

# Project Report

## Cole Douglas - 100988503

### “Bouquet of Proses”

Due Date: December 19<sup>th</sup> 11:59

Github URL:  
[https://github.com/ColeDouglas/Comp3005\\_Project](https://github.com/ColeDouglas/Comp3005_Project)

#### Foreword

I didn't do anything extra.  
There is a checklist of completed components in the appendices.  
I didn't factor in the bonus 20% for being alone.

TBD:  
architecture description  
relation normalization  
pg numbers

# PROJECT REPORT

## Table of Contents

Pg	
3	Conceptual Design
4	Assumptions
5	Reduction to Relation Schemas
	Normalization of Relation Schemas
	Database Schema Diagram
	Implementation
	Appendix A: Availability
	Appendix B: Checklist
	Appendix C: Unsolved Problems
	Appendix D: Screen Shots

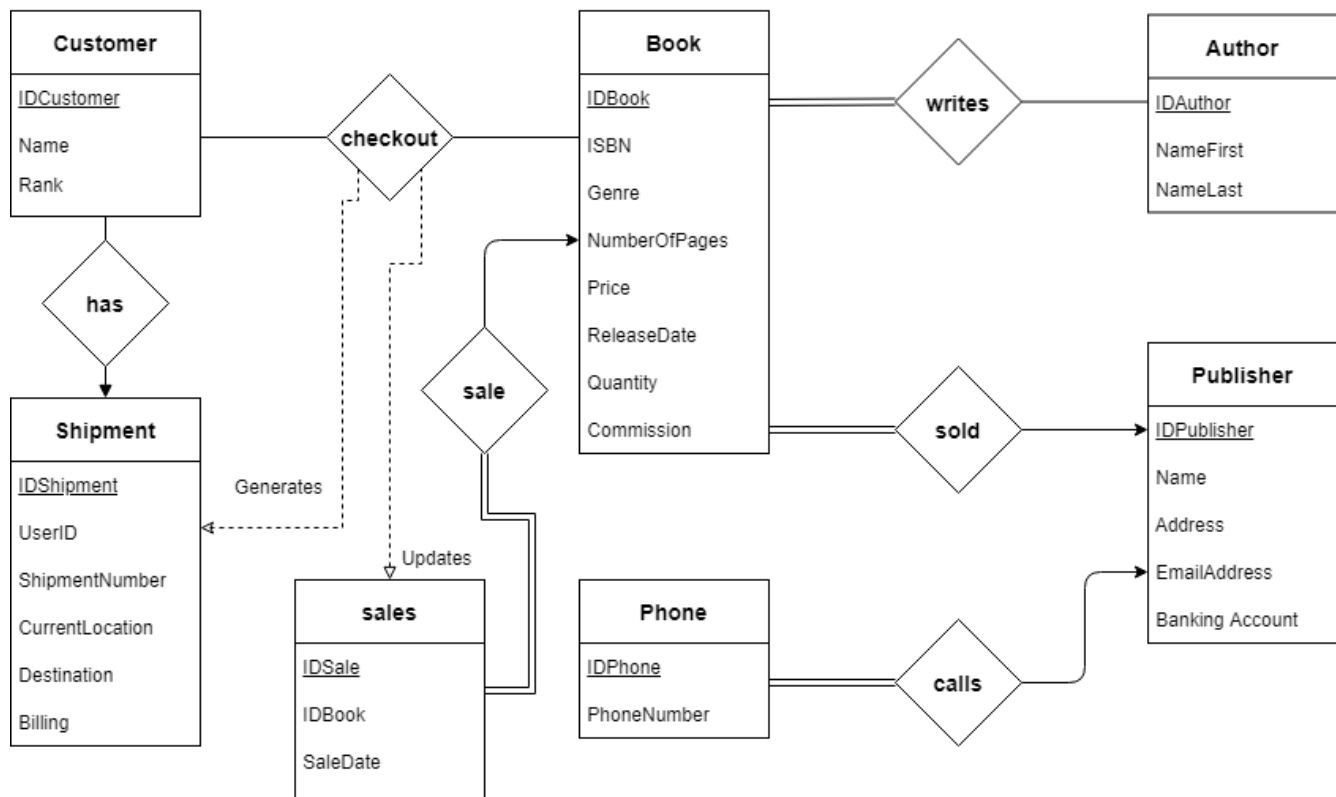
## PROJECT REPORT

### Conceptual Design

This section should explain the conceptual design of the database. That is, the ER-diagram of the database for the bookstore and explanation of all the assumptions made in the diagram regarding cardinalities and participation types. Make sure that the assumptions do not contradict with the problem statement in Section 1.

