UCN psu0219

1st semester project

Group 13

Logo

DATABASES & TEST REPORT

29th May 2019

Andreas Richardsen

Zahro-Madalina Khaji

Contents

[Database 3](#_Toc9583224)

[Introduction 3](#_Toc9583225)

[GTL Requirements 3](#_Toc9583226)

[GTL Features 3](#_Toc9583227)

[Database Schema Design 4](#_Toc9583228)

[Entity-Relationship (ER) Conceptual Model and Diagram 4](#_Toc9583229)

[Table Design 8](#_Toc9583230)

[Query Design 8](#_Toc9583231)

[Important Indexes 8](#_Toc9583232)

[Databases & Security 8](#_Toc9583233)

[Applied Database Logic 8](#_Toc9583234)

[Discussion 8](#_Toc9583235)

[Conclusion 8](#_Toc9583236)

[Appendices 8](#_Toc9583237)

[Test 8](#_Toc9583238)

[Introduction 8](#_Toc9583239)

[GTL PID 9](#_Toc9583240)

[Designing a testable architecture 10](#_Toc9583241)

[Coding standards 10](#_Toc9583242)

[Scope & Test Strategy 10](#_Toc9583243)

[Risks Analysis 10](#_Toc9583244)

[Applied Test Techniques 11](#_Toc9583245)

[Test Plan 11](#_Toc9583246)

[Test Coverage 11](#_Toc9583247)

[Test Automation 11](#_Toc9583248)

[Test Management 11](#_Toc9583249)

[Test Control 11](#_Toc9583250)

[Discussion 11](#_Toc9583251)

[Conclusion 11](#_Toc9583252)

[Appendices 11](#_Toc9583253)

# Database

## Introduction

In the contents of this report, we are going to present the Georgia Tech Library (GTL) case from a databases perspective. We will employ our skills to design a useful database solution fitted for their needs, while also considering future expansions. We make arguments to justify important decisions that we faced during the project and weighted the pros and cons using the knowledge we obtained thought the course.

### GTL Requirements

The main task that the database solution in question needs to handle is managing their library loaning and keeping tracks of certain areas that are vital to such a system. We have carefully read the case, with the intention of getting more acquainted with the mini world that the system would model. From this we have identified the following requirements:

1. Library wants to track the number of book copies that are currently on loan and those that are not
2. Library wants a list of all the books and their descriptions, which can be filtered by book author, title and subject area
3. Regular library members can checkout books for 21 days, can loan out a maximum of five books at a time and have a one-week grace period for returning borrowed books, before a notice is sent to them
4. Librarians require a member’s SSN, campus and home address, and phone numbers to register them into the system
5. Each member gets a numbered library card with their photo on it, that is valid for four years after its issue, a month before the card expires, a notice is sent
6. Professors can check out books for three months and have a two-week grace period.
7. Library wants a list of books that cannot be lent such as reference books, rare books and maps.
8. Library want a list of books that they are interested in acquiring but cannot acquire.
9. Library uses ISBN to uniquely identify books
10. Library cooperates with other libraries and is interested in exposing statistics about them

### GTL Features

As shown by the many requirements we have gathered, the library would like a system that implements several features. However, because of time and budget constraints, we have narrowed down all the possible features that we noticed in this case and split them into essential features which we will be adding to our solution and discuss in this report and features that would be included at a later time. These are the features that we selected:

* Book loaning feature, which includes the business rules: certain books cannot be lend, max. number of books per member, that member has a valid card and more – we considered vital in the context of the system, is a complex task and will be our focus
* A view of all the books that are on loan and those that are not
* Access to specialized views such as books that cannot be loaned etc.

These are features that we will omit for the moment:

* Books they are interested in acquiring
* Sending notifications
* Exposing statistics

Determining the features that will be implemented and those that will not is an important step in database design as is clear on the scope that is in the project.

## Database Schema Design

This section is dedicated to describing the conceptual design our database schema in detail. To aid us in modelling the schema we have used a high-level conceptual model called ER Model. Furthermore, an EER Model has also been employed as it extends the capabilities of the ER Model.

### Entity-Relationship (ER) Conceptual Model and Diagram

With the features that we have decided to implement in mind, we started to work on our Entity-Relationship (ER) diagram, which is a tool used to model data requirements into a model that will be the basis of our database schema.

#### Entities

In our ER model, we have added the following entities: person, card, book and map.

The Person entity models the users of the system which could be library staff or regular members such as Students or professors. The key attribute of this entity is SSN attribute which is unique for every Person. The simple attributes are FName which stands for the Person’s first name , LName which stands for the Person’s last name and Type which stands for the type of user the Person could take. For the type attribute we have decided on the following values: Librarian, Student and Professor. The Person entity has a multiple valued attribute called Phone\_No. In the requirements it was specified that the library would like to store multiple phone numbers associated to a Person. The composite attribute called Address has the sub-attributes Home and Campus. It was also noted that the library would like to store a Person’s home address and campus address.

