Comp 3005 Fall 2022

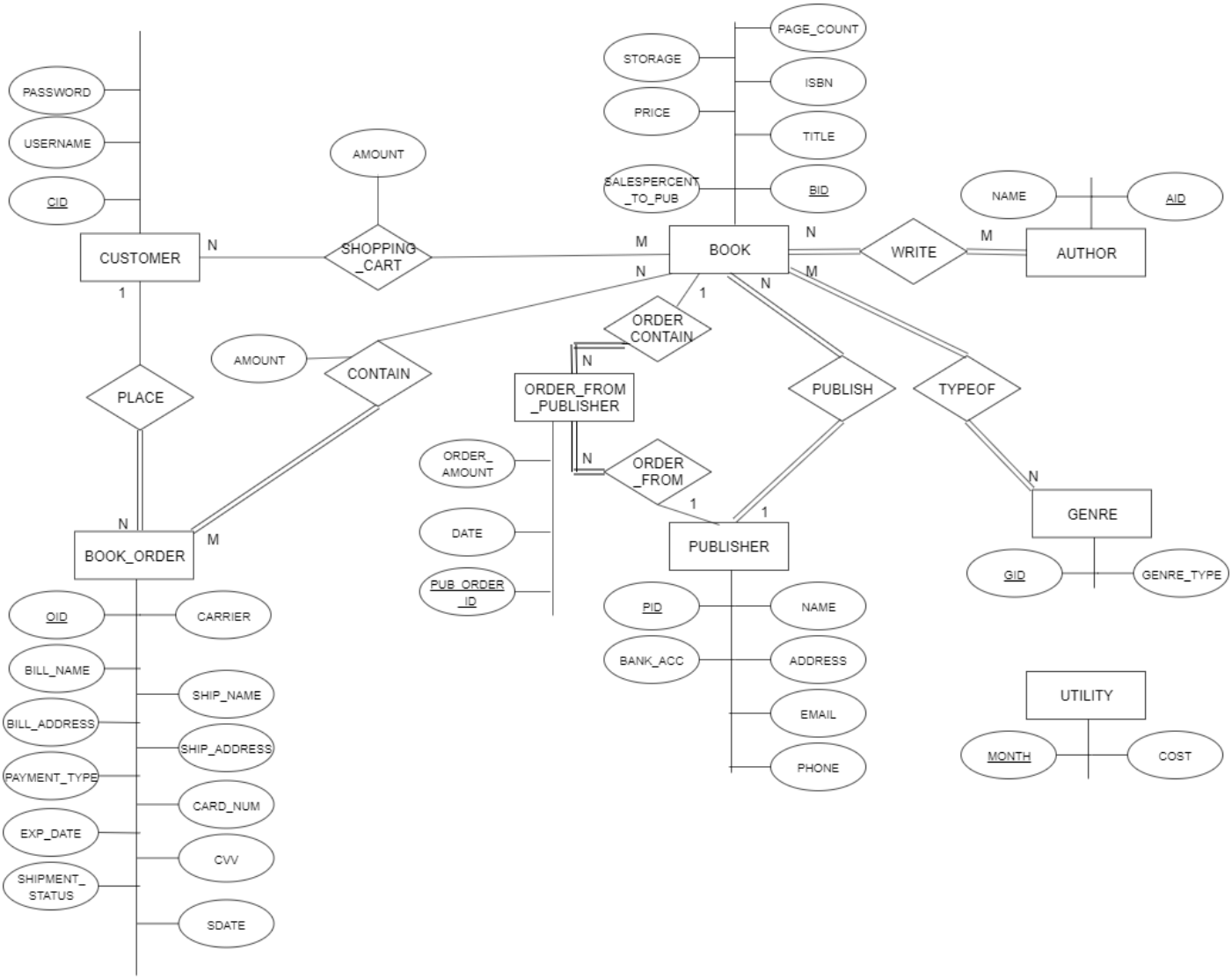
Project Report

Completed individually by:
Catherine Li
student ID:101209708

Table of Content

Conceptual Design

ER-diagram:



Assumptions regarding the ER diagram:

1. assume each book is published by only one publisher, one publisher can publish many books.
2. assume books may have same title
3. assume the ISBN of book is unique and the ISBN is 13 digits, stored as varchar type
4. assume the address in the database is stored within one attribute as varchar type
5. assume all publisher stored in database must publish a book
6. assume all author stored in database must write a book
7. assume authors could not have the same name
8. assume one author can write many books, one book can have more than one author.
9. assume publisher have unique name, but could have same bank account, same address, same phone and email (owed by same boss/person)
10. assume customer username must be unique

Other assumptions:

1. assume utility entity is updated every year, that is, database only save the utility for the current year
2. assume the bookstore opened the date of 2022/01/01
3. assume only the owner can only review the reports in the current year

Reduction to Relation Schemas

PUBLISHER

PID	BANK_ACC	NAME	ADDRESS	EMAIL	PHONE

BOOK

BID	TITLE	ISBN	PAGE_COUNT	STORAGE	PRICE	SALESPERCENT_TO_PUB	PID

GENRE

GID	GENRE_TYPE

TYPEOF

GID	BID

AUTHOR

AID	NAME

WRITE

AID	BID

CUSTOMER

CID	USERNAME	PASSWORD

SHOPPING_CART

CID	BID	AMOUNT

BOOK_ORDER

OID	CARRIER	SHIP_NAME	SHIP_ADDRESS	BILL_NAME	BILL_ADDRESS	CARD_NUM
CVV	SDATE	EXP_DATE	PAYMENT_TYPE	SHIPMENT_STATUS	CID	

CONTAIN

OID	BID	AMOUNT

ORDER_FROM_PUBLISHER

PUB_ORDER_ID	PID	BID	DATE	ORDER_AMOUNT

UTILITY

MONTH	COST

Normalization of Relation Schemas

Here I denote set of functional dependencies as F for each schema

PUBLISHER TABLE:
F={PID -> BANK_ACC, NAME, ADDRESS, EMAIL, PHONE
NAME - > BANK_ACC, PID, ADDRESS, EMAIL, PHONE}
pid is superkey for publisher table, name is superkey for publisher table. Therefore, BCNF holds.

BOOK TABLE:
F={BID ->TITLE, ISBN, PAGE_COUNT, STORAGE, PRICE, SALESPERCENT_TO_PUB, PID
ISBN -> TITLE, BID, PAGE_COUNT, STORAGE, PRICE, SALESPERCENT_TO_PUB, PID}
BID is superkey for BOOK table, ISBN is superkey for BOOK table. Therefore, BCNF holds.

GENRE TABLE
F={GID -> GENRE_TYPE
GENRE_TYPE->GID}
GID is superkey for GENRE table, GENRE_TYPE is superkey for GENRE table. Therefore, BCNF holds.