#### Bouquet of Proses

ER diagram



## PROJECT REPORT

### Assumptions

Each phone number is for only one publisher. The database allows otherwise, but the implementation prohibits it.

Each book has only one genre. The database does allow for more than one publisher, but I haven't coded certain functionality appropriately for that.

There are at least three types of users {owner, guest, customer} but the system could expand for more.

Each username is for only one user. The database allows otherwise, but the implementation prohibits it.

I assume garbage collection for irrelevant data will be implemented in the future. I have given each relation a distinct ID variable, so garbage removal is important. The data is all properly secured, so it's just a matter of deciding the expiration date and removal method.

I assume that the checkout data will be held for different times based on whether or not an order was placed.

## PROJECT REPORT

### Reduction to Relation Schemas

Reduce your ER-diagram into relation schemas and list these in this section.

*Book* ( IDBook, Name, ISBN, Genre, NumberOfPages, Price, ReleaseDate, Quantity, Commission )

*Author* ( IDAuthor, NameFirst, NameLast )

*writes* ( IDBook, IDAuthor )

*Publisher* ( IDPublisher, Name, Address, EmailAddress, BankingAccount )

*sold* ( IDBook, IDPublisher )

*sales* ( IdSale, IDBook, SaleDate )

*Phone* ( IDPhone, PhoneNumber )

*calls* ( IDPublisher, IDPhone )

*Customer* ( IDCustomer, Name )

*Shipment* ( IDShipment, CustomerID, ShipmentNumber, CurrentLocation, Destination, Billing )

*has* ( IDCustomer, IDShipment )

*Checkout* ( IDCustomer, IDBook, ShipmentNumber )

## PROJECT REPORT

### Normalization of Relation Schemas

Given the problem statement and your design, write the set of functional dependencies that apply to your database. Show that your relation schemas are either in a good normal form (show tests), or if they are not, show how to decompose them into a good normal form (show decomposition work), then show the testing work to make sure that they are in a good normal form.

