

Les notes d'aquest examen i totals del curs (provisionals) es publicaran dimarts 3 de juliol de 2018.

Le **revisions** es faran dijous 5 de juliol d'11:00 a 12:30 al despatx Omega-133.

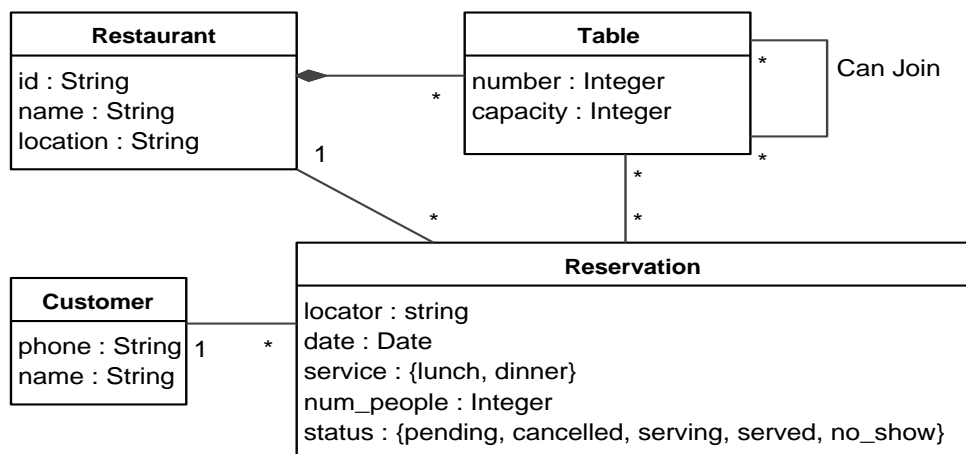
Pregunta 1 (1,5 punts): Expliqueu amb ajuda de diagrames de seqüència les dues variants del patró de disseny *Request/Acknowledge*.

Pregunta 2 (1,5 punts): Responen **una sola** de les preguntes següents, i que correspongui a un tema **no presentat per vosaltres**:

- Expliqueu quins tipus de proves de rendiment hi ha.
- Expliqueu les 3 diferències principals entre les SPA i les MPA.

Exercici (7 punts)

Considereu un sistema per reservar taules a restaurants. Supposeu que els possibles clients (programes) dels serveis que dissenyareu ho faran des d'una intranet segura i que, per tant, no cal considerar el tema de l'autenticació/autorització. El diagrama de classes que en descriu el domini és el següent:



Identificadors: (Restaurant, id), (Table, restaurant.id+number), (Reservation, locator), (Customer, phone)

Caldrà dissenyar els serveis que permetin implementar les 5 operacions següents:

- Consultar la info (tots els atributs) d'un *Restaurant* concret. Caldrà incloure també els atributs *number* i *capacity* de totes les seves taules (*tables*).
- Consultar totes les reserves (*reservations*), amb la possibilitat de filtrar-les per data, *Restaurant* (id), *Customer* (phone), servei (*lunch* | *dinner*) i/o

status (*pending*, ...). Per cada reserva, cal retornar tots els seus atributs així com els del *Customer*, *Restaurant* i taules corresponents.

- Modificar l'estatus d'una reserva. Cal que el canvi sigui coherent (p.ex. no es pot passar de *cancelled* a *served*).
- Eliminar una determinada taula (*table*) d'un restaurant. Cal que aquesta no estigui associada amb reserves *pending* o *serving*.
- Fer una reserva nova, indicant tota la info necessària, incloent-hi la de les taules que s'usaran. Cal fer totes les comprovacions pertinents (per ex., que les taules estiguin juntes i disponibles, que hi càpiguen totes les persones, etc.). El *locator* de la reserva el genera el sistema.

Més concretament,

- [2 punts]** Dissenyeu un servei web SOAP que inclogui les **4 primeres** operacions. Heu d'utilitzar els patrons RPC API i DTO. Per a cada operació cal definir els paràmetres d'entrada i el resultat. Utilitzeu el mode d'interacció *Request/Response*.
- [1 punt]** Dissenyeu un servei web SOAP utilitzant el mode d'interacció *Request/Acknowledge/Poll* per a l'operació de **fer una reserva nova**. Heu d'utilitzar els patrons RPC API i DTO.
- [3 punts]** Dissenyeu un servei web REST per a les **5 operacions**. Per descriure cada operació caldrà omplir la plantilla que teniu a continuació:

Title	Operation name	
URI	Example: /users or /users/:id	
Query Params	q=[String]: required opcional	
Method	GET POST DELETE PUT	
Body Data	(Only for POST and UPDATE requests) Example; {"name": String, "email": ArrayOf(String)}	
Returns	Success Response	Code and Content. Example: 200 OK { "id": String, "name": String, "email": ArrayOf(String) }
	Error Responses	Code and Content. Example: 404 Not Found { message: "no user exists with that id" }

