

Lab session 2

The RéZa resource management system

A company has hired you to design an application named RéZa, dedicated to management of time tables and resource allocation, in particular meeting rooms.

The system is designed for medium to large companies that manage large office spaces, typically universities or schools, conference centres...

The application will be used by 3 category of users :

- The administrator can create/modify/delete the description of rooms (room capacity, presence of equipment such as a video-projector ...)
- The manager validates or invalidates the room reservations submitted by the end users.
- The end user can visualize the time tables and submit a reservation for a room.

The time-tables display can be flexibly adapted so that user scan access the calendar presentation that best meets their search criteria. There are two main types of calendars:

- By category: each reservation contains a field « category », that allows to specify for which event the reservation is designated. For instance « Adv. UML SIA ». The search by category function allows to obtain a time table that covers all reservation occurrences that match a given category.
- By room: the user can also request a time-table for a specific room, over a given period. The user then only needs to specify the desired room and start/end dates.

The RéZa system should help users find an available room and make a reservation for it.

There are two type of reservations : single reservations and periodic reservations.

For any reservation, the user should provide his name (e.g. “John Smith”) the category of the reservation (e.g. « Adv. UML SIA »), a name for this reservation (e.g. « Final Exam»), the room desired (e.g. « Amphi Rouge »), and the date and time slot desired (e.g. « 20 March, 15h30-17h30 »).

If the reservation is periodic, the user will specify it on the same page, by giving the periodicity (daily, weekly or monthly) and the number of occurrences (e.g. « Conference ICALP », 5 daily occurrences for a full week reservation).

Any user can submit a room reservation. A periodic request for n occurrences will be handled as id the user had made n independent single reservations from the point of view of validation (i.e. each occurrence can individually be validated or not).

The manager can visualize all current reservations that are not yet validated. He can inspect them one by one and decides to validate or refuse them. In any case, the user that asked for this reservation will be notified by email of the evolution of his reservation status. The manager bases his decision of validating or not a request on what he knows of the event category, the user who made the reservation, and his knowledge of other constraints on agenda and room projected usage.

ANALYSIS

Question 1 (2 point)

Build a use case diagram for this system.

Comment/annotate your diagram.

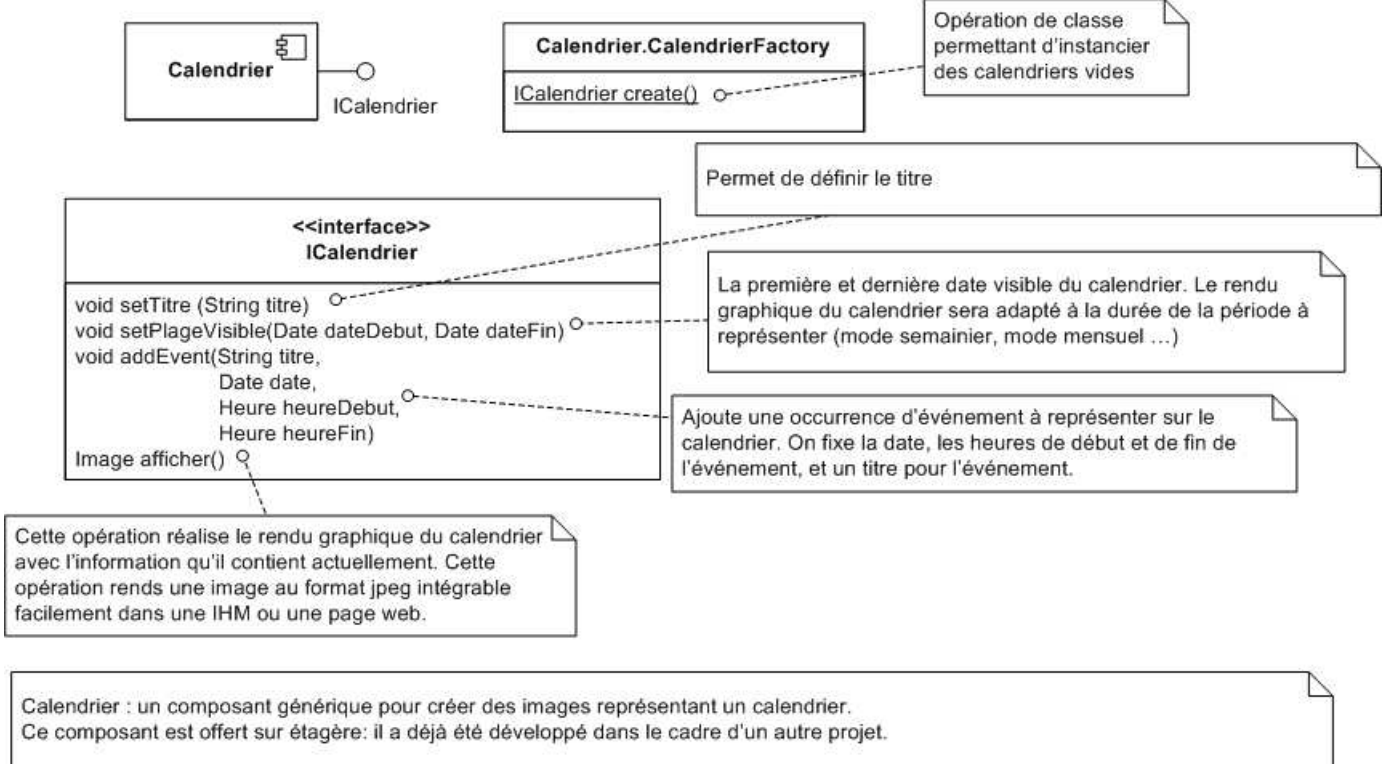
Question 2 (3 point)

