

# Final Project Report

## High School Book Tracking Application

ILLINOIS INSTITUTE OF TECHNOLOGY — CS 425, SPRING 2017

---

April, 2017

# Revision Sheet

Release No.	Date	Revision Desc.
<b>Rev. 0</b>	April 4, 2017	Initial project creation — structure implemented.
<b>Rev. 1</b>	April 14, 2017	Frontend work started, backend almost complete.
<b>Rev. 2</b>	April 16, 2017	Backend SQL 'routes' for 'Admin' user complete, frontend login and individual user pages started.
<b>Rev. 3</b>	April 24, 2017	First user page ('Admin') complete, backend almost fully implemented.
<b>Final Revision</b>	<b>April 25, 2017</b>	<b>Project complete.</b>

# Document Authorization Memorandum

I have carefully assessed the Final Project Report for the *High School Book Tracking Application*. This document has been completed in accordance with the requirements outlined of the assignment in Illinois Institute of Technology's CS 425, Spring 2017.

MANAGEMENT CERTIFICATION — Please check the appropriate statement.

☐ This document is accepted.

☐ This document is not accepted.

We fully accept the changes as needed improvements and authorize initiation of work to proceed. Based on our authority and judgement, the continued operation of this system is authorized.

---

MORGAN WILSON

---

April 25th, 2017

---

MUAYED WASIM

---

April 25th, 2017

---

METE MORRIS

---

April 25th, 2017

# Final Project Report

## Table of Contents

### **BACKGROUND**

- Purpose
- Scope
- System Overview
- Methodology
- Points of Contact

### **ANALYSIS & DESIGN**

- Database Design
- Data Characteristics and Categorization
  - Static Data
  - Dynamic data
- Design
- Impacts
- User Interface
  - Visualization

### **IMPLEMENTATION**

- Source of Input
- Medium & Device
- Recipients
- User Access
- Exception Handling
- Data Responsibilities
- Security

# 1.0 Background

## 1.1 Purpose

The purpose of the Final Project Report is to make the stakeholders requests' known and to address all of their needs through the functionalities, non-functionalities, and constraints of the created database application.

## 1.2 Scope

At the stakeholder's request, the functions built into the database will derive the database's use cases based on the data, which will be described in the software specifications. With the database's non-functional aspects in mind, accessibility, performance, and other miscellany will also be addressed in the software specifications. Lastly, the constraints of the database system will be outlined to ensure that the client's needs have been met thoroughly. With the database application, the database is made available to the stakeholder's potential user-base for informational and interactive usage with the database system. There will be system of traceability to is to ensure that all aspects of the database lead back to the stakeholder's request.

## 1.3 System Overview

A brief system overview description as a point of reference for the remainder of the document.

- Responsible organization: Morgan Wilson, Mete Morris, Muayed Wasim
- *High School Book Tracking Application*
- Database Management System
  - Utilization: manages a school's textbook system environment
  - General support system: provides general ADP or network support for a variety of users and applications
- Database Application System
  - Utilization: accessing, interacting, and modifying the database from a browser setting
- Database environment written in MySQL, database application written in HTML, Javascript, JQuery, and Node.js

## 1.4 Methodology

Project references included a set of requirements given to us based on the stakeholder's requests. The database was constructed to that set of requirements.

## 1.5 Points of Contact

Points of contact include Morgan Wilson ([mwilso20@hawk.iit.edu](mailto:mwilso20@hawk.iit.edu)), Mete Morris ([mmorri13@hawk.iit.edu](mailto:mmorri13@hawk.iit.edu)), Muayed Wasim ([mmohammed@hawk.iit.edu](mailto:mmohammed@hawk.iit.edu)). Direct contact by email is the method of communication.

