

## INFS2044 Assignment 1 Case Study: Cooking Robot

In this assignment you will be designing a system for a cooking robot (CRS) deployed in commercial kitchens. The system will be able to load recipes and control the sensors and actuators installed in the kitchen to execute the recipe. [This video](#) illustrates how such systems may operate.

*There may be commercially or open-source systems available that provide much or all the functionality described in this document. For purposes of the assignment, it is not acceptable to use these systems as part of your design.*

The system comprises of robotic stations that are installed in the kitchen. Examples of stations include:

- Ingredients dispenser (may be separate for each kind of ingredient, such as vegetables, cheese, sauce)
- Cooking station (may be multiple stations with same or different capabilities)
- Serving station

### Use Cases

Some of the Use Cases for this system are outlined below.

- UC1 Select Meal: User selects a meal among the recipes available in the system, and system creates a ticket for the meal in its queue. (Assume a separate ticket will be created for each meal.)
- UC2 Plan Cooking Process: System removes a ticket from its queue, identifies the recipe on the ticket, and creates a Cooking Plan for executing the activities defined in a recipe, taking into consideration the ingredients and available of stations in the kitchen. (That is, System assigns actual stations and time points to the template plan defined in the recipe.)
- UC3 Execute Cooking Process: System executes the activities defined in the cooking plan, resulting in a cooked meal.
- UC4 Serve Meal: System conveys the cooked meal to a human Chef for final preparation and quality control.
- UC5 Configure System (Epic): User enters the available stations, their capabilities, and any constraints about concurrent use of stations into the system. This information is used by the system when creating a Cooking Plan. This use case may be broken down further into more detailed use cases.
- UC6 Enter Recipe (Epic): User enters a recipe for a meal, including the ingredients, cooking instructions, and cooking times into the system. This use case may be broken down further into more detailed use cases.

## Other Requirements

### Performance:

- PR1: The system shall take no more than 10 seconds plus the nominated cooking time of the recipe for preparing each meal.

### Scalability:

- SR1: The system shall be able to scale to no less than 50 stations.
- SR2: The system shall be able to scale to no less than 500 recipes.
- SR3: The system shall be able to execute no less than 30 cooking plans concurrently (provided that stations are available to accommodate as many concurrent meals)

### Interoperability:

- IR1: The system shall be able to interface with stations from no less than 5 vendors (each may have a different programming interface)

### Configurability:

- CR1: The system shall be configurable, without requiring code changes, for different kitchens, including different number and type of stations. (Assume that the system already knows how to interact with each type of station via the station vendor's programming interface.)

### Auditability:

- AR1: All station's activities shall be recorded (and time-stamped) for system maintenance and fault localisation

## Scope

### In Scope:

- Component design for CRS

### Out of Scope:

- User interface design
- Database schema design
- Security mechanisms
- Network design
- Hardware design
- Costs/pricing
- Detailed component interface and implementation design

## Completeness and Assumptions

The use cases and requirements are most certainly incomplete. You may need to make additional assumptions to elicit volatilities and create your design. Ensure that you discuss relevant assumptions in your assignment submission document.

## Level of Detail

The level of detail expected for this assignment is aligned with that given in the Week 2 Workshop materials, supported by written discussion/justification.