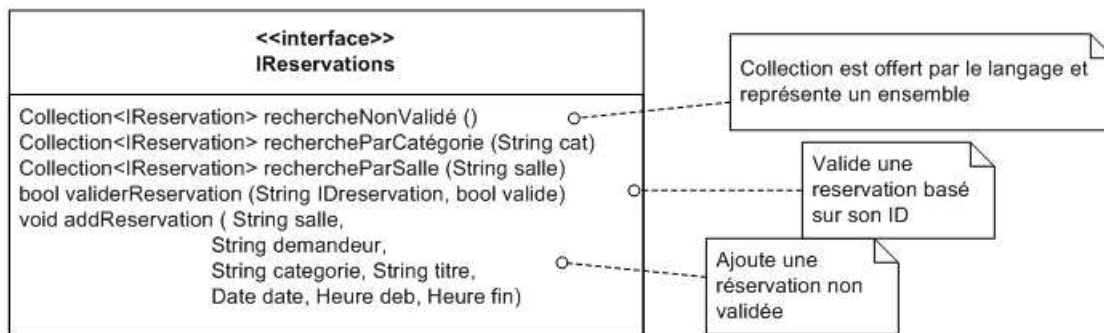
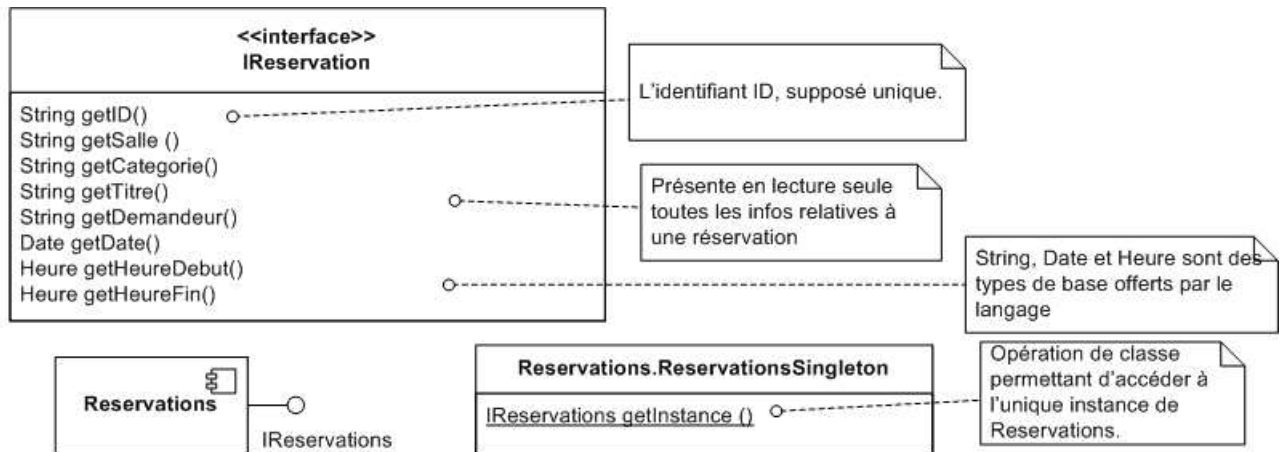
Build an analysis class diagram for this system (no operations, simple relationships and attributes only).

We will not detail the operations of the (fictive) “System” class.

Design Phase

Design does not start from scratch : the development team proposes to reuse two existing components developed for another project : the Reservation component and the Calendar compont. They are described on the following figure (the design team was French...).





Reservations : un composant « référentiel » pour stocker les réservations, validées ou non, de façon persistante. Offre diverses fonctions de recherche.

Question 3 (1,5 points)

- What is the role of the two classes « CalendrierFactory » and « ReservationsSingleton » ? How would you use them (in pseudo-code) ?
- The Reservations component is in reality implemented by interaction with a relational database. Why do you think a singleton is used here ?

Question 4 (2 point)

We have decided to build three GUI components dedicated respectively to the administrator, the manager and the user.

A component plays the role of controller for the system, and serves the needs of these GUI while using the Reservation and Calendar components defined above. It should play the role of Facade for the GUI, that only interact with this component.

Describe this situation using a component diagram.

Question 5 (4 points)

Using one or more sequence diagrams of integration level, i.e. where the lifelines represent components of the system, model the interactions needed to ask for a single reservation of room B1 for March 20, 15h-17h, then the steps needed to validate this request by a manager.

The diagrams can be annotated with free syntax comments or pseudo code to explain the more tricky parts (loops...).

Question 6 (3 point)

Using one or more sequence diagrams of integration level, i.e. where the lifelines represent components of the system, model the interactions needed to for a user to display a calendar showing room B1 for the period March 15 to March 30.

The diagrams can be annotated with free syntax comments or pseudo code to explain the more tricky parts (loops...).

Question 7 (3 point)

Deduce from questions 4 to 6 the signature of the operations needed in the interfaces that link the GUI of the user and of the manager and the RéZaCore controller component.

Question 8 (3 point)

Write an integration test for component Reservations, that allows to ensure the correct behavior of the AddReservation(...) operation. Use pseudo-code or a sequence diagram to answer.

Question 9 (3 point)

Provide a possible detailed design of the Reservations component, using a class diagram. Consider first the case of a DBMS implementation, with as much detail as possible (pseudo-code in notes for the more complex operations).

Then consider an implementation relying on local storage using XML files, and design an implementation (mostly the structure, details not required).