TYPEOF TABLE
F=∅, functional dependencies is empty for TYPEOF table, BCNF holds

AUTHOR TABLE
F={AID -> NAME}
F={ NAME ->AID}
AID is superkey for AUTHOR table. NAME is superkey for AUTHOR table. Therefore, BCNF holds

WRITE TABLE
F=∅, functional dependencies is empty for WRITE table, BCNF holds

CUSTOMER
F={CID -> USERNAME, PASSWORD
USERNAME -> CID, PASSWORD}
CID is superkey for CUSTOMER table, USERNAME is superkey for CUDTOMER table. Therefore, BCNF holds.

SHOPPING_CART TABLE
F={CID, BID -> AMOUNT}
(CID,BID) is superkey for SHOPPING_CART table. Therefore, BCNF holds.

BOOK_ORDER TABLE
F={OID -> CARRIER,SHIP_NAME, SHIP_ADDRESS, BILL_NAME, BILL_ADDRESS, CARD_NUM,
CVV, SDATE, EXP_DATE, PAYMENT_TYPE, SHIPMENT_STATUS, CID
CARD_NUM -> CVV, EXP_DATE, PAYMENT_TYPE, BILL_NAME, BILL_ADDRESS}
OID is superkey for BOOK_ORDER table.
However, CARD_NUM+ = CVV, EXP_DATE, PAYMENT_TYPE, BILL_NAME, BILL_ADDRESS
We know that CARD_NUM is not superkey for BOOK_ORDER table. This table needs to be decomposed.
Let R be the old table book_order
Let result = R, α = CARD_NUM, β = CVV, EXP_DATE, PAYMENT_TYPE, BILL_NAME, BILL_ADDRESS

FIRST LOOP:
result = ($R - \beta$) $\cup$ (α, β) =
*(*OID, CARRIER, SHIP_NAME, SHIP_ADDRESS, SDATE, SHIPMENT_STATUS, CID, CARD_NUM) $\cup$ (CARD_NUM, CVV, EXP_DATE, PAYMENT_TYPE, BILL_NAME, BILL_ADDRESS)
Check if ($R - \beta$) in BCNF:
Since OID -> CARRIER, SHIP_NAME, SHIP_ADDRESS, SDATE, SHIPMENT_STATUS, CID, CARD_NUM, OID is the superkey of ($R - \beta$). Therefore, ($R - \beta$) is in BCNF
So, BOOK_ORDER TABLE is decomposed to
BOOK_ORDER:

OID	CARRIER	SHIP_NAME	SHIP_ADDRESS	CARD_NUM	SDATE	SHIPMENT_STATUS	CID

PAYMENT_CARD:

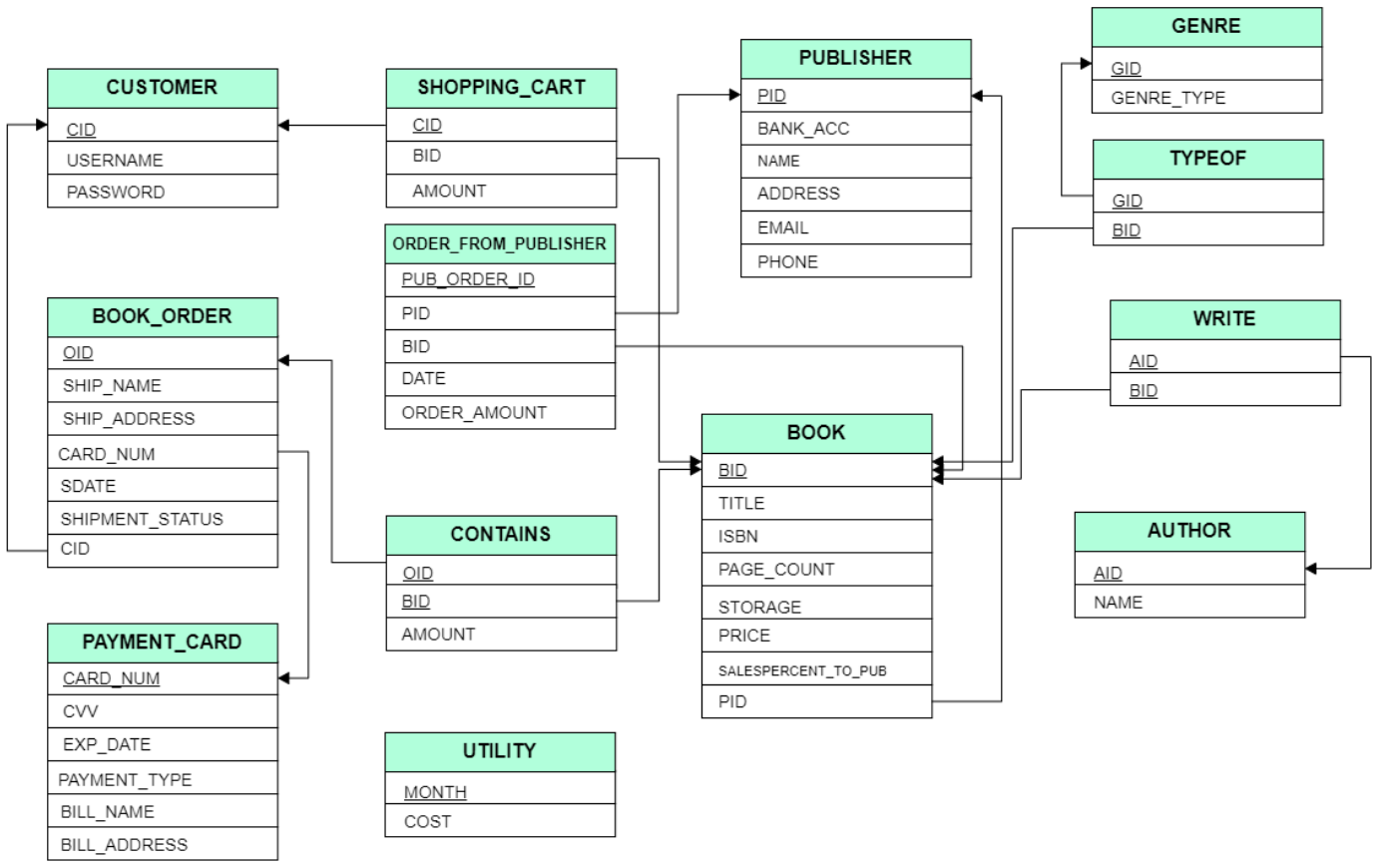
CARD_NUM	CVV	EXP_DATE	PAYMENT_TYPE	BILL_NAME	BILL_ADDRESS

CONTAIN TABLE
F={OID,BID->AMOUNT}
(OID,BID) is superkey for CONTAIN table. Therefore, BCNF holds.

ORDER_FROM_PUBLISHER TABLE
F={ PUB_ORDER_ID->PID, BID, DATE, ORDER_AMOUNT}
PUB_ORDER_ID is superkey for CONTAIN table. Therefore, BCNF holds.

UTILITY TABLE
F={MONTH -> COST}
MONTH is superkey for UTILITY table. Therefore, BCNF holds.

