**Book Publication Management System**

***PROJECT PROPOSAL***

The goal is to create an in-house management system for a book publication firm to handle its day to day operations. This system provides a common platform to handle/manage information about the various aspects such as authors, books, manuscripts, vendors, printing orders, employees, etc.

**OBJECTIVE:**

The main objective of the project is to efficiently maintain and store information about the

stakeholders of a publishing system, and track the status of the books throughout the publishing

lifecycle.

**SCOPE:**

The application will be used by the employees of a publication firm, and it includes the following

modules:

● Department management

● Author management

● Book management

● Vendor management

NOTE: We are not considering the sales and after sale processes.

## ***PROJECT ENVIRONMENT:***

The DBMS used for this project is MySQL 5.7.14. The framework chosen to implement the front-end is .NET v4.5, and we have chosen to develop a Windows Forms Application using Microsoft Visual Studio 2015, which connects to MySQL through .NET connector, Connector/Net 6.5.7.

## ***HIGH LEVEL REQUIREMENTS***

### **Initial user roles**

1) Editor

2) Contract manager

3) HR manager

4) Vendor manager

### 

### **Initial user story descriptions**

1) As an Editor I want to login into the application.

2) As an Editor I want to view books assigned to me.

3) As an Editor I want to update the status of reviews.

4) As an Contract Manager I want to login into the application.

5) As a Contract Manager I want to add new contracts.

6) As a Contract Manager I want to update the contract information.

7) As a Contract Manager I want to assign books to Editors.

8) As an HR Manager I want to login into the application.

9) As an HR Manager I want to add new employees.

10) As an HR Manager I want to update records of employees.

11) As an Vendor Manager I want to login into the application.

12) As a Vendor Manager I want to add new printing vendors

13) As a Vendor Manager I want to update printing vendors.

14) As a Vendor Manager I want to add printing orders

15) As a Vendor Manager I want to update printing orders.

## ***HIGH LEVEL CONCEPTUAL DESIGN***

### **Entities:**

1) User

2) Employee

3) Designation (namely Editor, Contract Manager, HR Manager, Vendor Manager)

4) Department

5) Author

6) Book

7) Printing Vendor

8) Printing Order

**Relationships:**

1) Employee **works in** Department

2) Employee **has** Designation

3) Author **writes** Book

4) Employee(Editor) **edits** Book

5) Employee(Vendor Manager) **places** Printing Order

6) Printing Vendor **takes** Printing Order

## ***SPRINT 1***

Refined user stories for Sprint 1:

1. As an Editor, I want to login into the application:

The Editor should be registered in the application and hold an account with a username and password to login to the application and use it.

2. As an Editor I want to view books assigned to me:

Each editor in the publication house is assigned a book/books as soon as the contract is signed with the author. The Editor views a dashboard as soon as he logs in to check if any new books are assigned to him or to check the status of the books he is currently working on.

3. As an Editor I want to update the status of reviews:

As the editor progresses the review of each book, he has an option to update the status of the books displayed in his dashboard.

**CONCEPTUAL DESIGN**

Entity: **User** (weak entity)

Attributes:

user\_id

emp\_id

password

Entity: **Employee**

Attributes:

emp\_id

SSN

name [composite]

first\_name

last\_name

DOB

address [composite]

addr\_line\_1

addr\_line\_2

city

state

country

zip\_code

DOJ

dept\_id

designation

salary

Entity: **Department**

Attributes:

dept\_id

dept\_name

description

Entity: **Author**

Attributes:

author\_id

name [composite]

first\_name

last\_name

address [composite]

addr\_line\_1

addr\_line\_2

city

state

country

zip\_code

contract\_start\_date

contract\_end\_date

Entity: **Book**

Attributes:

book\_id

book\_name

author\_id [multivalued]

genre [multi-valued]

editor\_id

status

start\_date

end\_date

Relationship: **Employee** has **User** credentials

Cardinality: One to One

Participation:

Employee has partial participation

User has total participation

NOTE: Not all employees have login rights to the application. For example, a clerk need not have special access to the system since his/her role is quite generic. Hence Employee has partial participation.

