

Table of Contents

Business Objects and Relationships	2
---	---

Business Objects and Relationships

The first and most important step in designing a system is to identify business objects and business rules of the system. If you get this right everything else will be relatively straightforward.

You need to carefully look at the requirements and identify the main business elements that the system deals with (business objects), relationships between them and rules according to which these objects behave. At this stage it is very important that you do not worry too much about how the system will look to the user and how the user will interact with the system.

So we start by identifying business objects and relationships between them. For clarity we will temporarily not consider some of the requirements – we will come back to them later after we have configured the core of the system. The requirements that we will not consider at this stage are the management of [communication with library members](#) and the [credit card payments](#).

Some of the business objects are quite straightforward – if we carefully study the requirements we can identify the following business objects:

1. Item
2. Member
3. Loan
4. Reservation
5. Fee
6. Payment

Let us take these objects as a starting point and see if there are any relationships between these objects. One of the straightforward relationships is the relationship between Member and Loan – a member can have multiple loans; but a loan is always for some particular member. So there is a multiple relationship between Member and Loan and a single relationship between Loan and Member.

Similarly there is a clear relationship between Loan and Item – a Loan knows which item has been borrowed; an Item knows which loan is registered against it (if any) or which loans have been registered in the past. So there is a single relationship between Loan and Item and a multiple relationship between Item and Loan.

Is there a relationship between Item and Member? Yes there is, since an item should know which member has borrowed it or which members have borrowed it in the past. However, the Item object already has the relationship with the Loan object and the Loan object – with the Member object, so we already have a relationship between the Item and Member objects – through the Loan object. Therefore the relationship between Item and Member is indirect. In our exercise of identifying relationships we should only focus on direct relationships.

Looking at other objects we can conclude that there are the following direct relationships:

Business Object	Is Related To
Item	Loan (an Item can have multiple Loans – one current and several past)
	Reservation (an Item can have multiple Reservations – both current and past)

Business Object	Is Related To
Member	Loan (a Member can have multiple Loans – both current and past)
	Reservation (a Member can have multiple Reservations – both current and past)
	Fee (these are Fees the Member have been charged – both current and past)
	Payment (these are Payments that the Member has made)
Loan	Member (this is the single Member that borrowed an item)
	Item (this is the single Item borrowed)
Reservation	Member (this is the single Member who made the Reservation)
	Item (this is the single Item that has been reserved)
Fee	Member (this is the single Member who was charged)
Payment	Member (this is the single Member who made the payment)

The above business objects and relationships are relatively obvious. Another business object is slightly harder to identify.

If we look at the [requirements](#) we will notice that items can be of different types. What is the type of the item? It is clearly something that belongs entirely to the item, so it could be the attribute of the Item object. However, if we look at the [table](#) we will notice that there is some extra information associated with the item type, i.e. maximum number of loans, loan period and maximum number of renewals. Where does this information belong? If item type were an attribute of the Item object then this information would have to belong to the Item object as well. In this case every instance of the Item object would store the values of maximum number of loans etc. This is very impractical since according to the requirement in Error: Reference source not found.2 every now and then the library manager needs to change this information. If this information were stored in every instance of the Item object the manager would have to change it in every item that exists in the library instead of changing these values in one place!

This means that the ItemType is an object in its own right and that the maximum number of loans, loan period and maximum number of renewals are attributes of this object. If a manager wants to change the values of these attributes she should just find the ItemType objects and change the values there. The Item object should therefore be related to the single ItemType object that it belongs to.

Therefore we add the ItemType object to our list of objects and the following relationships to our list of relationships:

Business Object	Is Related To
Item	ItemType (this is the single ItemType that it belongs to)
ItemType	Item object (these are multiple Items that belong to this ItemType)

- [Attributes of Business Objects](#)
- [Configuration of Business Objects](#)

From:
<http://www.awareim.com/dokuwiki/> - Documentation

Permanent link:
http://www.awareim.com/dokuwiki/docs/6000_case_study/0400_configuration/0500_business_objects_relationships

Last update: 2023/05/09 02:18