Database Schema Diagram

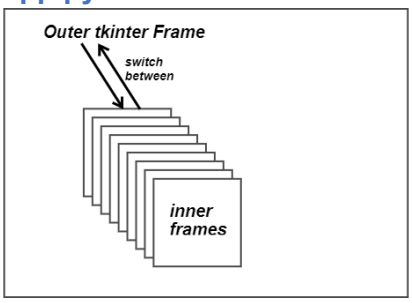


Implementation

This program is written in Python, tkinter is used to build up GUIs, postgresql is the choice of databse.
This program mainly contains two modules: dbinitializer.py and app.py
The database is connected to an online PostgreSQL database: https://www.elephantsql.com/

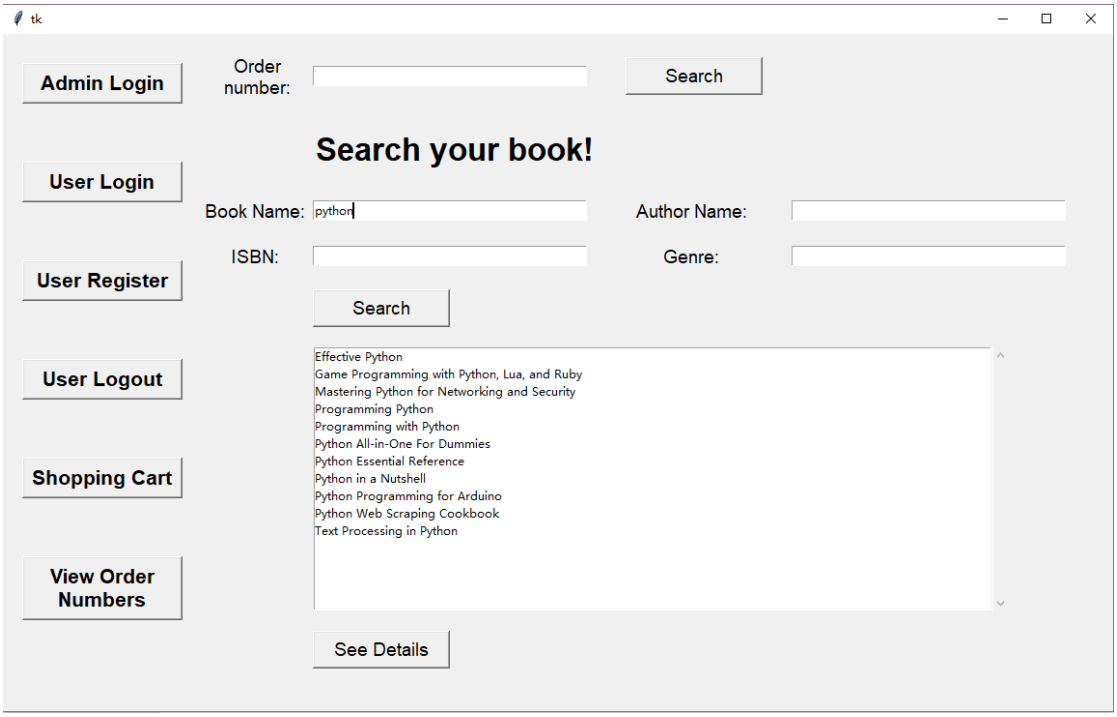
dbinitializer.py builds up the database using DDL and input the initial data into database including 606 books information download from online database, 255 fake publishers' information, 500 fake customer and 3000 fake book orders.

app.py contains the structure of the desktop app:



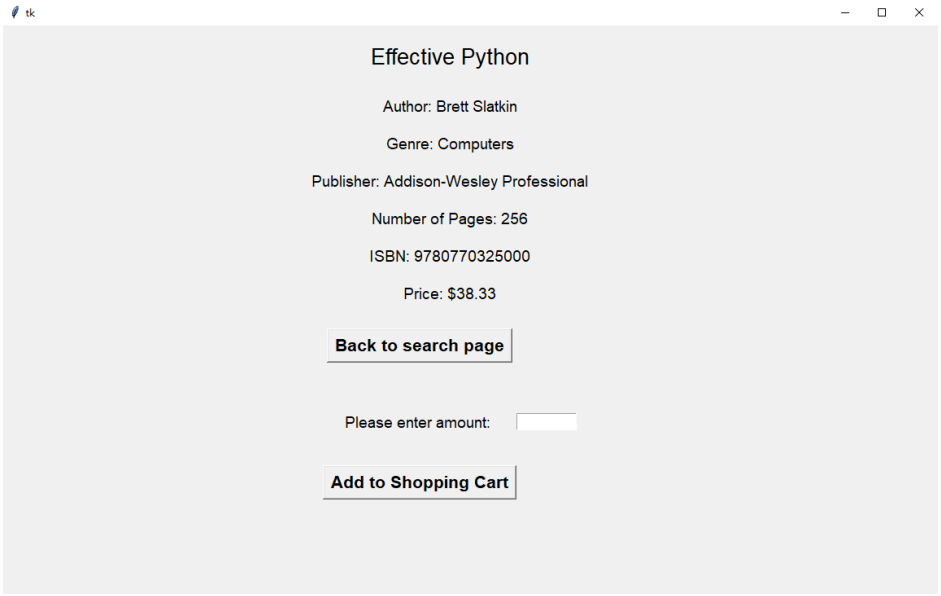
It contains one outer frame class and 15 inner frame classes:
Outer frame class: BookStoreAppGUI
This class builds up the outer frame of the app. It is passed as parent frame to other class. By controlling the swicth_frame() function in this class, we can go to any pages(these pages are built as inner frames).
Page class (inner frame class):

1. adminLoginPage: log in page for administration / the bookstore owner. After logging in through this page, owner will jump to the admin interface. The admin username and password are both "admin"
2. UserLoginPage: log in page for customer, after logging in, customer can add books to shopping cart and check out
3. UserRegPage: register page for customer
4. FirstPage: First Pages for both customer and owner. Owner can go to admin login page from this page. Customer can login/register from this page. Also, customer can search books in this page. The author name and genre should be enter only one name or genre type each time.
   Customer can also search orders by order numbers after logging in.

