UCN psu0219

1st semester project

Group 13

Logo

DATABASES & TEST REPORT

29th May 2019

Andreas Richardsen

Zahro-Madalina Khaji

Contents

[Database 3](#_Toc8808759)

[Introduction 3](#_Toc8808760)

[Database Schema Design 3](#_Toc8808761)

[Table Design 3](#_Toc8808762)

[Query Design 3](#_Toc8808763)

[Important Indexes 3](#_Toc8808764)

[Databases & Security 3](#_Toc8808765)

[Applied Database Logic 3](#_Toc8808766)

[Discussion 3](#_Toc8808767)

[Conclusion 3](#_Toc8808768)

[Appendices 3](#_Toc8808769)

[Test 3](#_Toc8808770)

[Introduction 3](#_Toc8808771)

[Architecture 3](#_Toc8808772)

[Coding standards 3](#_Toc8808773)

[Scope & Test Strategy 3](#_Toc8808774)

[Risks Analysis 3](#_Toc8808775)

[Applied Test Techniques 3](#_Toc8808776)

[Test Plan 3](#_Toc8808777)

[Test Coverage 3](#_Toc8808778)

[Test Automation 3](#_Toc8808779)

[Test Management 3](#_Toc8808780)

[Test Control 3](#_Toc8808781)

[Discussion 3](#_Toc8808782)

[Conclusion 3](#_Toc8808783)

[Appendices 3](#_Toc8808784)

# 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. 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 data 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

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, books they are interested in acquiring etc.

These are features that we will omit for the moment:

* Sending notifications
* Exposing statistics

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.

#### ER Diagram

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

We have the following special attributes such as multiple attributes on member and book entity…

We have the following relationships between entities.

Figure 1: ER diagram for the GTL case

#### EER diagram

What we added in EER and other

Figure 2: EER diagram for the GTL case

#### 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)

## Database Schema Design

## Table Design

## Query Design

## Important Indexes

## Databases & Security

## 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

## Test Plan

## Test Coverage

## Test Automation

## Test Management

## Test Control

## Discussion

## Conclusion

## Appendices