:00

Source details: The case study document states that the management team is sent a menu from the kitchen team every week. The menu for a given week must be sent three weeks in advance of that week. The manager(s) from the management team will review and update the menu. The lecture recording (12/02/24) states that the management team is only required to know about the dish name and the list of ingredients that corresponds to that dish.

The management team requires that the kitchen team send us a stock level update (for one or more ingredients) upon the management team's request. Alternatively, the kitchen team will send us a stock level update for an ingredient(s) when the stock level for that ingredient(s) has reached 0 or has reached a number below a specified "lowStockThreshold".

| Data Name | Data Type | Data Details |
|---|---|---|
| **stockLevel** | **int** | Upon the management team's request for a stock level update on a single ingredient, the current stockLevel for that ingredient is provided. |
| **ingredientID** | **int** | Alternatively, the kitchen team sends us a stock level update.<br>ingredientID is used as the primary key for an ingredient. Therefore, the provision of ingredientID allows us to identify which ingredient needs to be re-ordered. |

Source(s): Case Study -> Kitchen -> Stock and Waste Tracking

Source details: The case study document states that the kitchen team keeps track of the current stock of ingredients. Although there is no explicit mention of this data being sent to the management team, we believe that the management team should be made aware of current stock levels so as to facilitate re-ordering of ingredients from the supplier as and when necessary. In addition, the menu can be updated so that dishes that are rendered unavailable (due to depleted stock of relevant ingredients) are omitted. This streamlines the business's daily operations by removing the need for the front-of-house team to print incorrect menus and verbally communicate the unavailability of certain dishes to diners.