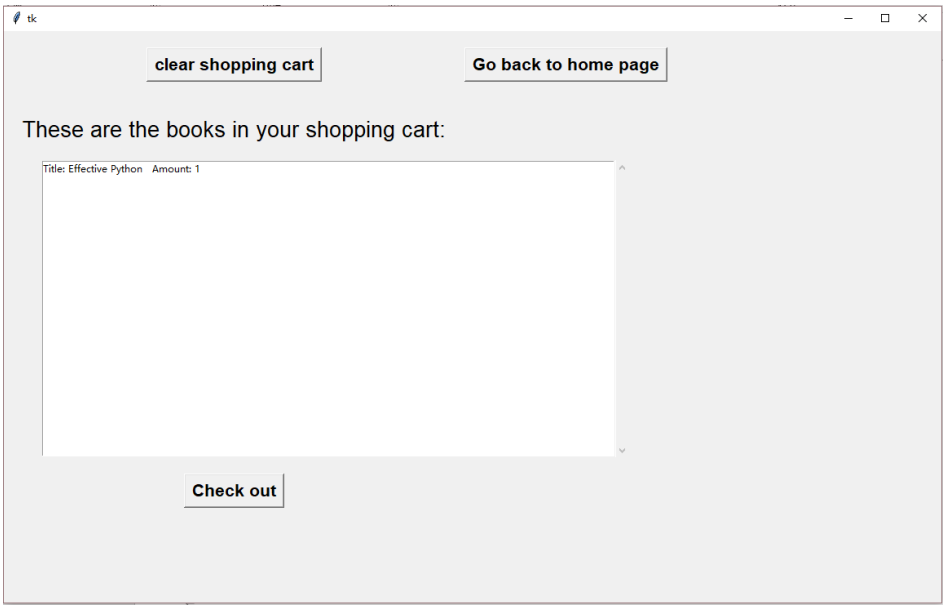


Customer interface:

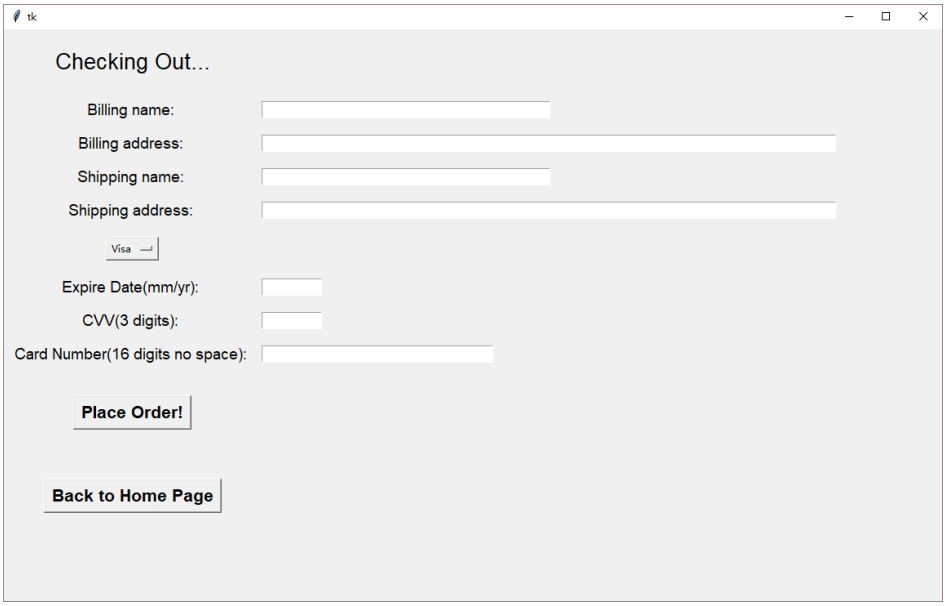
5. BookPage: shows the details of a book, customer can add this book to shopping cart after logging in



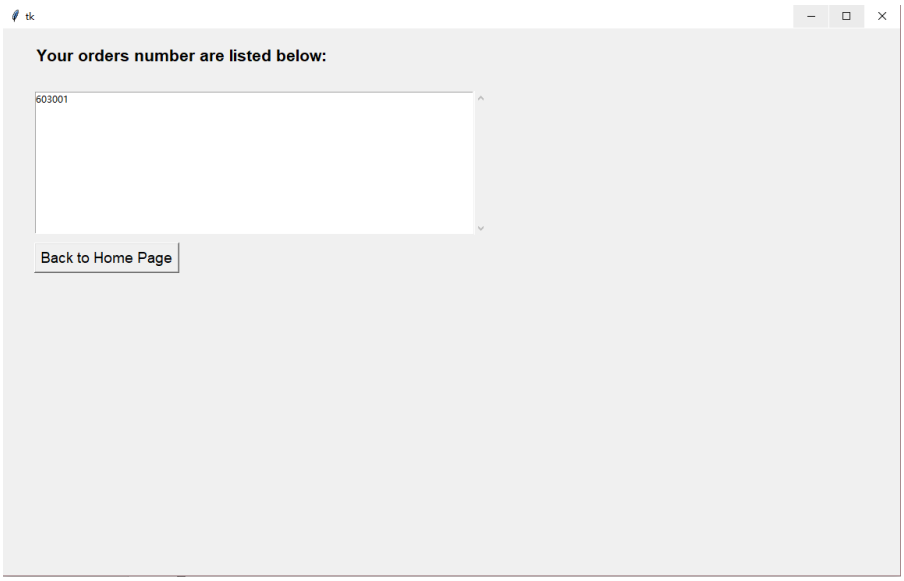
6. CartPage: shows all the books in customer's shopping cart, check out button redirect to check out page



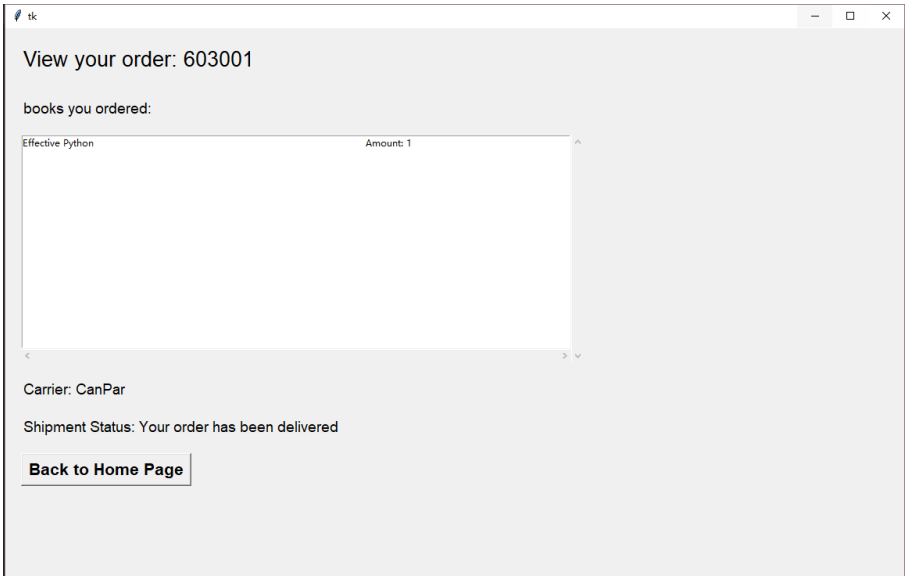
7. CheckoutPage: customer enter bill/ship information and place order in this page, after clicking place order button, an order number will be given.