- [1 punt]** Definiu l'escenari que utilitzi el servei web REST anterior i que permeti cancel·lar totes les reserves pendents del *Customer* amb *phone* = 965414605.

a) Servei web SOAP per a les 4 primeres operacions

Operation name	Input	Output
getRestaurantByld	getRestaurantByldReq= <id: String>	getRestaurantByldRes= <id: String, name: String, location: String, tables: InfoTable [*]> where InfoTable = <number: Integer, capacity: Integer>
getReservations	getReservationsReq= <date: Date [0..1], restaurant_id: String [0..1], customer_phone: String [0..1], service: {lunch, dinner} [0..1], status: {pending, ...} [0..1]>	getReservationsRes= Set(InfoReservation) where InfoReservation= <locator: String, ... status: {pending, ...} customer: <phone: String, name: String> restaurant: <id: String, name: String, location: String> ..tables: InfoTable [*]>
changeReservationStatus	changeReservationStatusReq= <locator: String, status {pending, ...}>	changeReservationStatusRes= <code: Integer, message: String>
deleteRestaurantTable	deleteRestaurantTableReq= <restaurant_id: String, table_number: String>	deleteRestaurantTableRes= <code: Integer, message: String>

b) Servei web SOAP utilitzant el mode d'interacció Request/Acknowledge/Poll per a l'operació fer una reserva nova

Operation name	Input	Output
makeNewReservation	makeNewReservationReq= <restaurant_id: String, customer_phone: String, ..table_numbers: Integer [*], date: Date, ..service: {lunch, dinner} ..num_people: Integer>	makeNewReservationAck= <id: String>
getResults	makeNewReservationAck = <id: String>	makeNewReservationRes= <code: Integer, message: String, locator: String [0..1]>

c) Api REST per a les 5 operacions

Title	Consultar la info d'un Restaurant concret	
URI	/api/restaurants/:id	
Query Params		
Method	GET	
Body Data		
Returns	Success Response	200 OK { "id": : "CELLERROCA" "name": "El Celler de Rocallaura", "location": "Rocallaura", "tables": [{"number": 1, "capacity": 4}, ...] }
	Error Response	404 Not Found {message: "no Restaurant with this id"}

Title	Consultar reserves	
URI	/api/reservations	
Query Params	restaurant_id=[String]: optional customer_phone=[String]: optional date=[String]: optional service=[...]: optional status=[...]: optional	
Method	GET	
Body Data		
Returns	Success Response	200 OK [{"locator": "AEE344", "date": "2018-10-03", "service": "lunch", "num_people": 6, "status": "pending", "restaurant": {"id": "CELLERROCA", ...}, "customer": {"phone": "965414605", ...} "tables": [{"number": 4, "capacity": 8}] }, ...]
	Error Response	400 Bad Request {message: "incorrect service / status"}

Title	Modificar status d'una reserva	
URI	/api/reservations/:locator	
Query Params		
Method	PUT	
Body Data	{ "status": "cancelled" }	
Returns	Success Response	200 OK { "locator": "AEE344", "date": "2018-10-03", "service": "lunch", "num_people": 6, "status": "cancelled", "restaurant": { "id": "CELLERROCA", ... }, "customer": { "phone": "965414605", ... } "tables": [{ "number": 4, "capacity": 8 }] }
	Error Response	404 Not Found { message: "no reservation for this locator" } 400 Bad request { message: "wrong status" } 409 Conflict { message: "illegal status change request" }

Title	Eliminar una taula	
URI	/api/restaurants/:id/tables/:number	
Query Params		
Method	DELETE	
Body Data		
Returns	Success Response	204 No content
	Error Response	404 Not Found { message: "no table for these id & number" } 409 Conflict { message: "cannot delete: active reservations" }

Title	Fer una nova reserva	
URI	/api/reservations	
Query Params		
Method	POST	
Body Data	<pre>{ "date": "2018-10-04", "service": "lunch", "num_people": 6, "restaurant_id": "CELLERROCA", "customer_phone": "965414605", "table_numbers": [4] }</pre>	
Returns	Success Response	<pre>201 Created {"locator": "AEE34X", "date": "2018-10-04", "service": "lunch", "num_people": 6, "status": "pending", "restaurant": {"id": "CELLERROCA", ...}, "customer": {"phone": "965414605", ...} "tables": [{"number": 4, "capacity": 8}] }</pre>
	Error Response	<pre>400 Bad request {message: "wrong parameters"} 409 Conflict {message: "no enough room in tables2", etc.}</pre>

d) Client

1. Obtinc totes les reserves amb status=pending i customer_phone=965414605:

GET /api/reservations?customer_phone=**965414605**&status=**pending**

2. Per cada reserva amb locator = NN obtingut, faig la petició següent:

PUT /api/reservations/**NN**
{"status": "cancelled"}