OOD & WAD final assignment
KidsSP - the Shopping Playroom

Parents that go shopping often need help with the caring of their children. The company *KidsSP* has created a solution for that problem by installing playrooms in several shopping centers. Parents can leave their children there to play while being carefully attended and supervised by specialised personnel.

In this assignment, you will design part of a software system that will support the business activities of the company *KidsSP*. The system will comprise a small website and a couple of desktop applications.

Your job is to provide an object-oriented design of the global logic layer (also called business layer) of the system. This logic layer must be represented in only one cohesive class diagram and must be usable from all possible presentation layers, be it the front-end of the website or the user interface of the different desktop applications. You may assume a common object-oriented programming language would be used both for the back-end of the website and the desktop applications. Therefore, it is important that you design just *one integrated logic layer* for the whole system. Information related to the data access layer will be provided later in this document.

Registration

To assume the responsibility of taking care of children of other people is a delicate issue. Therefore, parents[1] are first required to register their family unit via a website and, afterwards, *KidsSP* does some background checking on the parents before the family is accepted as client.

Data that must be provided in the registration include full name and date of birth of each parent, address, mobile telephone numbers and e-mail addresses. For each child, full name and date of birth suffice but, optionally, a comment might be added about her/his possible special needs (for example, allergies or behavioural peculiarities). A family unit can comprise up to four parents and will be registered under just one website account.

The background check on the families is performed by *KidsSP* administrators through contact with relevant governmental organisations. This check takes place completely out of our software system. However, administrators must finally mark each family as approved or rejected. Parents must be able to check this outcome via the website. In any case, once a decision has been reached, an e-mail is sent

[1] Here, "parent" is understood in the most possible general sense, which includes biological parents, adoptive parents, step-parents and legal guardians.

to the family and, if accepted, they will receive a unique *KidsSP* identification number. Rejected families are allowed to modify their information and apply for a new check.

Only after successful registration and approval, a family will be able to make use of the playrooms.

Checking in and checking out

A successfully registered and approved family can make use of any of the available *KidsSP* playrooms when visiting a shopping center with one. The procedure that must be followed for checking in at the beginning of the stay and for checking out at the end of it are now described.

For checking in, one of the parents provides the *KidsSP* identification number and her/his date of birth to the receptionist at the entrance of the playroom, and the receptionist then checks via the system whether this information is correct. Subsequently, the parent informs which of the children (that the receptionist can already see via the system as registered in the family unit) will stay in the playroom on that day. The receptionist also asks which of the registered mobile telephone numbers must be used on that day in case of an emergency. Each of the children will be provided with a bracelet with an RFID-tag[2], which will all be read by an RFID-scanner connected to the system. These RFID-bracelets can then be used to easily and quickly get the information of the parents if needed. It is important that each child can be immediately identified by her/his RFID-bracelet during the playroom stay. After the checking in process is completed, the parent can kiss the children goodbye and go away to do the shopping.

After the shopping is done, the parent comes back to the *KidsSP* playroom to pick up the children and do the corresponding checking out. Based on the starting time and the finishing time, the system calculates the basic cost of the stay according to a certain hourly rate. To this basic cost, possible extras are added to determine the final total cost (information on this in the next two sections). The final bill is then settled and the system receives confirmation that the payment was completed[3]. The RFID-bracelets used by the children are returned, which the receptionist stores for re-use.

The *KidsSP* manager has stated that it can be safely assumed that all children of a family that stay together will also leave together at the same time. However, if your design allows for some children to leave first and the rest at a different later time, that would be a welcome improvement.

Candies and drinks

Some candies and drinks are available in the *KidsSP* playrooms for the children to buy and consume during their stay. To avoid the use of money by the children, their RFID-bracelets will be used to add these costs to the bill of the stay.

[2] See section "Extra: On RFID scanning" below for more information.
[3] For this initial version, it is assumed that payment occurs outside of our software system. Therefore, our system only needs confirmation that payment was succesfully made.

In case the parents want to limit the possible consumption of their children, a limit can be set to the amount of money to be spent in the playroom stay during the checking in process. This limit is global for the group of children of the family staying together.

Candies and drinks are identified in the system only with a name and a price.

Special gadgets

The playrooms of *KidsSP* contain a considerable assortment of toys and games to be used freely. They are all in the open room where the children stay and play. That is what makes the company an attractive business for the families.

However, a few special gadgets are, due to safety concerns or to their high cost, only to be used by the children by loan. Same as with buying candies or drinks, children can be registered to have loaned one of these gadgets during their stay through their RFID-bracelet. Once the gadget is returned, the cost is added to the bill using an hourly rate. (Sometimes, this hourly rate is just zero, which means that the gadget can be loaned for free, but *KidsSP* still needs to keep control of which child was the borrower.)

The limit that can be set to the amount of money that can be spent on candies and drinks, also holds for the rental of special gadgets. At the moment of loan, it will be checked that one hour of rent does not exceed the limit; it will then be up to the corresponding *KidsSP* employee to guarantee a timely return of the item.

These gadgets are identified in the system just with a name, an hourly rate and a total cost to be paid in case of loss or breakage (this total price is not to be compared to the money limit of stays when loaning).