## 2.0 Analysis & Design

### 2.1 Logical Database Design

As depicted in the graphic representation, the logical organization of the data and defined relationships have been outlined. There are a total of eight entities: Teacher, Student, Parent / Guardian, LibraryBook, Checked Out Book, Class, Semester, and Admin. Student, Parent / Guardian, and Teacher share similar attributes (characteristics) such as the nominal designations (First Name, Middle Name, and Last Name). The three entities also have in common a school-wide User Identification Number along with a password for security. The student entity is populated with more use cases than the other two entities, it holds a weak entity: 'CheckedOutBook,' which tracks what textbooks the student is currently borrowing, NumberOfClasses, which tracks the number of classes a student is currently enrolled in, the due date and date checked out of a book, and proprietary library book actions such as Return/CheckOut. Additionally, 'Parent / Guardian' is also a weak entity in its relationship to 'Student.' The entity class holds basic identification-related attributes where as Semester handles the organization of classes in the school system. The entity LibraryBook is the complete construction of a book in a library system and contains all actions necessary as well as a book Identification Number. Lastly, the Admin, short for Administrator, is a root-account responsible for overseeing additions to the system or necessary changes. Admin holds root-level access and should be used with the utmost caution.

### 2.2 Data Characteristics and Categorization

The data elements utilized by the system mostly exist on a referential basis. Few elements are subject to change, thus the database is mostly static until a few key dates, such as the beginning of each semester, or the enrollment status of each student.

#### 2.2.1 Static Data

Static data is defined as data primarily used for reference during operation and usually generated or updated in widely separated time frames, independent of normal software execution.

Specifically, our static data elements are the following: 'First Name,' 'Middle Name,' 'Last Name,' and 'Password' for all human entities that are updated on a need basis. 'ISBN Number,' 'Book Name,' 'Total Book Number,' 'Course Name,' 'Course Number,' and all attributes of the 'Semester' entity are static.

#### 2.2.1 Dynamic Data

Dynamic data includes all data to be updated and either input during normal execution or output.

Specifically, 'Number of Classes,' dependent on the 'Student' entity, 'Checked Out Books,' (integer-attribute) is dependent on a derived calculated as a weak entity under

'Student,' 'Checked Out Books.' Under 'Library / Book,' 'Number Checked Out' and 'Number Available' are also dependent on the status of the database.

## 2.3 Design

Based on the current implementation of the database, the design does not disallow any maximum of data elements, however, there are restraints as to what data types can be entered into the database. The user-entered data must match its location's type to ensure data validity and safety. These constraints will be implemented as checks before data is pushed to the main database.

Limitations may be placed on the data because of such factors as:

- Source of data input
- Devices used for input and output of data
- Recipients of the data
- Conversion processes, the converted form the input and output data will take
- How often the data will be updated

## 2.4 Impacts

The impact of the data requirements for the database management system constructed will allow a school system to better organize and keep track of its library books as well as class-assigned textbooks and maintain a high-level of accountability and order. In terms of data storage, the system will account for a responding growth in the number of entities in the system. For however many objects and entities there need exist in the system, that will be the size of the database and data. The expected growth of data will be dependent in a one-to-one relationship based on the said number of objects and entities that need to exist in the system.

## 2.5 User Interface

The database application was created in order to achieve user-friendliness. It uses a GUI, implemented in HTML and Javascript in order to connect with the database using JQuery calling Node.js functions. Forms are easy to fill in and buttons provide visible hints to available actions. List boxes are used to display requests that users have called so that information can be easily portrayed.

As an overview, the user interface is intuitive, provides clean separation of various components to facilitate simple modification and revision, and the configuration is maintained throughout organized pages in order to best serve each user's needs.

### 2.2.1 Visualization

Static data will need to be viewed time to time depending on the varying nature of the users' requests. The models of displaying static data are straightforward and organized.

Buttons and fields handle the input of information order to seamlessly gather required information from the database itself and display it in a meaningful sense to the user, all with minimal effort.