Relationship: **Employee** belongs to **Department**

Attributes:

start\_date

Cardinality: Many to One

Participation:

Employee has total participation

Department has partial participation

Relationship: **Employee** reviews **Book**

Cardinality: One to Many

Participation:

Employee has partial participation

Book has partial participation

Relationship: **Author** writes **Book**

Cardinality: Many to Many

Participation:

Author has partial participation

Book has total participation

**LOGICAL DESIGN**

Table: **User**

Column:

user\_id

emp\_id [Foreign Key ; references emp\_id of Employee table]

Password

Table: **Employee**

Column:

emp\_id

SSN

first\_name

last\_name

DOB

addr\_line\_1

addr\_line\_2

city

state

country

zip\_code

phone

email

DOJ

dept\_id [foreign key ;references dept\_id of Department table]

designation

salary

Table: **Department**

Columns:

dept\_id

dept\_name

description

Table: **Author**

Column:

author\_id

first\_name

last\_name

Addr\_line\_1

addr\_line\_2

city

state

country

zip\_code

phone

email

contract\_start\_date

contract\_end\_date

Table: **Book**

Column:

book\_id

book\_name

editor\_id [foreign key ;references emp\_id of Employee table]

status

start\_date

end\_date

last\_updated

Table: **Book\_Author**

Columns:

book\_id [foreign key ;references book\_id of Book table]

author\_id [foreign key ;references author\_id of Author table]

Table: **Book\_Genre**

Columns:

book\_id [foreign key ;references book\_id of Book table]

genre

***SPRINT 2***

Refined user stories for Sprint 2:

1) As an HR Manager I want to login into the application:

The HR Manager should be registered in the application and hold an account with a username and password to login to the application and use it.

2) As an HR Manager I want to add new Employees:

The HR manager should initially add all the employees, namely, Editor, Contract Manager, Vendor Manager to the system.

3) As an HR Manager I want to view records of Employees:

The HR Manager should have an option to view all the employee records in the system.

4) As an HR Manager I want to modify records of Employees:

The HR Manager must be able to add/remove Employees at any point of time, and update their personal information.

5) As a Contract Manager I want to login into the application:

The Contract Manager should be registered in the application and hold an account with a username and password to login to the application and use it.

6) As a Contract Manager I want to add Books to the system:

The Contract Manager should add Books into the system.

7) As a Contract Manager I want to add Authors to the system:

The Contract Manager should add the Authors that he signed a Contract with.

8) As a Contract Manager I want to modify the Authors that are already present in the system:

The Contract Manager must be able to add/remove Authors at any point of time, and update their personal information.

9) As a Contract Manager I want to assign books to Editors:

It is the Contract Manager’s responsibility to initially assign each signed Book to an Editor. (The Editor should be present in the system)

**CONCEPTUAL DESIGN**

Entity: **User** (weak entity)

Attributes:

user\_id

emp\_id

password

Entity: **Employee**

Attributes:

emp\_id

SSN

name [composite]

first\_name

last\_name

DOB

address [composite]

city

state

country

zip\_code

DOJ

dept\_id

designation

salary

Entity: **Department**

Attributes:

dept\_id

dept\_name

description

Entity: **Author**

Attributes:

author\_id

name [composite]

first\_name

last\_name

address [composite]

addr\_line\_1

addr\_line\_2

city

state

country

zip\_code

Entity: **Book**

Attributes:

book\_id

book\_name

author\_id [multivalued]

genre [multi-valued]

editor\_id

status

start\_date

end\_date

Entity: **Contract**

Attributes:

contract\_id

emp\_id

author\_id

contract\_start\_date

contract\_end\_date

Relationship: **Employee** has **User** credentials

Cardinality: One to One

Participation:

Employee has partial participation

User has total participation

NOTE: Not all employees have login rights to the application. For example, a clerk need not have special access to the system since his/her role is quite generic. Hence Employee has partial participation.

Relationship: **Employee** belongs to **Department**

Attributes:

start\_date

Cardinality: Many to One

Participation:

Employee has total participation

Department has partial participation

Relationship: **Employee** edits **Book**

Cardinality: One to Many

Participation:

Employee has partial participation

Book has partial participation

Relationship: **Author** writes **Book**

Cardinality: Many to Many

Participation:

Author has partial participation

Book has total participation

Relationship: **Author** has **Contract**

Cardinality: One to Many

Participation:

Author has partial participation

Contract has total participation

**LOGICAL DESIGN**

Table: **User**

Column:

user\_id

emp\_id [Foreign Key ; references emp\_id of Employee\_PersonalDetails table]

Password

Highest normalization level:4NF

Table: **Employee\_PersonalDetails**  
Column:  
 emp\_id  
 SSN  
 first\_name  
 last\_name  
 DOB  
 addr\_line\_1  
 addr\_line\_2  
 city  
 state  
 country  
 zip\_code  
 phone  
 email

Highest normalization level:4NF

Table: **Employee\_OrganizationalDetails**

Columns:

emp\_id [foreign key ;references emp\_id of Employee\_PersonalDetails table]

DOJ  
 dept\_id [foreign key ;references dept\_id of Department table]  
 designation  
 Salary