Handling of emergencies

If an emergency situation arises with one of the children, scanning her/his current RFID-bracelet should allow for quick access to the family unit information. This must specially include the highlighted mobile telephone number for the stay, but also all the rest of the information of the family that might be helpful in handling the emergency.

Note that the retrieved information as described above is very important in emergency situations: the highlighted number must be called first and, if this number is not quickly reachable, all other numbers of the family and all names of the parents must be readily available to continue trying to reach someone in the family.

Management

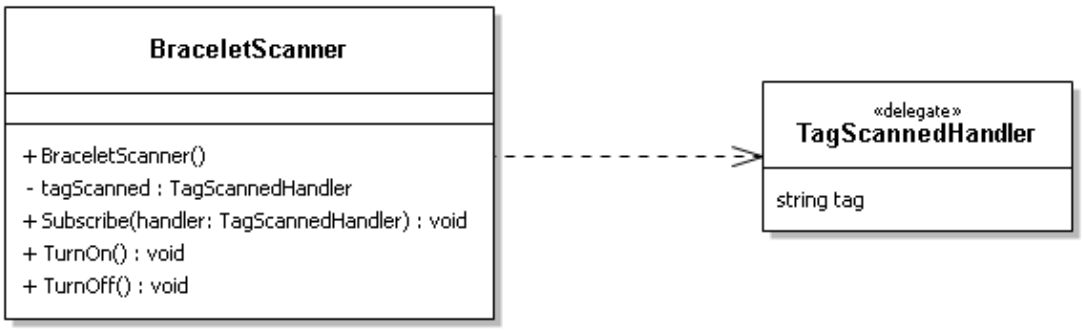
A general management application should allow administrators of *KidsSP* to see overviews of all paid bills, possibly filtering by range of dates or by playroom or by family. This application will also provide the typical (CRUD) operations for administering the playrooms, which only need an address and an automatically generated identification code. Finally, the approval or rejection of family units will also be handled here.

The rest of the (CRUD) operations for candies/drinks and for special gadgets will be provided by the application to be used in each playroom by the general employees. The checking in and checking out procedures will be carried out through this application, as well as the handling of emergencies.

Extra: On RFID-scanning

The technology of RFID-tags (Radio-Frequency IDentification tags) is based on electromagnetic fields and allows automatic identification of objects, in our case, rubber bracelets. When an RFID-bracelet (that is, a bracelet containing an RFID-tag) is placed near an RFID-scanner, also called RFID-reader, digital communication is activated and the scanner obtains the tag.

We can assume that connection to an RFID-scanner has already been implemented for us. After attaching a scanner to the computer, it can be used through the following class:



Objects that might need to obtain tags from the RFID-bracelet just need to subscribe to the scanning event with the "Subscribe" method above and a parameter of the corresponding delegate type.

Extra: On the data access layer (and more)

The *KidsSP* manager has stated that it is not clear if they can initially invest in access to a database server. This means that an initial version of the system could work just with a non-sophisticated data access layer, like some simple object serialisation on files. However, your job will be to design only the global logic layer *without including the data access layer*.

Actually, it might even happen that the initial version of the system will not use a web server. This would force relocating the functionalities of the website described above to the desktop application used at the reception of the playrooms. This makes it even more important to count with just *one integrated global logic layer*.

Assignment

- First, design a base UML class diagram for the system with the classes, attributes and their types, associations, multiplicities, generalisations and dependencies. Make sure that you give all your associations reasonable navigability directions. We recommend that you perform this first step without including methods/constructors in the design (which is why we call it a *base* diagram).

  The presentation layer and the data access layer of the system must not be included in the class diagram.

- Next, modify your base UML class diagram by adding methods/constructors and, if necessary, other components to support the following functional requirements:

  As a parent I should be able to:
  FR-1-01(-): Register my family unit via the website.
  FR-1-02(-): Modify information of my family unit via the website.
  FR-1-03(-): Check the approval/rejection status of my request via the website.
  FR-1-04(-): Pay the bill for my children's stay.
  As a receptionist/caretaker employee I should be able to:
  FR-2-01(-): Check in children for a playroom stay. FR-2-01 Includes setting a limit to the consu
  FR-2-02(-): Check out children at the end of their stay.
  FR-2-03(-/+): Give out candies/drinks or gadgets to children in the playroom.
  FR-2-04(+): Receive back loaned gadgets.
  FR-2-05(-/+): CRUD candies/drinks and gadgets in my playroom.
  FR-2-06(+): Handle possible emergency of a child.

  As an administrator I should be able to:
  FR-3-01(-): Approve/reject family units (after "manual" checking finished).
  FR-3-02(-): CRUD playrooms.
  FR-3-03(-): See overview of all paid bills.
  FR-3-04(+): See filtered overview of paid bills[4].

- Requirements marked as **(-)** are the minimum needed for a design that is satisfactory enough (read: formative indication S), while requirements marked as **(+)** are necessary for a good design (read: formative indication G or O).

[4] See filters above in section "Management".

There are two requirements above marked as **(-/+)**. In both cases, **(-)** refers to candies/drinks and **(+)** to gadgets.

- Submit your UML class diagram on Canvas after converting it to PDF format or PNG format (or both).

----- ----- ----- ----- -----