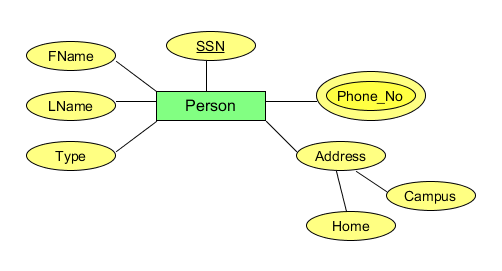


Figure 1: The Person entity in the ER diagram

The Card entity is used to model the card issued when a Person registers in the library system. The library assigns each card a unique card number which we designated as Card\_No as key attribute for the entity. We have added the simple Card\_Issue\_Date attribute as we wanted to make sure that the system does not loan books (feature) to instances of the Person entity that have an expired card. Per the library’s stated business rules, a card expires four years after its issue date.

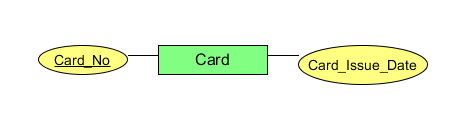


Figure 2: The Card entity in the ER digram

The Book entity is a central entity for our model. The key attribute is the ISBN of a book, which is what the GTL library uses to uniquely identity its books. The simple attributes of the Book entity are Title, Author, Description, Publishing, Year\_Publishing and Type. Except for the Type attribute all of them are self-explanatory. The Type stands for normal, reference or rare book. The reasoning for choosing these three types will be explained as we continue a bit further down. Book has two multiple-valued attributes called Subjects and Copies. We assumed each book would fall into one or more subject areas therefore we wanted to store multiple values for them. The Copies attribute is an interesting one. The library usually has multiple copies of a book at its disposal. For the loaning feature, we realized that we needed to keep track of the individual copies of books that can be loaned. To identify a copy of a book, the ISBN could not be used, so we decided to assign the copies a unique barcode. This allows us to easily keep track of the individual copies of the books. When it came to model the concept of the book copies, we had three options: to have the book copy as a separate normal entity, to store the copies as multi-valued attribute of the Book or as a weak entity dependent on the Book entity. The reason why we chose the first option is because unlike a typical weak entity the book copy has a barcode that can be used to uniquely identity it. In addition, the book copy is tightly related to a book so it could not stand on its own as an entity and we did not want to facilitate a situation where copies of a book that does not exist are stored in the system. Therefore, we chose to add it as a multiple-valued attribute of the Book entity.

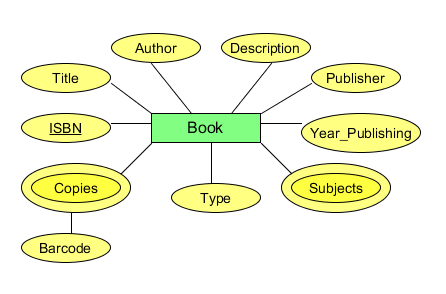


Figure 3: The Book entity in the ER digram

The Map entity is also included, as it is mentioned in the requirements. The key attribute is the ISBN, which is allowed because ISBNs can be assigned to maps as well.[isbn reference ] The entity has the following simple attributes Title, Author, Description, Publishing, Year\_Publishing and No\_Copies. The No\_Copies is added because we assumed that the library would like to store of the number of copies of each map it has. We are not concerned with tracking the individual copies of the maps in the system because the library does not loan out maps. We assumed that although maps are not loanable, the library would like to store information about them in the system.

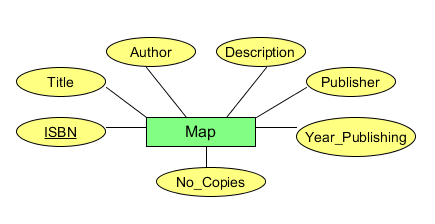


Figure 4: The Map entity in the ER diagram

#### Relationships

We have the following relationships between entities.

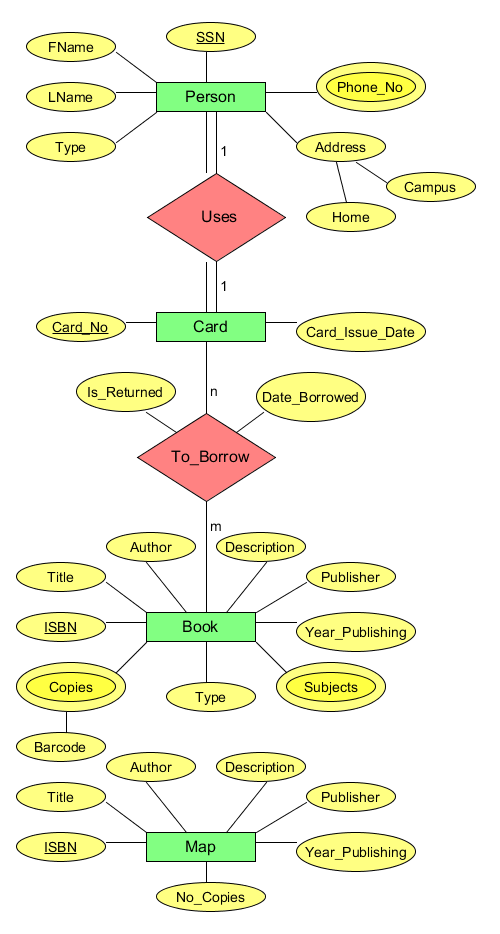


Figure 5: Entity-Relationship (ER) conceptual data model

#### EER diagram

What we added in EER and other

Figure 2: EER diagram for the GTL case

## Table Design

#### Relational model

How we mapped our EER into a Relational model and what rules we used

Figure 3: Relational model GTL Case

Normalization if any

(include picture after normalization)