8. GetOrderNumberPage: If the customer forgot their order number, they can click the view order numbers button in the first page and view all the history order numbers in this page

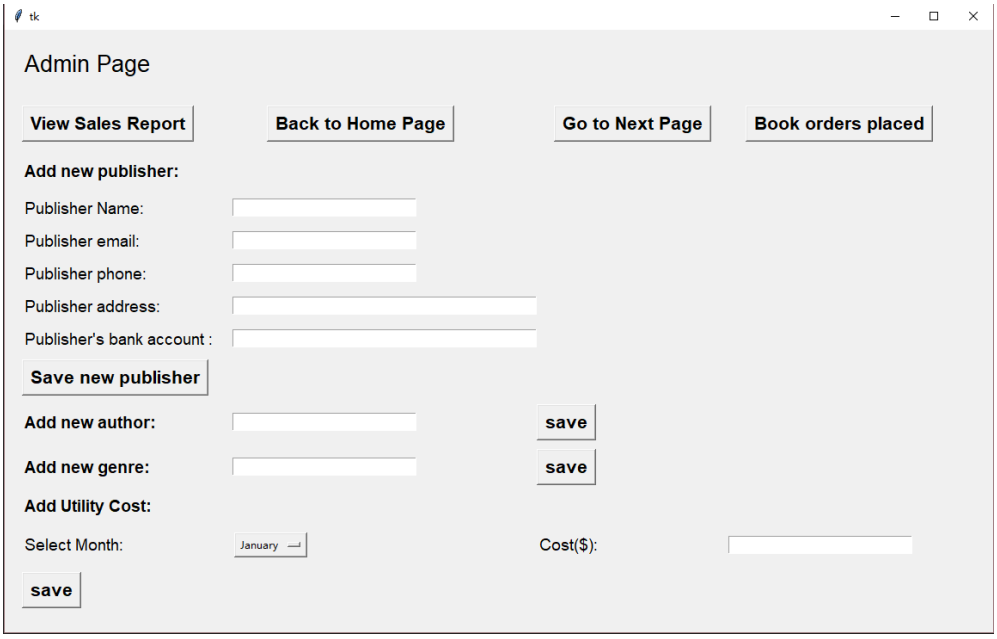


9. OrderPage: shows the details of the order number searched.



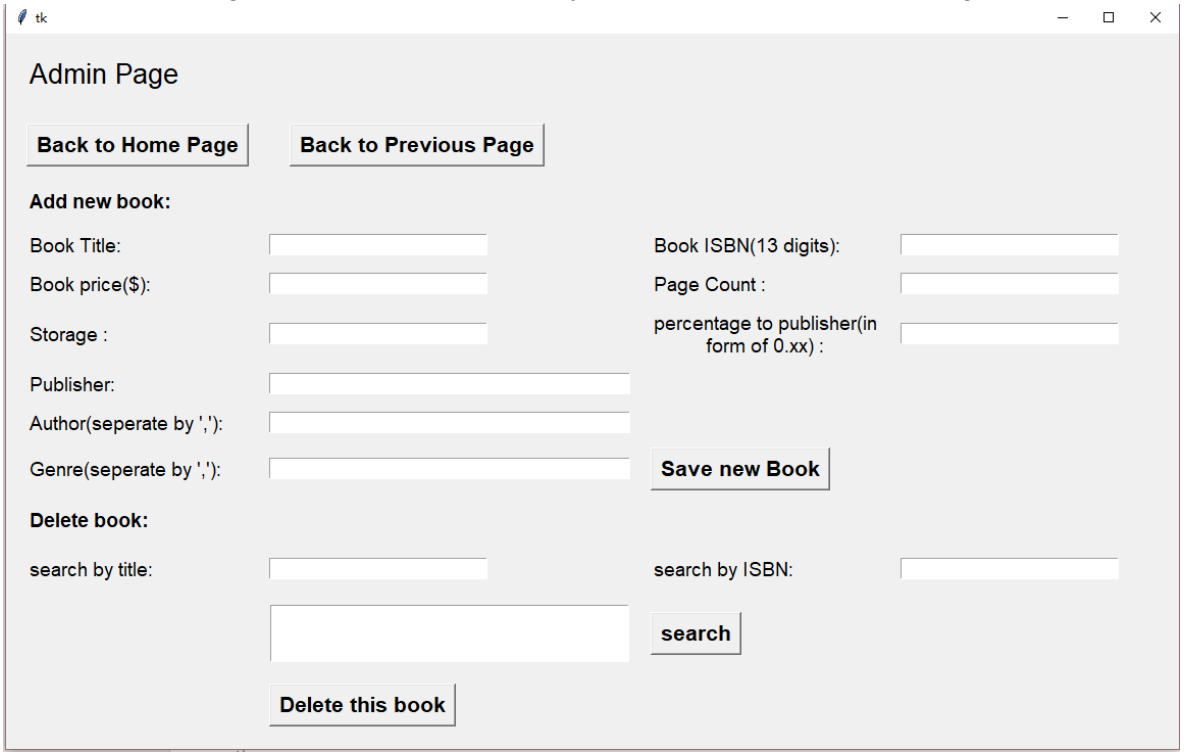
Admin interface：

10. adminFirstPage: Accessed after completing the admin login page.