Highest normalization level:4NF

Table: **Department**

Columns:

dept\_id

dept\_name

description

Highest normalization level:4NF

Table: **Author\_PersonalDetails**

Column:

author\_id

first\_name

last\_name

Addr\_line\_1

addr\_line\_2

city

state

country

zip\_code

phone

email

Highest normalization level:4NF

Table: **Author\_Contract**

Columns:

contract\_id

emp\_id[foreign key ;references emp\_id of Employee\_PersonalDetails table]

author\_id[foreign key ;references author\_id of Author\_PersonalDetails table]

contract\_start\_date

contract\_end\_date

Highest normalization level:4NF

Table: **Book**

Column:

book\_id

book\_name

status

start\_date

end\_date

last\_updated

Highest normalization level:4NF

Table: **Book\_Author**

Columns:

book\_id [foreign key ;references book\_id of Book table]

author\_id [foreign key ;references author\_id of Author\_PersonalDetails table]

Highest normalization level:4NF

Table: **Book\_Genre**

Columns:

book\_id [foreign key ;references book\_id of Book table]

genre

Highest normalization level:4NF

Table: **Book\_Editor**

Column:

book\_id [foreign key ;references book\_id of Book table]

editor\_id [foreign key ;references emp\_id of Employee\_PersonalDetails table]

***SPRINT 3***

Refined user stories for sprint 3:

1. As a Vendor Manager I want to login into the application:

The HR Manager should be registered in the application and hold an account with a username and password to login to the application and use it.

1. As a Vendor Manager I want to add new Printing Vendors:

The Vendor Manager must be able to insert new records of Printing Vendors to the system.

1. As a Vendor Manager I want to update Printing Vendors:

The Vendor Manager must be able to edit existing records of Printing Vendors.

1. As a Vendor Manager I want to view the Books ready to print:

The Vendor Manager must be able to view on his dashboard the Books that are ready to print.

1. As a Vendor Manager I want to add Printing Orders:

The Vendor Manager must be able add new records of Printing Order to the system.

1. As a Vendor Manager I want to update Printing Orders:

The Vendor Manager must be able to edit existing records of Printing Orders.

**CONCEPTUAL DESIGN**

Entity: **User** (weak entity)

Attributes:

emp\_id

password

Entity: **Employee**

Attributes:

emp\_id

SSN

name [composite]

first\_name

last\_name

DOB

email

contact\_number

address [composite]

city

state

country

zip\_code

DOJ

designation

salary

Entity: **Department**

Attributes:

dept\_id

dept\_name

description

Entity: **Author**

Attributes:

author\_id

name [composite]

first\_name

Last\_name

email

contact\_number

address [composite]

addr\_line\_1

addr\_line\_2

city

state

country

zip\_code

Entity: **Book**

Attributes:

book\_id

book\_name

genre [multi-valued]

status

start\_date

end\_date

Entity: **Contract**

Attributes:

contract\_id

contract\_start\_date

contract\_end\_date

Entity: **PrintingVendor**

Attributes:

vendor\_id

vendor\_name

email

contact\_number

address [composite]

addr\_line\_1

addr\_line\_2

city

state

country

zip\_code

Entity: **PrintingOrder**

Attributes:

order\_id

order\_date

quantity

order\_status

last\_updated

Relationship: **Employee** has **User** credentials

Cardinality: One to One

Participation:

Employee has partial participation

User has total participation

NOTE: Not all employees have login rights to the application. For example, a clerk need not have special access to the system since his/her role is quite generic. Hence Employee has partial participation.

Relationship: **Employee** belongs to **Department**

Attributes:

start\_date

Cardinality: Many to One

Participation:

Employee has total participation

Department has partial participation

Relationship: **Editor(Employee)** edits **Book**

Cardinality: One to Many

Participation:

Employee has partial participation

Book has partial participation

Relationship: **Author** writes **Book**

Cardinality: Many to Many

Participation:

Author has partial participation

Book has total participation

Relationship: **Author** has **Contract**

Cardinality: One to Many

Participation:

Author has partial participation

Contract has total participation

Relationship: **PrintingVendor** receives **PrintOrder**

Cardinality: One to Many

Participation:

PrintingVendor has partial participation

PrintOrder has total participation

Relationship: **PrintOrder** placed for **Book**

Cardinality: Many to One

Participation:

Book has partial participation

PrintOrder has total participation

**LOGICAL DESIGN**

Table: **User**

Column:

emp\_id [Foreign Key ; references emp\_id of Employee\_PersonalDetails table]

Password

Highest normalization level:4NF

Indexes:

Index 1: clustered

Columns: emp\_id

Justification: Records mostly searched by this column

Table: **Employee\_PersonalDetails**  
Column:  
 emp\_id  
 SSN  
 first\_name  
 last\_name  
 DOB  
 addr\_line\_1  
 addr\_line\_2  
 city  
 state  
 country  
 zip\_code  
 phone  
 email

Highest normalization level:2NF

zip\_code -> city There are dependencies in non-key attributes

Indexes:

