# ORGANIZATIONAL LIBRARY

**Loredana MOCEAN**
**Miranda Petronella VLAD**

***Abstract:*** *This paper describes a web application for managing devices and books that are in a company's library, making the library accessible in digital format through the application.*
*By using the tools provided, the company's employees have the opportunity to see the items that are made available to them for the loan, right from the office. They have a user account, which can be accessed from computer and tablet or phone. The loan process becomes safer and shorter, and communication more efficient, with the app providing real-time accurate stock status information as well as the loan period.*
*The real benefit to the company is primarily offered by the existence of a library that contains titles that fall within the scope of activity and smart devices designed to provide support in the testing phase of projects.*
*The application, being implemented locally, is designed to fit the design and functionality of the company's needs and profile. This customization, which takes over the user experience, is not possible by using an existing system on the internet.*
***Keywords:*** *library, management, database, technology, web application*

## Introduction

Many organizations that want to provide their employees with technical activity books or smart devices do not have a computerized system or at least not all of them. Because of this, they continue to pursue a manual approach to dealing with transactions and database management. The tasks that come with the need to maintain such a service performed manually, lead to a slow, tiring activity and prone to errors. The design of the application aims to reduce the time required to carry out the activity of the library within the companies that have one. It is useful both in terms of loan functionality and administration.

We find here two types of causes that have generated the need for such a system.

Main categories: human resources, data recording, professional development, communication and the need for digitization.

We can distinguish the following secondary causes, which derive from those mentioned above:

- Lack of loan management staff;
- Creating media for employee development
- Information on existing stocks;
- Employee time efficiency;
- The need for integration into the digital age and alignment with modern times;
- Statistics based on user history;
- Control over the loan process;
- Administration of books and devices;
- User account management.

## General objectives of the application

The top-level objectives are as follows (see Figure 1):

- User authentication;
- Access to a gallery with available articles;
- Gallery search and filtering functionality;
- Account administration;
- Personal loan management;

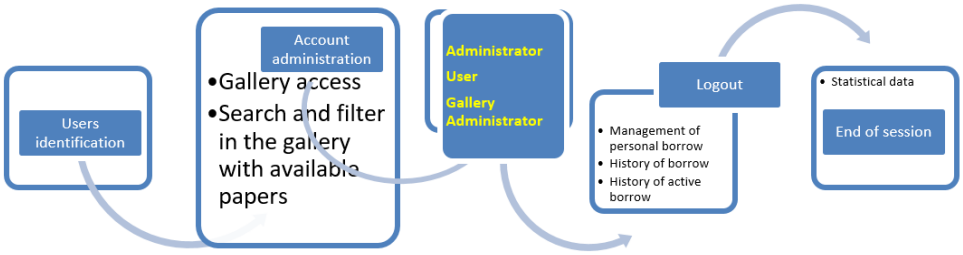



**Fig. 1.** General objectives of the application

From the objectives described above derive, following the principle of inheritance, the following:

- Distinction between administrator and user roles;
- Making the loan;
- Gallery administration;

- Editing the user profile;
- Editing active loans;
- View a loan history and statistics.

Defining a goal is the best way to extract system requirements. Thus, the success of the project depends on whether the objectives have been achieved or not.

## Technological concepts

The web application aims to provide end users with a pleasant and easy experience in the process of borrowing an item available in the company's library. Thus, it becomes more secure and shorter, and communication more efficient, the application providing real-time accurate information about the status of the stock, as well as the loan period.

One of the most important conditions that must be met for a business to succeed is to motivate employees. Once a company makes an effort to meet this requirement, the chances of employees being more efficient and productive increase significantly. Another benefit to the company, observed in the long run, is the loyalty of employees by providing the necessary devices to facilitate their work. The idea of motivating employees also involves creating an attractive work environment. Among the tools used in this regard are tangible motivational factors, among which we find the material benefits.

At present, the Internet is full of library management systems. Although these software solutions offer a wide range of options, the costs are quite high and tend not to be borne by many of the institutions that would like to implement such software (for example, libraries or schools). A cheaper option is cloud-based or subscription-based systems. Another existing option is that of open-source systems that can provide free support for the entire period of use or for a limited period.