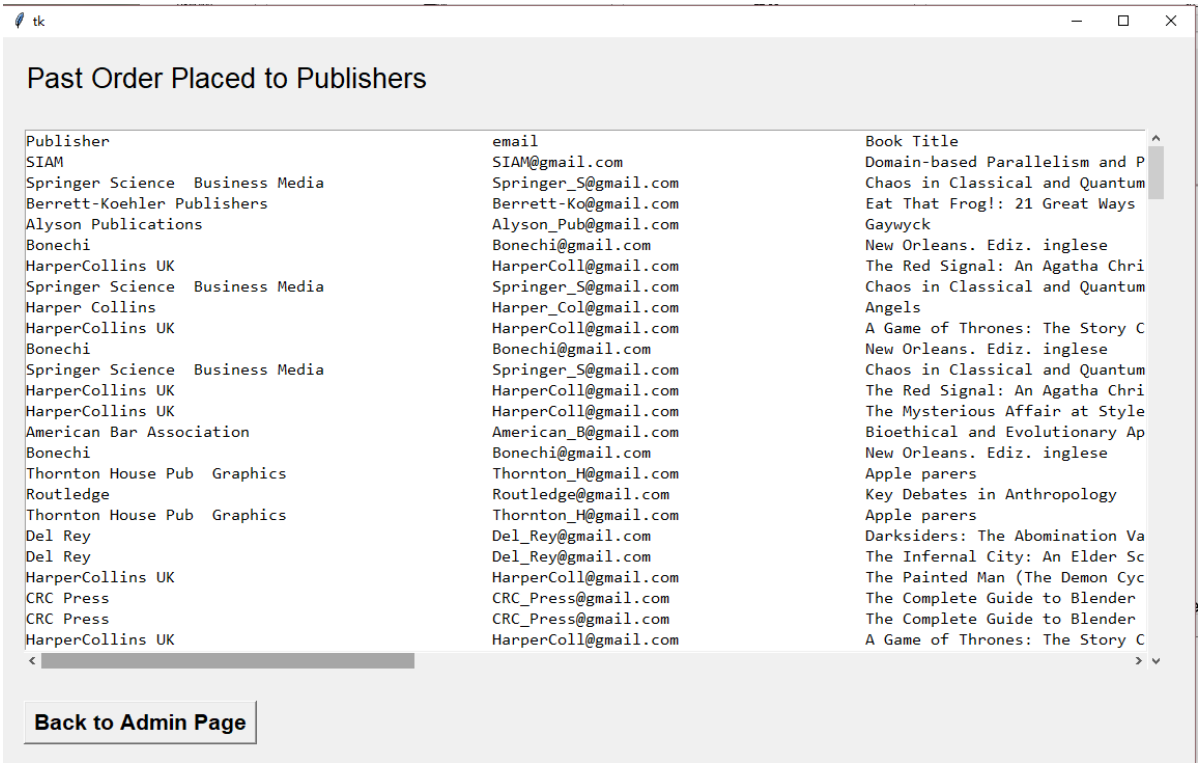


Owners can add new publisher, new author, new genre, and record utility cost in this page. Owner must add single author name and single genre type per time.

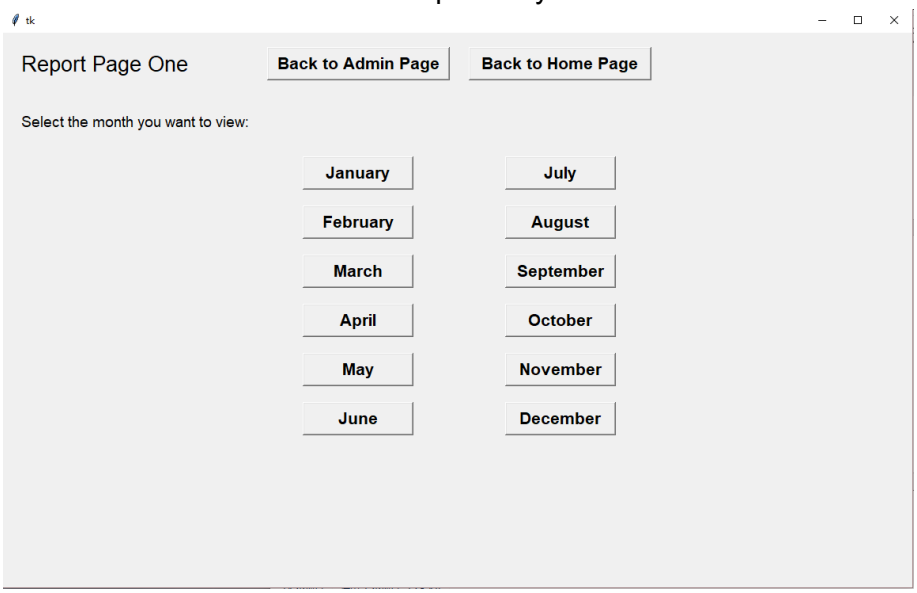
11. adminSecondPage： owners can add/delete books in this page. Owner can enter multiple authors, genres (separate them with comma), the percentage sent to publisher must entered as decimal number, if the publisher is not added, owner must go to adminFirstPage and add the full information of the publisher, if the author or genre are not added, the system will add the author and genre automatically in this page



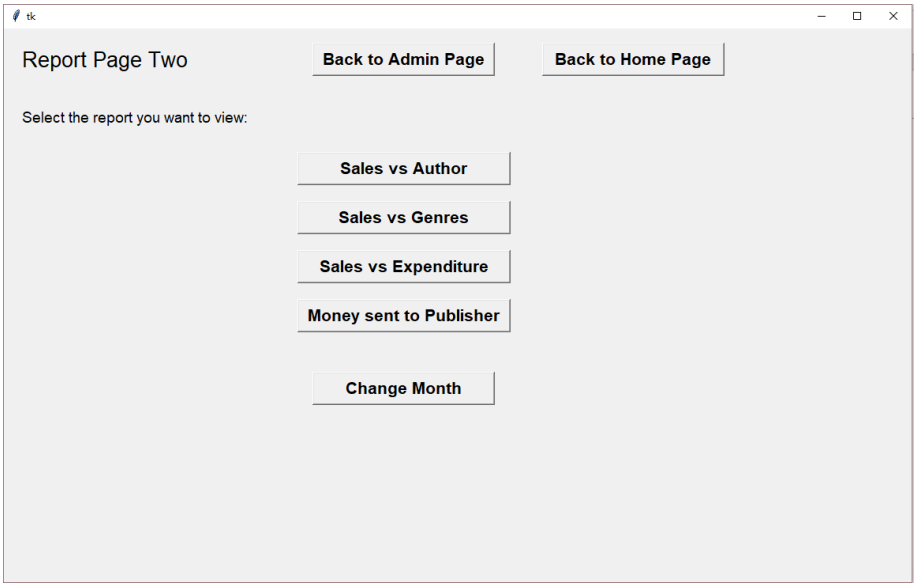
12. OrderfromPubPage: The page is accessed by the ‘books order placed’ button in the admin first page. It shows the previous order automatically placed by the bookstore when the storage of a book is less than 20. The email sending component is not implemented in this project, but the information used to write to email is stored in the database and shown as record in this table.



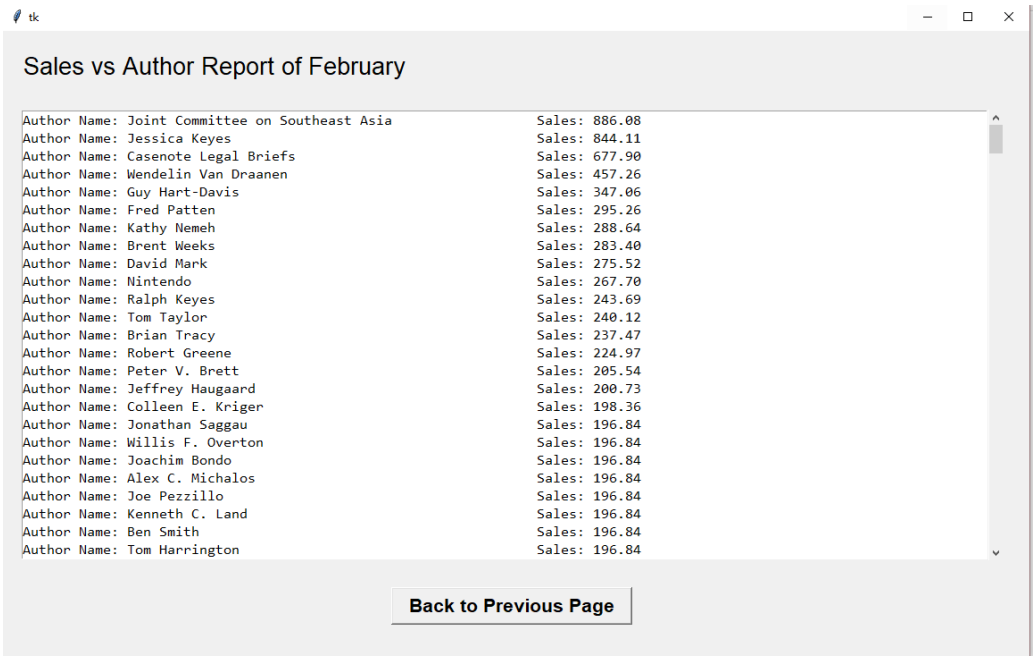
13. reportFirstPage: This page is accessed by clicking the view sales report button in the first page. Owner can select the month of report they want to view.