TBD

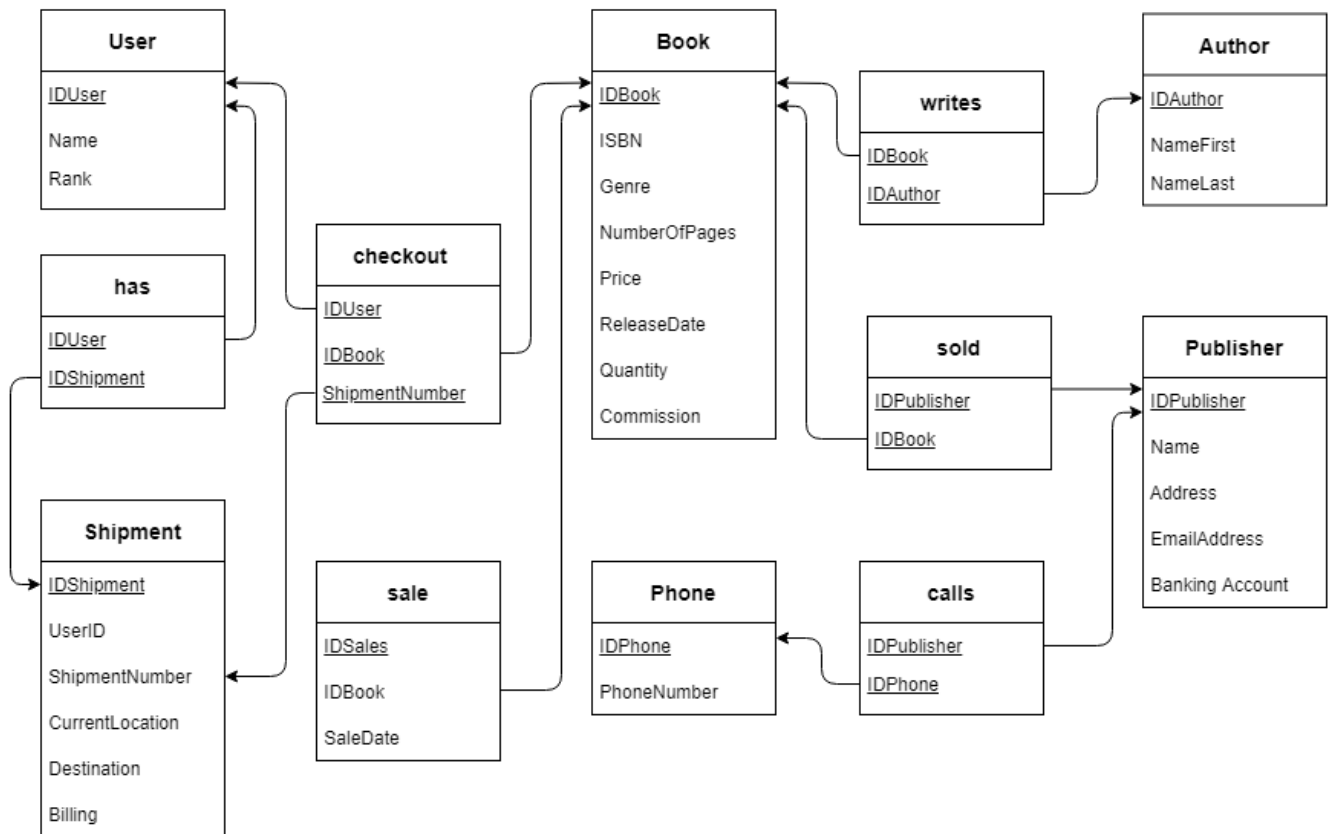
## PROJECT REPORT

### Database Schema Diagram

This section should show the final schema diagram for the database of the bookstore. This diagram should be similar to the schema diagram of the university database that we study in this course.

#### Bouquet of Proses

Schema Diagram



## PROJECT REPORT

### Implementation

Feel free to use whichever programming language(s) for your application. Your application can be a web-based application, or a desktop application. In this section, you should describe the architecture of your application. That is, what the modules in your application are and how they interact. You are encouraged to include a diagram of the application's architecture and explain (in text) scenarios of using the application and the workflow of your application. It is mandatory that you use a relational database to store all your data (e.g., no key-value stores).

Your application should have two different user interfaces: The first is for the users of the application through which the user can browse and buy books. The second is for the bookstore owners/managers through which they can add/remove books, display reports, etc. (more details about the application's features can be found in Section 1).

Include screenshots of your application's two interfaces in different scenarios (e.g., checking out, displaying a report, etc.).

I made a command line desktop java program.

View: the Users touch point.

- What the user sees and talks to

Control: the way the app handles user input

Main: the starting point

PostgresHandler:

All the interactions with the server

Consts: environment variables for each setup



## Appendices

# PROJECT REPORT

## Appendix A: Availability

December 20 <sup>th</sup>							
9 am - 12 pm			12 pm - 3 pm			3 pm - 5 pm	
Time	Available		Time	Available		Time	Available
9:00	Yes		12:00	Yes		3:00	Yes
9:20	Yes		12:20	Yes		3:20	Yes
9:40	Yes		12:40	Yes		3:40	Yes
10:00	Yes		1:00	Yes		4:00	Yes
10:20	Yes		1:20	Yes		4:20	Nice
10:40	Yes		1:40	Yes		4:40	No
11:00	Yes		2:00	Yes			
11:20	Yes		2:20	Yes			
11:40	Yes		2:40	Yes			

Appendix B:  
Checklist

**23.75 / 34 = ~ 67%**

Functionality			
This application lets users browse a collection of books that are available in the bookstore			✓
A user can search the bookstore by:			✓
	book name	✓	
	author name	✓	
	ISBN	✓	
	genre	✓	
	Etc.	✓	
A user can select as many books as they like to be added to the checkout basket			✓
A user needs to be registered in the bookstore to be able to checkout			✓
When checking out, the user inserts billing and shipping information (can be different than those used in registration), and completes the order			✓
The bookstore has the feature of tracking an order via an order number.			✓
A user can use this order number to track where the order is currently			✓
Although shipping is carried out by a third-party shipping service, the online bookstore should have the tracking information available for when the user inquires about an order using the order number.			✓
Assume all books are shipped from only one warehouse (no multiple order numbers for multiple books shipped from multiple warehouses).			✓
The bookstore owners can add new books to their collections, or remove books from their store.			✓
They also need to store information on the publishers of books such as			✓
	name,	✓	
	address,	✓	
	email address,	✓	
	phone number(s),	✓	
	banking account,	✓	
	etc.	✓	
The banking account for publishers is used to transfer a percentage of the sales of books published by these publishers.			✓