For personal use or in the context of a small number of users, mobile applications have also been developed that are used to manage articles in a library. The idea of the application started from a system of organizing books in a library, adapted to the needs of a company in which each employee has the opportunity to manage the loans of items from their own account. Thus, the administrator who uses the same application, has privileges on the loans made and on the management of books, devices and users.

The application, being implemented locally, is designed to represent the needs and profile of the company through design and functionality. This customization, which takes over the user experience, is not possible by using an existing system on the internet.

An application of this type in a company does not offer employees only a software product, but an experience. This experience aims to promote both personal and professional development, and encourages and supports the lending of books and smart devices.

Implementing such a system within a company becomes imperative in solving the problems raised. A web application that provides easy access to this service by reducing physical work, is the answer that comes after the needs of users. The design of the application aims to reduce the time required to carry out the activity, both in terms of loan functionality and on the administrative side. Thus, the increase in efficiency will have a positive impact on the interest shown by employees in the arranged library. We can say that it is a way to bring it closer to them and easier to access.

## Computer system architecture

At the technological level, the computer system of the application consists of three components (see Figure 2):

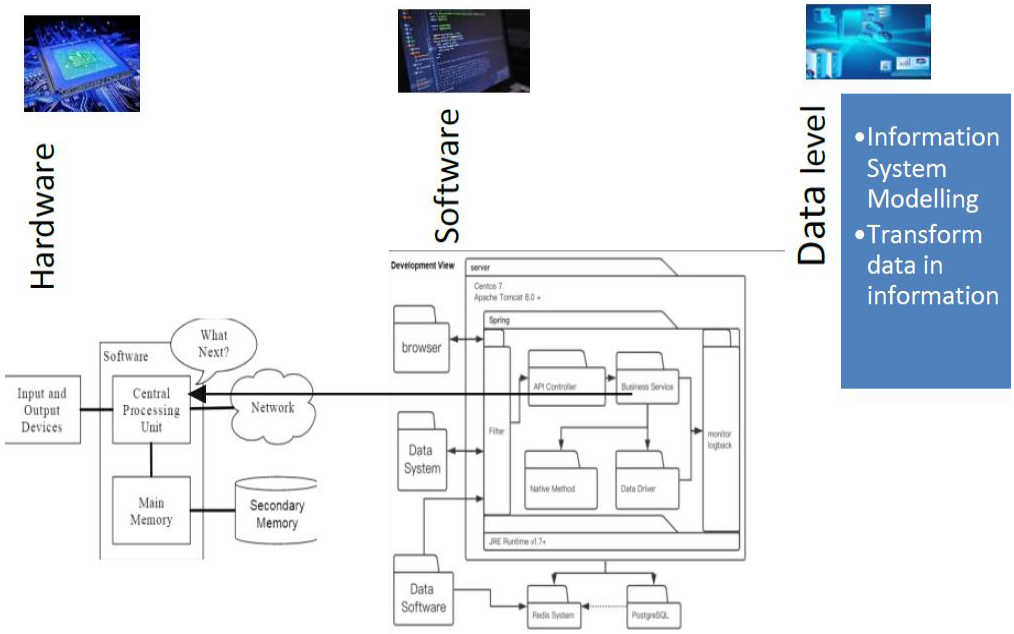



**Fig. 2**. Computer system components structured on three levels.

The data is integrated into a database. Thus, it can become a powerful tool in modeling an information system.

Among the advantages of implementing such a system we list:

- offers an elegant user experience,
- make quick updates possible,
- is more cost-effective and resource-efficient (such as memory and hardware),
- helps to significantly reduce the redundancy that can occur on data
- maximizes its' integrity.

Due to the fact that the data is stored in one place, the system offers a higher general degree of security and makes it easier to access and manipulate the database. The disadvantages of a centralized computer system are not a real impediment in this context. Its implementation for an average number of users and the nature of the information it is intended to provide make its negative parts unlikely.

## Development model and technical design

As a methodology for developing a software product, we chose a method that falls under the scope of the Agile philosophy, namely Kanban. Kanban is a way to outline, manage and improve your workflow to achieve your goals. This system aims to achieve efficiency and agility in the production process of the software.

