

CS480 - Course Project

Library Management

Boris Glavic

February 19, 2024

1 Project Timeline

The project has three deliverables. Deadlines are announced on the course webpage <https://www.cs.uic.edu/~bglavic/cs480/2024-spring/schedule/importantdates/>. This document can be downloaded from <https://www.cs.uic.edu/~bglavic/cs480/2024-spring/pdfs/project.pdf>. Each group will **demo** their application at the end of the semester. The deliverables are:

- **ER-model:** Each group should develop an ER-model for the application. This can be uploaded as any type of image file (please do not use esoteric formats).
- **Relational schema:** The second deliverable is a translation of the ER-model into a relational schema implemented as an SQL script. The script should use Oracle's SQL dialect and should execute without errors on the `fourier.cs.iit.edu` Oracle accounts. Besides from defining tables and constraints, this script should create indexes where appropriate. Please upload the script as a simple text file.
- **Application:** The last deliverable is a online grocery management application that uses the relational schema defined in the first two deliverables. This application can be either a web or desktop application.

Some of the requirements are marked as optional **bonus** requirements. You are free to not realize these requirements, but you can get extra credits by implementing them.

Every member of the group has to contribute in each phase of the project and you will be graded based on your individual contribution and on the overall project result.

2 Overview

The goal is to build a library management application. There will be two roles for the application: clients and librarians. Librarians can register and manage clients, register documents and delete documents from the library, and retrieve information about which books are lend out. Clients can search for documents in the library, lend out documents, and give back documents they are currently lending.

3 Data Requirements

3.1 Librarians

- For each librarian we record their **name**, their **ssn**, their **email**, and **salary**. Librarians are identified by their **ssn**.

3.2 Clients

- For each client we should record the **name** and **email address** of the client, one or more **addresses**, and **credit card** information. Clients are identified by their *email address*. A client can have multiple credit cards and for each credit card we associate it with a payment address (one of the client's addresses). Furthermore, we record which books are currently lend out to the client (see below).

3.3 Documents

The database should record information about documents in the library. This includes information about each document that depends on the type of document. There may be multiple copies of a document in the library and at each point in time each copy may be lend out to one client. For each copy that is lend out we need to record the date when the document was lend out. You cannot lend out more copies of a document than exist in the library. We will consider three types of documents:

- **Book:**
- **Magazine:**
- **Journal Article:**

Note that some information such as authors and publishers is shared across multiple documents of the same types and the different types of documents share some common information. This should be taken into account in the database design.

3.3.1 BONUS: Electronic Documents

- **Electronic Documents:** In addition to having paper copies of documents, a library may also have electronic copies. In contrast to paper documents, electronic copies of documents can be lend to any number of clients at the same time.

4 Application Requirements

The application should support the following actions for **librarians** and **clients**. Both clients and libraries have to log into the system with their email address and a password (created when the user is first registered with the system).

4.1 Librarians

4.1.1 Managing Documents

Librarians can manage documents:

- insert new documents
- update existing documents to fix errors, changing the number of copies the library owns, and deleting copies

Documents themselves cannot be deleted, but a library may at some point own 0 copies of a documents. Also copies of documents can only be deleted if they are currently not borrowed by a client.

4.1.2 Registration & Dropping Clients

Librarians can:

- register new clients
- update client information (such as address, name, and payment information)
- delete clients

4.2 Clients

4.2.1 Search for Documents

The main functionality is a search interface for documents. A client should be able to search for documents based on several types of information. Search should allow for combining multiple conditions with “OR” and “AND”. Several types of comparisons should be supported for the attributes of a document:

- **Equality:** tests whether that attribute’s value is equal to a given value
- **Contains:** tests whether the attribute’s string value contains the search term exactly
- **Search with placeholders:** tests whether the attribute matches a string with placeholders (see SQL’s LIKE in <https://www.postgresql.org/docs/16/functions-matching.html>).

The search interface should also support sorting the results based on a user-provided criterion and limiting the results to a maximum number of results. For each result you should also show the number of available copies or if no copy is currently available what is the earliest time a copy will become available.

4.2.2 Borrow a Document

The search interface should enable clients to borrow a document shown in a search result if there is at least one copy of the document available right now. The borrow period is 4 weeks.

4.2.3 Return a Documents

The client should be able to return a document they have borrowed. If the document is overdue, then a late fee of \$5 dollar per week should be added to the client’s account.

4.2.4 Paying Overdue Fees

Clients can pay overdue fees of their account using one of their payment methods. One of the existing payment methods of the client have to be used (see below).

4.2.5 Adding & Deleting Payment Methods

The client should be able to register payment methods (credit cards) for their account and update / delete existing payment methods.

4.2.6 BONUS: Full text search

In addition to searching based on conditions on individual attributes, the search interface should also support full text search where the user provides keywords that are matched against any of the attributes of a document. For that you may want to utilize full text indexes supported in Postgres (<https://www.postgresql.org/docs/current/textsearch.html>).