ANALISI REQUISITI

1. clarify the meaning of the *names* and of the *verbs* included in the requirement text given by the customer.
2. define a first working prototype (and a first set of functional TestPlan) as a formal model of the requirements.

**Manager:** Entity that wishes to check the current state of the tea room. To do so he wants to access a browser connected to a web server where he can check the state of each table, the waiter, barman and hall.

**Tearoom**: has been clearly described in the requirements.

**Teatable**: Place where the client will be assigned a seat so that he can enjoy tea at a safe distance from other clients. It can be in 3 states tableClean, tableDirty, tableOccupied. In the requirements the tableClean is specified. It is assumed that the other two states exist to fully describe the possible states of the tables.

**ServiceArea**: Area of the tearoom containing the home area and serviceDesk.

**ServiceDesk**: Area of the tearoom where the Barman *works* which consists in preparing tea for clients when he receives a request from the waiter.

**Hall:** intermediate area that connects the tearoom and the outside world. Its structure is clearly defined in the requirements. To implement the optional requirement only one client at a time is admitted to the hall so the waiter may open the exitDoor to let the client leave the tearoom only when there is not a client already waiting at the entranceDoor.

**Client notifies his interest in entering the tearoom**: once in the hall, before entering the tearoom, the client will send a notification to the smartbell who’s behavior is clearly described in the requirements.

**Client sits at a free tea table**: Client sits at a table that is in the state tableClean. The state of the table will then become tableOccupied.

**Client orders some tea**: send a request to the waiter for some tea. The waiter will then send the request to the barman.

**Client consumes the tea**: The client stays at the table until the tea is finished which will happen within the limit maxStayTime.

**Client pays the service with a credit card**: the client sends a request to the waiter when the tea is finished, receives the bill from the waiter, pays with his credit card.

**Client leaves the tearoom**: once the client has paid he leaves the teaRoom if there is nobody in the hall (optional requirement), changing the state of the table to tableDirty

**Waiter tasks:** a set of activities that the waiter can perform. When he is performing one of these actions he enters a Busy state and cannot start another action until the previous one is finished.The list of tasks is described in the requirements. The waiter will choose what task to do in a way as to reduce as much as possible the waiting time of the requests coming from each client.

**Waiter informs the client of the maximum waiting time**: the waiter sends a message to the client telling him the maximum waiting time if there is no clean table. When the client reaches the maximum waiting time without having entered the tea room he will have to leave the hall.

**Waiter convoys the client**: The waiter walks in front of the client to show him the way from the door to the table or from the table to the door.

**Waiter rests at the home**: the waiter remains stationary at the home area in the service area when there are no tasks to execute.

**Model**

The entities of the system are Waiter, Barman, smartBell and the Clients. Each of them will have a state associated to them that describes what they are doing in each moment which will influence how they interact with each other. For example: a waiter that is busy cleaning a table may not serve a drink prepared by the barman. If the waiter was idle in the home he would immediately take the drink from the barman and bring it to the client when he receives the request.

The states of the actors care:

* Waiter can be busy or free
* Barman can be busy or free
* Client can be Outside, waitingInTheHall, seated, finished, payed and left
* SmartBell can be busy or free

The manager must also be able to verify the state of the tables which can be tableClean, tableDirty and tableOccupied

The various entities of the system will be coordinated by exchanging messages of various kinds (dispatches, request/reply and events). These messages will let us model various activities such as informing the client of the maximum waiting time, taking the order of the client and transmitting it to the barman.

We will model the entities as Actors and we will use the meta-model Qak to minimize the abstraction Gap.

**Test plan**

We plan to test various activities:

* The smartBell must let in the hall only clients that have a temperature below 37.5° Celsius.
* When a client enters the tea room he may only sit at a tables that is both clean and free. If there is no such table available then the waiter must inform the client about the maximum waiting time.
* The client must leave when the maximum waiting time is over
* The client must receive the drink he ordered.
* The client must pay and leave when the maximum stay time is over.

Per minimizzare l’abstraction gap utilizzeremo il metamodelo Qak