His approach helped to maintain transparency about the tasks to be performed, always having a vision of what needs to be done and what has already been done. The tool used to implement this strategy was a virtual board that allowed the encapsulation of tasks and the visualization of how they progress.

The first step in the actual development was to outline the entities, their attributes and the relationships between them, thus constituting the schema of the database. This step is identified as "data modeling and involves a system documentation technique according to Rădulescu (2015), so through this representation we obtained a high level of data abstraction". (see figure 3).

We chose to approach a relational basis, because we needed a structuring of the data and the certainty that it will not take another form. At the same time, the motivation is based on the fact that we do not work with large data sets that exceed the capacity of such a database.

The main feature of relational databases is the rigid structure they have to follow. The data is organized into tables that represent the entities in the database, to which attributes are attached, which are the properties used to describe each entity.

Tables are linked by relationships that can be of several types. Each instance of the entities must be uniquely identified in one way or another. The primary keys are used for this purpose. Secondary keys are used to show the dependency of the tables.

In the case of this application, the data are based on the scheme of 6 entities that we have named suggestively, depending on what each one represents.

Probably the most common technique for building a physical database is starting from its diagram, which is called "forward engineering". The opposite method, the one I also approached, involves the creation of the digram starting from the schema of the existing database, a process called in reverse terms "reverse engineering".

We started from an initial scheme, and by implementing it in the form of models we also normalized the database. Standardization involves the application of rules on the preliminary database to ensure the correct structure of the tables. Once we had the models corresponding to each table implemented and the database configured with its name and the port on which it runs, we performed the migration operation.

The last step is to create an additional SequelizeMeta table if it does not exist, which is used to record the rolled migrations, run the current migration, and create the related tables.

*The structure of the database is as follows:*

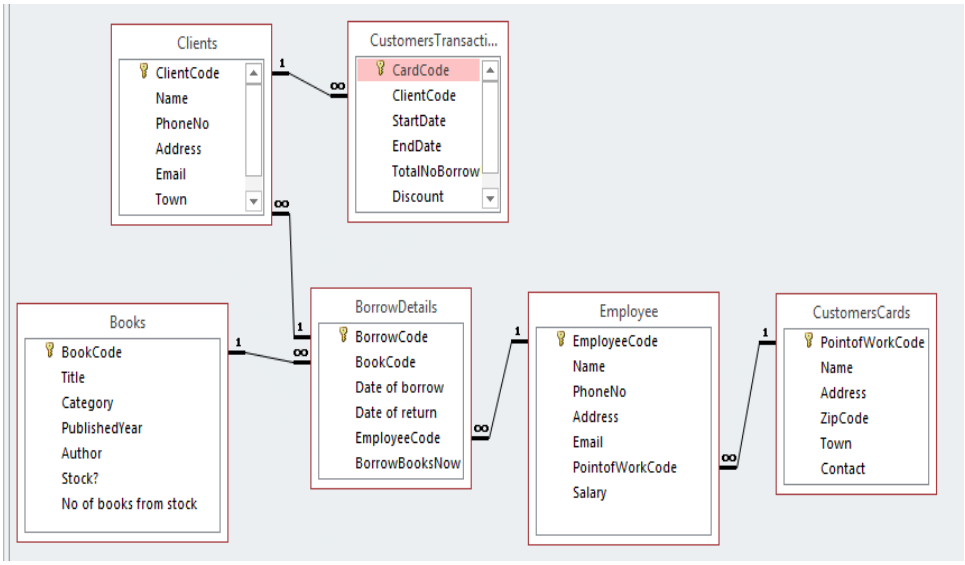



**Fig. 3**. Database relationships

## Technology description

To write the source code needed to make this web application the JavaScript programming language subject to the ECMAScript standard (ES6) is used. According to studies conducted by Stackoverflow for 2019, it appears that the JavaScript programming language ranks 1st for the 7th year in a row in programming, scripting and markup languages. This is a very popular language for web development. Basically being a client-friendly language that helps create dynamic and interactive content, "integrated with Node.js technology, JS can also be used to implement server-side code, as stated in Brown ( 2014) ".

A web application is an improved form of a website in that it provides various features that allow it to perform many tasks. This is also the purpose of this application. Its implementation aims to facilitate a user's access to a library, as well as to facilitate the work done by the administrator of the library in question.