Index 1: clustered

Columns: emp\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: SSN

Justification: Unique record and alternative to emp\_id to search

Index 3: non-clustered

Columns: last\_name

Justification: Records mostly searched by this column; searched by the last\_name

Table: **Employee\_OrganizationalDetails**

Columns:

emp\_id [foreign key ;references emp\_id of Employee\_PersonalDetails table]

DOJ  
 dept\_id [foreign key ;references dept\_id of Department table]  
 designation  
 Salary

Highest normalization level: 4NF

Indexes:

Index 1: clustered

Columns: emp\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: dept\_id

Justification: Records mostly searched by this column also

Table: **Department**

Columns:

dept\_id

dept\_name

description

Highest normalization level:4NF

Indexes:

Index 1: clustered

Columns: dept\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: dept\_name

Justification: Records mostly searched by this column also; searched by the name of the department

Table: **Author\_PersonalDetails**

Column:

author\_id

first\_name

last\_name

Addr\_line\_1

addr\_line\_2

city

state

country

zip\_code

phone

email

Highest normalization level:2NF

zip\_code -> city There are dependencies in non-key attributes

Indexes:

Index 1: clustered

Columns: author\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: last\_name

Justification: Records mostly searched by this column also; searched by the last name of the Author

Table: **Author\_Contract**

Columns:

contract\_id

contract\_manager\_id[foreign key ;references emp\_id of Employee\_PersonalDetails table]

author\_id[foreign key ;references author\_id of Author\_PersonalDetails table]

contract\_start\_date

contract\_end\_date

Highest normalization level:4NF

Indexes:

Index 1: clustered

Columns: contract\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: contract\_manager\_id

Justification: Records mostly searched by this column also; contract\_manager\_id is also used a lot to search

Index 3: non-clustered

Columns: author\_id

Justification: Records mostly searched by this column also;

Table: **Book**

Column:

book\_id

book\_name

status

start\_date

end\_date

last\_updated

Highest normalization level:4NF

Indexes:

Index 1: clustered

Columns: book\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: book\_name

Justification: Records mostly searched by this column also; book\_name is often used to search

Table: **Book\_Author**

Columns:

book\_id [foreign key ;references book\_id of Book table]

author\_id [foreign key ;references author\_id of Author\_PersonalDetails table]

Highest normalization level:4NF

Indexes:

Index 1: non-clustered

Columns: book\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: author\_id

Justification: Records mostly searched by this column also

Table: **Book\_Genre**

Columns:

book\_id [foreign key ;references book\_id of Book table]

genre

Highest normalization level:4NF

Indexes:

Index 1: non-clustered

Columns: book\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: genre

Justification: Records mostly searched by this column also

Table: **Book\_Editor**

Column:

book\_id [foreign key ;references book\_id of Book table]

editor\_id [foreign key ;references emp\_id of Employee\_PersonalDetails table]

Highest normalization level:4NF

Indexes:

Index 1: non-clustered

Columns: book\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: editor\_id

Justification: Records mostly searched by this column also

Table: **PrintingVendor**

Column:

vendor\_id

vendor\_name

email

contact\_number

addr\_line\_1

addr\_line\_2

city

state

country

zip\_code

Highest normalization level:2NF

zip\_code -> city There are dependencies in non-key attributes

Indexes:

Index 1: clustered

Columns: vendor\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: vendor\_name

Justification: Records mostly searched by this column also; vendor\_name is used quite often to search

Table: **PrintingOrder**

Column:

order\_id

vendor\_id [foreign key ;references vendor\_id of Vendor table]

book\_id [foreign key ;references book\_id of Book table]

order\_date

quantity

order\_status

last\_updated

Highest normalization level:4NF

Indexes:

Index 1: clustered

Columns: order\_id

Justification: Records mostly searched by this column

Index 2: non-clustered

Columns: vendor\_id

Justification: Records mostly searched by this column also

Index 3: non-clustered

Columns: book\_id

Justification: Records mostly searched by this column also

### **Stored programs**

**Stored procedure**: <get\_employee\_details>

Parameters: <IN employee\_id>, <OUT first\_name>,

<OUT last\_name>, <OUT designation>, <OUT department>

Goal: <Fetches all employee details to display on the dashboard from Employee\_PersonalDetails and Employee\_OrganizationalDetails>

**Trigger**: <Before Insert> on <Book>

Goal: <Update the last\_updated attribute with current timestamp>

**Trigger**: <Before Insert> on <PrintOrder>

Goal: <Update the last\_updated attribute with current timestamp and the status of the Book for which order is placed>

**Trigger**: <Before Update> on <Book>

Goal: <Update the last\_updated attribute with current timestamp>

**Trigger**: <Before Update> on <PrintOrder>

Goal: <Update the last\_updated attribute with current timestamp>