## PROJECT REPORT

This percentage is variable and changes from one book to another.			✓
The owners should have access to reports that show			¾
	sales vs. expenditures,	X	
	sales per genres,	✓	
	sales per author,	✓	
	etc.	✓	
The application should also be able to automatically place orders for new books if the remaining quantity is less than a given threshold (e.g., 10 books).			✓
This is done by sending an email to the publisher of the limited books to order a number of books equal to how many books were sold in the previous month (you do not have to implement the email sending component).			X
			14.75 / 16

Conceptual Design	
This section should explain the conceptual design of the database.	✓
Make sure that the assumptions do not contradict with the problem statement in Section 1.	✓
	2 / 2

Reduction to Relation Schemas	
Reduce your ER-diagram into relation schemas and list these in this section.	✓
	1 / 1

Normalization of Relation Schemas	
Given the problem statement and your design, write the set of functional dependencies that apply to your database.	
Show that your relation schemas are either in a good normal form (show tests), or if they are not, show how to decompose them into a good normal form (show decomposition work)	
Show the testing work to make sure that they are in a good normal form.	
	0 / 3

## PROJECT REPORT

Database Schema Diagram		
This section should show the final schema diagram for the database of the bookstore.		✓
		1 / 1

Implementation			
Describe the architecture of your application.			
Include a diagram of the application's architecture and explain (in text) scenarios of using the application and the workflow of your application.			
It is mandatory that you use a relational database to store all your data (e.g., no key-value stores).			✓
Your application should have two different user interfaces:			✓
	The first is for the users of the application through which the user can browse/buy books.	✓	
	The second is for the bookstore owners/managers through which they can add/remove books, display reports, etc. (more details about the application's features can be found in Section 1).	✓	
Include screenshots of your application's two interfaces in different scenarios (e.g., checking out, displaying report, etc.).			✓
			3 / 5

GitHub Repository		
All your source code for your application should be uploaded to a public GitHub repository.		✓
The code needs to be well-documented and a decent README file that clearly states the instructions for running your code should be provided.		✓
The GitHub repository should also include a directory titled "SQL" that includes all the SQL DDL statements and SQL queries used in your application		✓
This section (of the report) should include the url to this GitHub repository.		✓
Make sure that your GitHub repository is public.		✓
🔗 <a href="https://github.com/ColeDouglas/Comp3005_Project">https://github.com/ColeDouglas/Comp3005_Project</a>		5 / 5

Appendix A		
This section should include your availability (from 9 am to 5 pm) for a 20 minutes demonstration of your work on December 20th.		✓
		1 / 1

## PROJECT REPORT

### Appendix C: Unsolved Problems

#### Shipment Removal:

There is no method for a shipment to be deleted without completing the delivery.

It must be manually deleted by the owner.

The assumption is that the owner is deleting the shipment to sign off on completion of delivery.

As such, manually deleting a shipment doesn't lower the sales of that book, or raise the quantity of that book in stock.

However there are several situations where an owner would want to delete a shipment without declaring it complete. This project doesn't cover those situations.

#### Shipment ID:

There isn't really a need for both IDShipment and ShipmentNumber. I was planning something cool there, but didn't really put the work in to justify it.

As such, the "has" relation is redundant and should have been removed.

#### Redundancy:

I planned to have had stored the IDBooks in a variable fields in Shipment, then deleted the checkout relations after. Ran out of time for that

#### Authors:

There is no way to add or remove authors from books

Creating a book only allows you to assign one author.

So the only books that have multiple authors are the ones inserted directly into the server.

#### Publishers:

There is no way to add or remove publishers from books

Creating a book only allows you to assign one publisher.

So the only books that have multiple publishers are the ones inserted directly into the server.

#### Phone Numbers:

There is no way to add or remove phone numbers from publishers.

## PROJECT REPORT

Creating a publisher only allows you to assign one phone number.

So the only publishers that have multiple phone numbers are the ones inserted directly into the server.

Privacy:

Any customer can check on any shipment as long as they have the shipment number. I started to implement security measures but ran out of time.

Concurrency:

This entire system could be brought to a grinding halt by two people using it at the same time.

Quantity:

You can only order one copy of a book at a time. Multiple copies needs multiple orders.

Social Issues:

You can make an order with zero books. This is logical, but still feels like it should be denied. Etc.