In order to develop the frontend component we used the Vue.js framewrok. This is "a progressive technology according to Filipova (2016)", suitable for the MVC architecture we have implemented. It is used to create the application interface. Released in 2014, it is in continuous development, offers HTML extension and is based on JS.

The unknown side of the consumer that benefits from the application is the server side, called the backend.

In order to make it possible to run the JS language on the server side, ie independent of the browser, the Node.js platform is used. The setup process is easier and more accurate. Another aspect to mention is that "it works in single thread system, using asynchronous programming, as explained in Brown (2014)", so the logic of the application becomes more simplified, but without sacrificing performance. Another compelling benefit is the independence of the platform, even if it is not the first or only technology of this kind, node.js can be configured on all major operating systems (Windows, OS X, Linux) and allows easy collaboration.

The programming environment used is the main tool in developing an application. Visual Studio Code is a suitable and capable editor for the JS language. It manages to combine simplicity with powerful tools in application development, such as IntelliSense, understanding and navigating code semantics, and refactoring it. It also allows the installation of various extensions that become necessary in optimizing the experience.

Filtering is a useful way to see the data we want to display in Access databases. We may use filters to display certain records in a form, report, query, or data sheet, or to print only certain records in a report, table, or

query. By applying a filter, we can limit th data in a view without changing the design of the object.

*We will present some filters in the database. The proposed queries are described below.*

*1. Books borrowed by clients*

The SQL Code is printed in the next caption.

```
SELECT Clients.Name, Clients.BorrowCode, BorrowDetails.BookCode, Books.Title
FROM Books INNER JOIN (BorrowDetails INNER JOIN Clients
ON BorrowDetails.[BorrowCode] = Clients.[BorrowCode]) ON Books.[BookCode] = BorrowDetails.[BookCode];
```

The results:

| Name | BorrowCode | BookCode | Title |
|---|---|---|---|
| Maria | 500 | 200 | Pe aripile vantului |
| Ana | 501 | 201 | La rasarit de eden |
| Ioana | 502 | 202 | Rosu si negru |
| Diana | 503 | 203 | Fundatia si imperiul |
| Bianca | 504 | 204 | Singur pe lume |
| Alexandra | 505 | 205 | Zece negri mititei |

*2. Clients who borrowed at least one book since now*

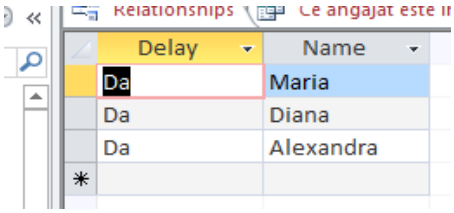
```
      SELECT BorrowDetails.BorrowBooksNow, Clients.Name
      FROM BorrowDetails INNER JOIN Clients ON
BorrowDetails.[BorrowCode] = Clients.[BorrowCode]
      WHERE (((BorrowDetails.BorrowBooksNow)<>0));
```

| BorrowBook | Name |
|---|---|
| 3 | Ana |
| 2 | Ioana |
| 1 | Diana |
| 1 | Alexandra |

*3. Clients who delayed in returning the books:*

```
      SELECT CustomersTransactions.Delay, Clients.Name
      FROM Clients INNER JOIN CustomersTransactions ON
Clients.[ClientCode] =
CustomersTransactions.[ClientCode]
      WHERE (((CustomersTransactions.Delay) = "Yes"));
```

The results are printed in the next caption.

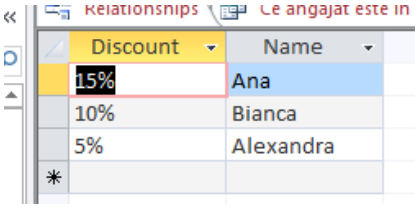


| Delay | Name |
|---|---|
| Da | Maria |
| Da | Diana |
| Da | Alexandra |

*4. Customers who have a discount*

```
      SELECT CustomersTransactions.Discount,
Clients.Name
      FROM Clients INNER JOIN CustomersTransactions ON
Clients [ClientCode] = CustomersTransactions
[ClientCode]
      WHERE (((CustomersTransactions.Discount) <>
"0%"));
```

The results are printed in the next caption.