#### Problems that arise

Data generation inserted not normal books into the Borrow because we had not set a constraint, the same copy of the same book has been loaned by two different persons on the same day. We did not think to add constraints to the Data generation require us to.

## Query Design

## Important Indexes

## Databases & Security

Only librarians can use the system. In a completed system, librarians would have to login and depending on their position they would have different privileges e.g. Chief can do everything, reference librarians can only look up books etc.

## Applied Database Logic

## Discussion

## Conclusion

## Appendices

# Test

## Introduction

In this report, we will discuss the Georgia Tech Library (GTL) project from a testing perspective. The starting point of this project is a document which describes the GTL case. In short, the library is interested in a system that will be used for managing their business. Throughout this project, we made several assumptions about the case, when information was missing. These assumptions are mentioned and explained in the continuation of the report. First, we assumed that the system the library wants is a new custom software solution. Therefore, we categorized this project as a software development project. Going forward, to the project start-up stage, we constructed a Project Initiation Document (PID). The purpose of a PID is to clearly define what the library, which is the customer in this case, should expect to receive from us, the suppliers of the system, at the end of the project.

### GTL PID

Business objective

* To modernize the library’s loaning system by switching to a digital solution

Project objective

* To implement a working prototype for a library loaning system that will keep track of the members, books and borrowing activity

Scope

* Boundaries
  + Member registration
  + Book management
  + Tracking of items status such as available or borrowed / on loan
  + Reliable item loaning functionality
  + Sending notifications according to member type
  + Enforcing the communicated business rules such as
    - Restricting the borrowing of certain items
    - Restricting the number of books, a member can loan
* Activities
  + Supplier will only provide the software of the discussed solution; additional software that may be required as a platform for the solution to run on is not provided;
  + Supplier will not provide any hardware components; that is considered the customers responsibility
  + Supplier will not be responsible for any security issues of the system
  + Supplier will not be taking on the task of maintaining the system, after the completion of the project
  + Supplier will not be tasked with training the library’s employees on how to use the system
* Deliverables
  + Software system consisting of a working prototype with the functionalities that were agreed upon by both parties
  + Documentation about the systems design, implementation, usability and other relevant features

Constraints

* The development lifecycle that will be used is the spiral model
* The development approach is an Agile one, based on SCRUM
* The proposed software system will run on the Windows 10 operating system
* The technology stack used is the .NET framework with the C# programming language

Authority

* The person responsible for representing the customers interests and tasked with approving the final product is one of the developers

Resources

* The projects cost is fixed
* The staff working on developing the system consists of two developers
* Time is two months to complete
* The hardware required to develop the system is provided by the supplier

Besides the specifics of the product that is to be delivered, the PID also describes the project regarding important constraints such as project scope, delivery time, the desired quality and the cost of the project. These factors were taken into consideration by us, when we chose our test strategy and test approach.

## Designing a testable architecture

Dependency injection

Inversion of control

Simple factory

Abstract factory

## Coding standards

Meaningful names, intention-revealing names, pronounceable name, searchable name, one word per concept (clean code references)

## Scope & Test Strategy

## Risks Analysis

Taking into consideration risks is a crucial step in the development of any system. Several risks have been identified by us over the course of this project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Risk | Description | Impact | Priority | Mitigation |
| 1 | Lack of testing experience | Team doesn’t have enough concrete experience with the process of thoroughly testing a software system | High | 1 | Team must rely on following best practices and standards instead of experience |
| 2 | Lack of testing tools knowledge | Team is unacquainted with the testing tools necessary to properly test the system | High | 4 | Team must allocate time resources to reduce knowledge gaps |
| 3 | Product owners not available | The actual product owners cannot be involved in the process which is an important factor in agile development | High | 2 | Team must assign one of its members to act as a product owner for the duration of the project |
| 4 | Lack of business case | No business case has been provided which makes it hard to figure out what the main objectives that the system should solve are | Medium | 5 | Team must resign to select some possible objectives from the description of the case |
| 5 | Lack of time | Team is assigned a fixed deadline to complete the project which may contribute to lower the quality of the system | Medium | 3 | Team must manage their time effectively and stick to the agreed project schedule without sacrificing standards for quality |

Quality plan – QA with testing

## Applied Test Techniques

Unit test – we follow the naming convention Unit under test, scenario, expected outcome[shown in picture x].

## Test Plan

## Test Coverage

## Test Automation

## Test Management

## Test Control

## Discussion

## Conclusion

## Appendices