26/08/2020

Tea-room covid-19: Requirement Analysis

# Setting

## A **teaRoom** rectangular room composed of:

## **N teaTable**: tables placed inside the tearoom, where the admitted **Client** can consume his tea.

## **serviceArea**, composed of:

### **serviceDesk**: where the entity barman prepares the tea after receiving a request by the waiter;

### **home**: where the waiter can rest if it has no tasks to do.

## **hall**: where, when he arrives, a **Client** has to wait here before entering the teaRoom (client behaviour explained in point 2). It is equipped with:

### **a presenceDetector,** which can detect the presence of a person or other entity;

### **a smartbell,** which measures the temperature of the Client that wants to enter the tearoom and, if the Client’s temperature is less than 37.5˚, ***sends a request message*** *to the waiter*.

#### **clientIdentifier:** value to univocally represent a client’s request of entering the tearoom. It is assigned by the smartbell to the client and it is given to the waiter with the aforemetined request.

### an **entranceDoor**, through which the Client will be admitted inside the tearoom;

### an **exitDoor**, through which the Client can leave the tearoom.

## A **tearoom is considered safe** if there are no people inside with a temperature greater or equal than 37.5˚ and if there are clean tea-tables posed at a proper distance.

# Client

## **notifiy interest in entering the tearoom**: once in the **hall**, the Client ***has to send a notification*** to the **smartbell**, whose behavior has been described in point [[1.A.iii.2](#_a_smartbell,_which)].

## ***maxWaitTime***: time value that will be given by the **waiter** if there are no free and clean tables available and he’s not been sent away because of the temperature, after which either he has entered, or he has to leave.

## ***maxStayTime***: maximum time that the Client can spend at a teaTable. After it expires, the client has to leave, no matter if he’s finished the tea or not.

# Waiter

## The **waiter tasks**, listed in the requirements, are a set of actions the waiter should be able to perform one at the time, *optimizing as much as possible the execution so that the waiting time of the requests coming from each client is minimized*.

## **Convoy the Client**: once a Client is free to enter the tearoom or ready to leave it, the waiter has to accompany them to and from the table, from and to the entranceDoor and exitDoor, respectively.

## **Clean the table**:

# Barman

## **receive order from waiter**: the orders are transmitted through a wifi device by the waiter to the barman.

## **notify waiter of drink ready**: the barman has to send a notification to the waiter once the drink order by a Client is ready.

# Entites as Actors

# Exchange of Messages