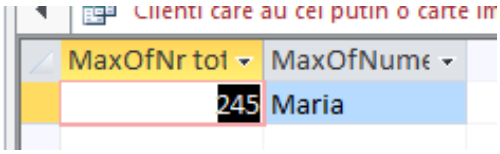


| Discount | Name |
|---|---|
| 15% | Ana |
| 10% | Bianca |
| 5% | Alexandra |

*5. The client who borrowed the maximum number of books*

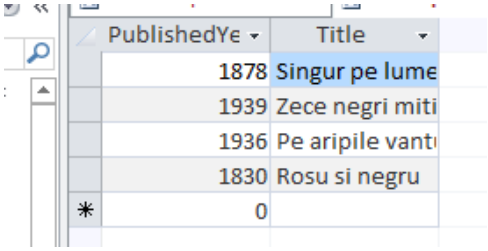
```
      SELECT Max
(CustomersTransactions.TotalNoBorrowBooks) AS [MaxOfNo
total borrowed books], Max (Clients.Name) AS MaxOfName
      FROM Clients INNER JOIN CustomersTransactions ON
Clients [ClientCode] =
CustomersTransactions.[ClientCode];
```

The results are printed in the next caption.



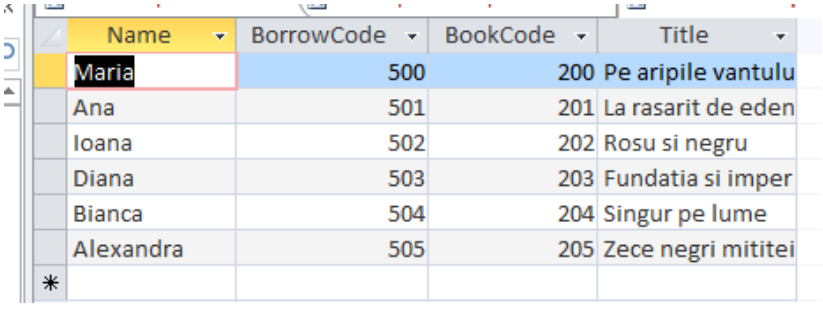

*6. The books edited until 1939*

```
SELECT Books.PublishedYear, Books.Title
FROM Books WHERE (((Books.PublishedYear)<=1939));
```



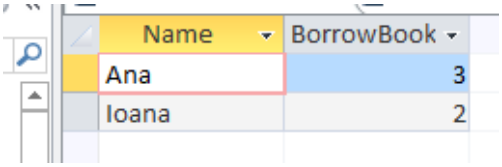
| PublishedYe | Title |
|---|---|
| 1878 | Singur pe lume |
| 1939 | Zece negri miti |
| 1936 | Pe aripile vanti |
| 1830 | Rosu si negru |
| 0 | |

*7. The borrowed books filterd by clients*



| Name | BorrowCode | BookCode | Title |
|---|---|---|---|
| Maria | 500 | 200 | Pe aripile vantulu |
| Ana | 501 | 201 | La rasarit de eden |
| Ioana | 502 | 202 | Rosu si negru |
| Diana | 503 | 203 | Fundatia si imper |
| Bianca | 504 | 204 | Singur pe lume |
| Alexandra | 505 | 205 | Zece negri mititei |

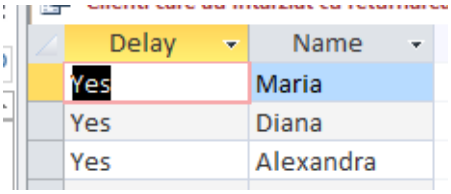
```
     SELECT Clients.Name, Clients.BorrowCode,
BorrowDetails.BookCode, Books.Title
     FROM Books INNER JOIN (BorrowDetails INNER JOIN
Clients ON BorrowDetails.[BorrowCode] =
Clients.[BorrowCode]) ON Books.[BookCode] =
BorrowDetails.[BookCode];
```



| Name | BorrowBook |
|---|---|
| Ana | 3 |
| Ioana | 2 |

*8. Clients that borrowed at least two books*

```
      SELECT Clients.Name, BorrowDetails.BorrowBooksNow
      FROM BorrowDetails INNER JOIN Clients ON
BorrowDetails.[BorrowCode] = Clients.[BorrowCode]
      GROUP BY Clients.Name,
BorrowDetails.BorrowBooksNow
      HAVING (((BorrowDetails.BorrowBooksNow)> = 2));
```