## 3.0 Implementation

### 3.1 Source of Input

The source of all data element will come from a tool to handle the processing of new users, new books / textbooks, and new classes derived some future semesters. The tool can be adapted to have each user fill his / her own information to reduce the load on the administrator to create and commit potentially hundreds of new entities. This system will transmit data to the database in order to be handled by the database processing functions.

Data Elements	Source	Interfaced with...	Frequency
<b>Entities</b>	Relational schema	Database	Dependent on requested action frequency
<b>Products</b>	Database	User	Dependent on user action
<b>Database</b>	User input / modification	User	Dependent on user action

### 3.2 Medium & Device

In short, the details and format of data to be input into the proposed database system is only to identify the entities in the database themselves. Attributes or characteristics such as a user's 'First Name' or 'User Identification Number' will be populated either by the system or by the input as specified by the said user. Other types of data, such as the 'Book Identification Number' are dependent on the system itself to generate, dependent on the state of the external system the database is managing (i.e. year, semester, number of textbooks).

The database is primarily used to manage a system in the outside world. Each data element will be identified as well as the processes used to identify them. The system relies heavily on the concept of user input, where each necessary data value will require a human entity to manually enter them. This way, the necessary information the data carries holds a significance easily identifiable and understood in the outside world. The process used to transmit this data will be built on a database management system. Data will be transmitted and received in a more human-readable manner therefore the medium of data is reliant on human intelligence and comprehension.

### 3.3 Recipients

The organization receiving output data will be the stakeholders. The creation of this system is so that they can process data in a more efficient manner while maintaining the real-world significance for each entity. They will be able to review reports or searches outputted by the database management system.



### 3.4 User Access

There will be three types of users. First, the 'Student' user that is allowed the powers as created in the relational schema. Specifically, the student will be able to modify and update his / her / their: 'First Name,' 'Last Name,' 'Middle Name,' 'Password.' Additionally, he / she / they will retain the power to 'Check Out Books' as well as 'Return them.' To manage books, he / she / they will send a 'Request' to the 'Library,' containing the store of books. The 'Parent' user is similar to the Student user except in the single change that a 'Parent' cannot handle a 'Book.' The second type of user is the 'Teacher.' Like the Student and Parent, he / she / they will be able to modify their nominal handlers ('First Name,' etc...). Unlike the Student, however, the Teacher is unable to 'Check Out Books,' instead, Teacher entities manage 'Classes' that 'Require Books.' The last type of user is the 'Admin.' The Admin holds all root powers to add, modify, and delete all other types of entities as well as manage the relational schema between them.

### 3.5 Exception Handling

Processes for handling errors or data incompatibilities or data faults will only be required in the formatting of input data. If users choose to enter data that is not in the correct format, systems of checking will ensure that no data is added or removed or modified from the database unless it meets all the requirements of the data.

Since user input is known to be unreliable, at every step of data access and modification, there are checks so that the user correctly inputs credentials to maintain correct user-level authorization and data security. Specifically, at login, the user will choose which type of user he / she is ('Student,' 'Teacher,' 'Parent,' and 'Admin'), then, from there all appropriate authorizations are handled by the specific user page.

### 3.6 Data Responsibilities

The organization that the stakeholders are a part of are fully responsible for the source of the data. The management as well as the input and output of data are taken care of by the database management system.

### 3.7 Security

There are three classifications of security to match the three types of users. Students are unable to modify 'sensitive' parts of the database, such as the existence of a 'Book' entity or the assignment of 'Classes,' 'Semesters,' or 'Years.' The student can only handle his / her / their nominal attributes as to retain for the modification, addition, and deletion processes to higher-level users that determine the real-world educational processes. Teachers are allowed to modify 'Classes,' however, are unable to modify 'Books,' since it is not necessary for their user-level to handle. Lastly, all power is held in the 'Admin' user that is able to add, modify, and delete all aspects of the database and database management system for uses that are only arise in dire need and have been thoroughly vetted for their potential impacts.