# PROJECT REPORT

## Appendix C: Screen Shots

### User Interface

```
>> Welcome to
>> A Bouquet of Proses

>> Would you like to login?

>> (a) Login
>> (b) Register
>> (z) Exit
>

>> Please input your numeric UserID
>> or press z to exit
>> Hint: Owner is 0, Guest is 1, Registered users are +2
>

!! WARNING !!
>> YOU ARE LOGGING IS AS A GUEST
>> AND WILL NOT HAVE FULL FUNCTIONALITY
>> Successfully Logged in as user guest

>> What would you like to do?

>> (a) Look at all books
>> (b) Search for a book
>> (z) Log out
|
```

Unregistered user logging in as guest

```
>> Welcome to
>> A Bouquet of Proses

>> Would you like to login?

>> (a) Login
>> (b) Register
>> (z) Exit
>

>> Please input your numeric UserID
>> or press z to exit
>> Hint: Owner is 0, Guest is 1, Registered users are +2
>

>> Successfully logged in as user 2

>> What would you like to do?

>> (a) Look at all books
>> (b) Search for a book
>> (c) See my orders
>> (d) Check the status of an order
>> (e) Create an order
>> (z) Log out
|
```

Registered user logging in

```
>> Successfully logged in as user 2

>> What would you like to do?

>> (a) Look at all books
>> (b) Search for a book
>> (c) See my orders
>> (d) Check the status of an order
>> (e) Create an order
>> (z) Log out
>

Customer: 2
Ordered: A Slip of the Keyboard
Shipment: 1

Ordered: Good Omens
Shipment: 1

Ordered: Going Postal
Shipment: 1

Ordered: A Slip of the Keyboard
Shipment: 2

>> What would you like to do?

>> (a) Look at all books
>> (b) Search for a book
>> (c) See my orders
>> (d) Check the status of an order
>> (e) Create an order
>> (z) Log out
|
```

User checking their orders

```
>> What would you like to do?

>> (a) See current cart
>> (b) Add to cart
>> (c) Finish order
>> (d) Cancel order
>

>> What is the Book ID you want to add?
>

>> What would you like to do?

>> (a) See current cart
>> (b) Add to cart
>> (c) Finish order
>> (d) Cancel order
>

Current Cart:
Book : Going Postal
Book : Good Omens

>> What would you like to do?

>> (a) See current cart
>> (b) Add to cart
>> (c) Finish order
>> (d) Cancel order
|
```

User making a new order



# PROJECT REPORT

## Owner Interface

```
>> Welcome to
>> A Bouquet of Proses

>> Would you like to login?

>> (a) Login
>> (b) Register
>> (z) Exit

>> Please input your numeric UserID
>> or press z to exit
>> Hint: Owner is 0, Guest is 1, Registered users are +2

>> Successfully logged in as owner

>> What would you like to do?

>> (a) Manage Users
>> (b) Manage Books
>> (c) Manage Authors
>> (d) Manage Publishers
>> (e) Manage Orders
>> (f) See Sales Reports
>> (z) Log out
|
```

The Owner logging in

```
>> Successfully logged in as owner

>> What would you like to do?

>> (a) Manage Users
>> (b) Manage Books
>> (c) Manage Authors
>> (d) Manage Publishers
>> (e) Manage Orders
>> (f) See Sales Reports
>> (z) Log out

>> What would you like to do?

>> (a) See all books
>> (b) Search for a book
>> (c) Remove a book
>> (d) Create a book
>> (e) Order more of an existing book
>> (z) Go Back

>> What is the ID of the Book being removed?
|
```

The Owner removing a Book

```
>> What would you like to do?

>> (a) See Sales Per Genre
>> (b) See Sales Per Author
>> (c) See Sales Per Publishers
>> (z) Go Back

Publisher Name: Vertigo
Sales Amount: 1

Publisher Name: HarperTorch
Sales Amount: 3

Publisher Name: Beacon Press
Sales Amount: 1

Publisher Name: WilliamMorrow
Sales Amount: 10

Publisher Name: RandomHouse
Sales Amount: 7

Publisher Name: Penguin Books
Sales Amount: 3

>> What would you like to do?

>> (a) See Sales Per Genre
>> (b) See Sales Per Author
>> (c) See Sales Per Publishers
>> (z) Go Back
|
```

The Owner looking at a sales report

```
>> What would you like to do?

>> (a) See Orders
>> (b) Get all orders for one customer
>> (c) Get one order
>> (d) Manually delete an order
>> (z) Go Back

Shipment Number: 2
Ordered by: 2
Destination: Somewhere else
Number of items: 4

Shipment Number: 1
Ordered by: 2
Destination: Somewhere else
Number of items: 4

>> What would you like to do?

>> (a) See Orders
>> (b) Get all orders for one customer
>> (c) Get one order
>> (d) Manually delete an order
>> (z) Go Back
|
```

The owner looking at current shipments