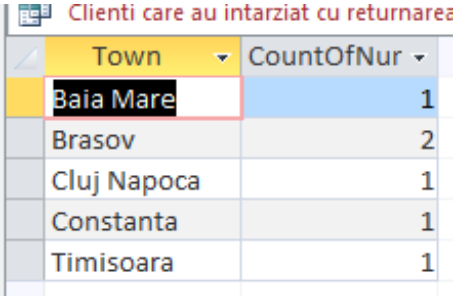


| Delay | Name |
|---|---|
| Yes | Maria |
| Yes | Diana |
| Yes | Alexandra |

*9. Who are the customers who delayed the return of the books?*

The code is presented in the next query.

```
      SELECT CustomersTransactions.Delay, Clients.Name
      FROM Clients INNER JOIN CustomersTransactions ON
Clients.[ClientCode] =
CustomersTransactions.[ClientCode]
      WHERE (((CustomersTransactions.Delay)="Yes"));
```



Clienti care au intarziat cu returnarea

| Town | CountOfNur |
|---|---|
| Baia Mare | 1 |
| Brasov | 2 |
| Cluj Napoca | 1 |
| Constanta | 1 |
| Timisoara | 1 |

*10. The number of customers in each city. In the next querry we filtered the data.*

```
      SELECT Clients.Town, Count(Clients.Name) AS
CountOfNume FROM Clients
      GROUP BY Clients.Town;
```

## Conclusions

The classic method of manually managing the bookstore organized within a company proved to be inefficient, following the survey conducted based on employee experience. Thus, a transition in the online environment of their activity related to the library is a step towards integration in the digital age in which we live.

The objective of this project is to develop a system that, by including modern techniques, allows the control and management of the activities that the bookstore involves.

The aim is also to create a design, an interface with the user who wants the tasks to be performed easily and productively, thus generating his satisfaction.

The implemented system helps the integrated system manager to be able to follow the loan process and to have access to updated data in real time (about the state of the stock and users), as well as control over them.

Employees have access to the articles made available to them in an organized format, and they are also facilitated to use the services offered by the digital library.

**REFERENCES**

Bourgeois, T.D., *Information Systems for Business and Beyond*, Saylor Publishing, USA, 2014.

Brown, E., Web *Development with Node and Express: Leveraging the JavaScript Stack*, O'reilly Media Publishing, USA, 2014.

Filipova, O., *Learning Vue.js 2, Packt Publishing*, Birmingham, UK, 2016.

*** Developer Survey Results 2019, https://insights.stackoverflow.com/survey/2019.

Parikshit, H., *Comparison - Centralized, Decentralized and Distributed Systems*, Working paper https://www.geeksforgeeks.org/comparison-centralized-decentralized-and-distributedsystems/, 2018.

https://dictionary.cambridge.org.

**NOTE ON THE AUTHORS**

**Loredana MOCEAN** has graduated Babes-Bolyai University of Cluj-Napoca, the Faculty of Computer Science, she holds a PhD diploma in Cybernetics and she had gone through didactic position of assistant and lecturer, since 2000 when she joined the staff of the Babes-Bolyai University of Cluj-Napoca, Faculty of Economics and Business Administration. She is associate professor. Ph.D., Babes-Bolyai University of Cluj-Napoca, Faculty of Economics and Business Administration, Cluj-Napoca. Also, she graduated Faculty of Economics and Business Administration. She is the author of more than 20 books and over 40 journal articles. She is member in more than 20 grants and research projects, national and international. Email*:* loredana.mocean@econ.ubbcluj.ro.

**Miranda Petronella VLAD** has graduated Babes-Bolyai University of Cluj-Napoca, the Faculty of Mathematics and Computer Science and she holds a PhD diploma in Cybernetics and Economic Statistics. She is associate professor. Ph.D., „Dimitrie Cantemir" Christian University Bucharest, Faculty of Economic Sciences, Cluj-Napoca. She is the author of numerous scientific papers and books in the field of computer science. Email: mirandavlad@gmail.com.