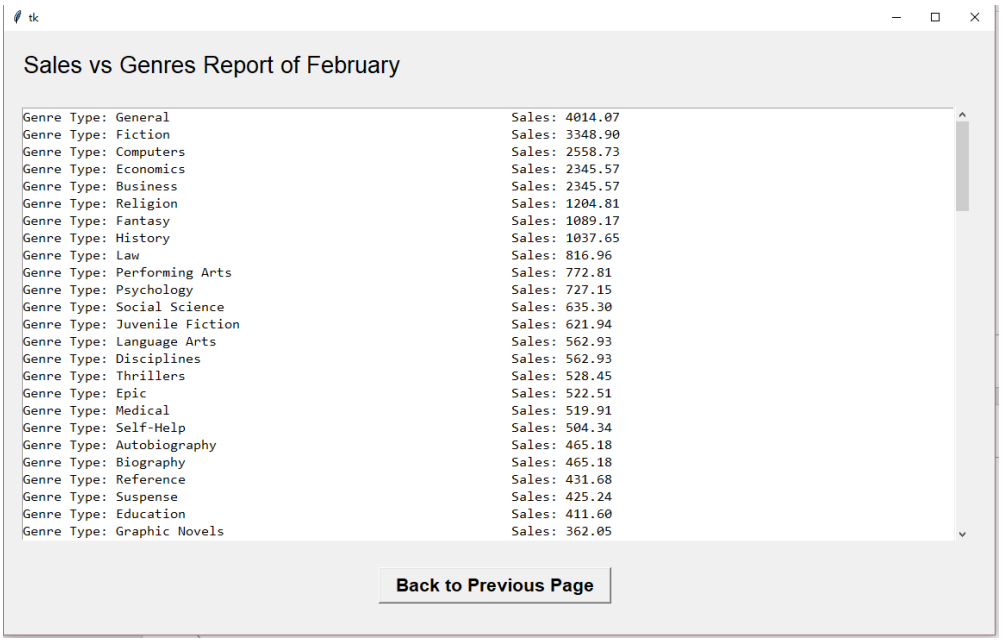


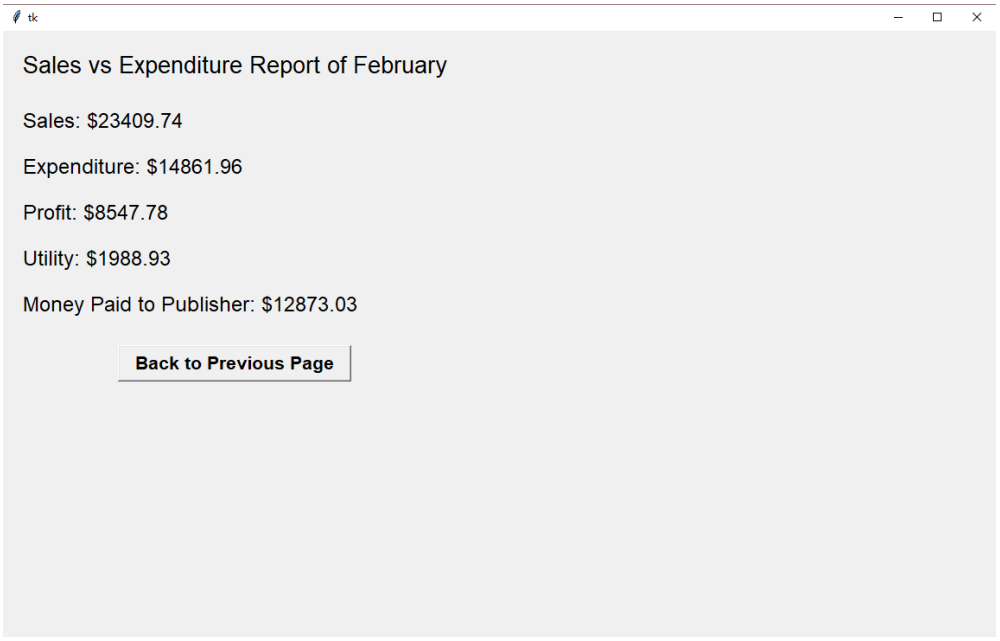
14. reportSecondPage: Accessed after selecting the month, owner choose which report they want to view

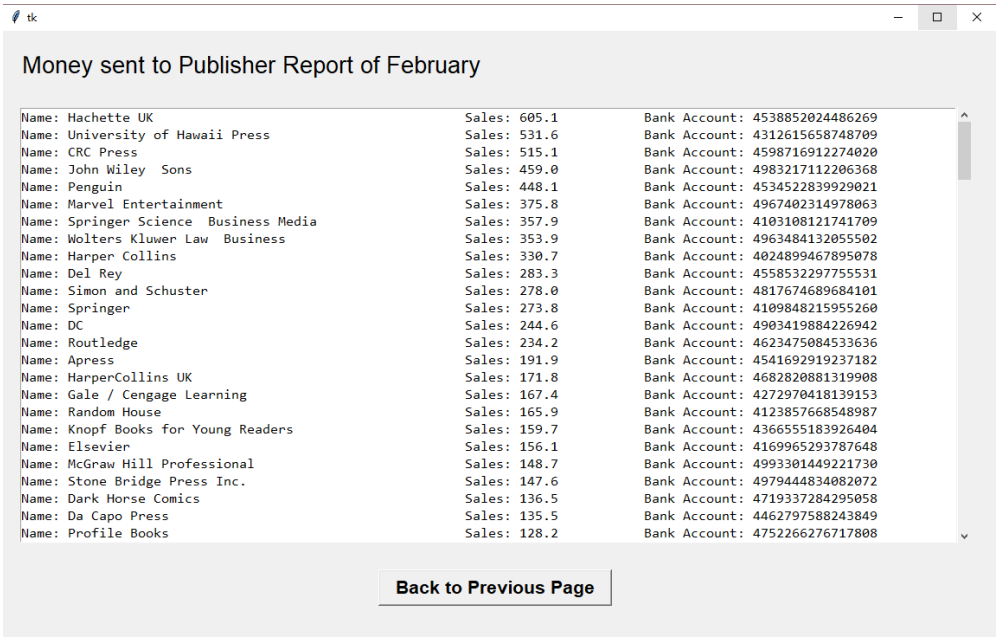


15. reportThirdPage: Accessed after chosen the type of reports. There are four reports shown in this page:

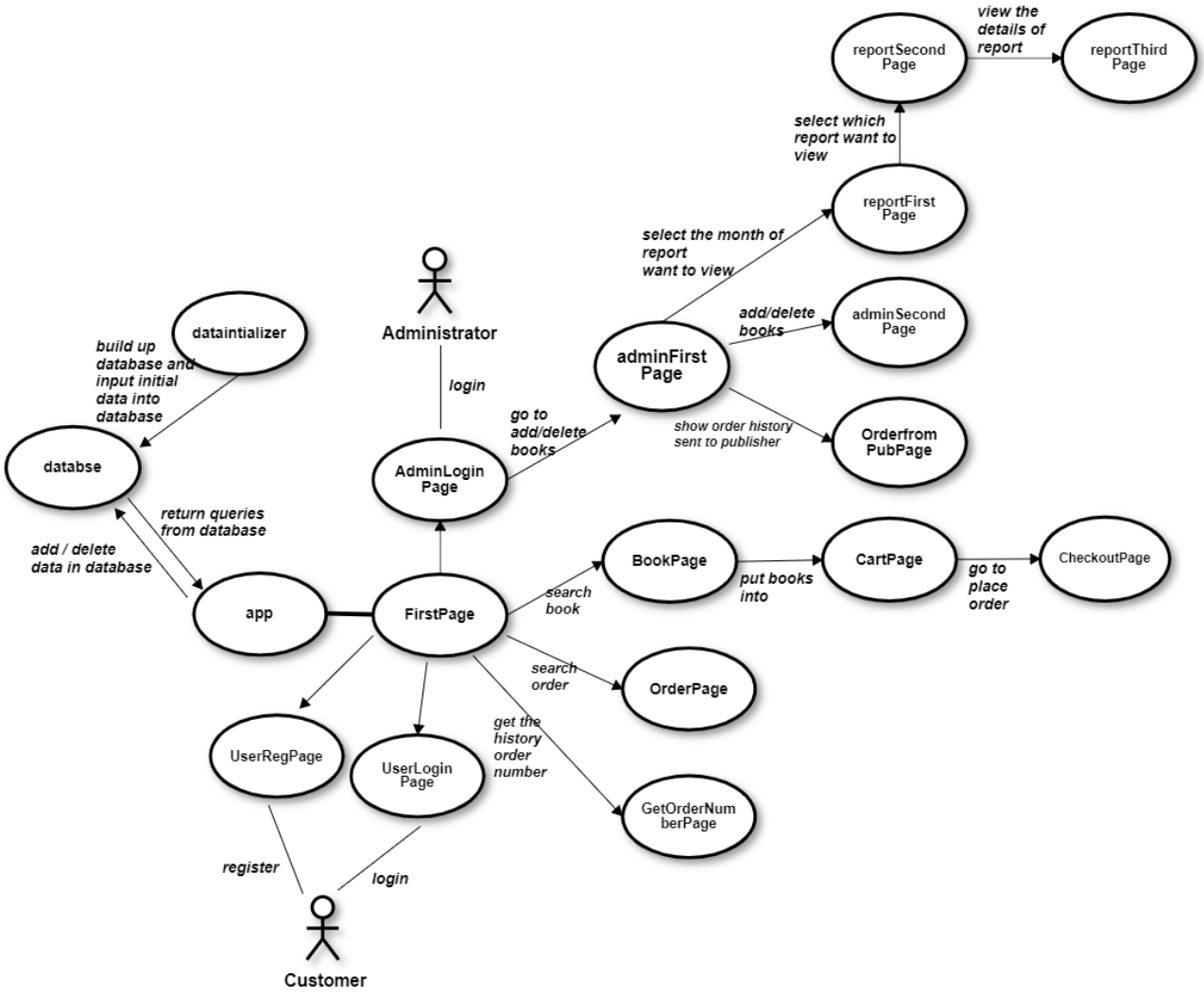



tk
Sales vs Genres Report of February
Genre Type: General Sales: 4014.07
Genre Type: Fiction Sales: 3348.90
Genre Type: Computers Sales: 2558.73
Genre Type: Economics Sales: 2345.57
Genre Type: Business Sales: 2345.57
Genre Type: Religion Sales: 1204.81
Genre Type: Fantasy Sales: 1089.17
Genre Type: History Sales: 1037.65
Genre Type: Law Sales: 816.96
Genre Type: Performing Arts Sales: 772.81
Genre Type: Psychology Sales: 727.15
Genre Type: Social Science Sales: 635.30
Genre Type: Juvenile Fiction Sales: 621.94
Genre Type: Language Arts Sales: 562.93
Genre Type: Disciplines Sales: 562.93
Genre Type: Thrillers Sales: 528.45
Genre Type: Epic Sales: 522.51
Genre Type: Medical Sales: 519.91
Genre Type: Self-Help Sales: 504.34
Genre Type: Autobiography Sales: 465.18
Genre Type: Biography Sales: 465.18
Genre Type: Reference Sales: 431.68
Genre Type: Suspense Sales: 425.24
Genre Type: Education Sales: 411.60
Genre Type: Graphic Novels Sales: 362.05
Back to Previous Page


tk
Sales vs Expenditure Report of February
Sales: $23409.74
Expenditure: $14861.96
Profit: $8547.78
Utility: $1988.93
Money Paid to Publisher: $12873.03
Back to Previous Page


tk
Money sent to Publisher Report of February
Name: Hachette UK Sales: 605.1 Bank Account: 4538852024486269
Name: University of Hawaii Press Sales: 531.6 Bank Account: 4312615658748709
Name: CRC Press Sales: 515.1 Bank Account: 4598716912274020
Name: John Wiley Sons Sales: 459.0 Bank Account: 4983217112206368
Name: Penguin Sales: 448.1 Bank Account: 4534522839929021
Name: Marvel Entertainment Sales: 375.8 Bank Account: 4967402314978063
Name: Springer Science Business Media Sales: 357.9 Bank Account: 4103108121741709
Name: Wolters Kluwer Law Business Sales: 353.9 Bank Account: 4963484132055502
Name: Harper Collins Sales: 330.7 Bank Account: 4024899467895078
Name: Del Rey Sales: 283.3 Bank Account: 4558532297755531
Name: Simon and Schuster Sales: 278.0 Bank Account: 4817674689684101
Name: Springer Sales: 273.8 Bank Account: 4109848215955260
Name: DC Sales: 244.6 Bank Account: 4903419884226942
Name: Routledge Sales: 234.2 Bank Account: 4623475084533636
Name: Apress Sales: 191.9 Bank Account: 4541692919237182
Name: HarperCollins UK Sales: 171.8 Bank Account: 4682820881319908
Name: Gale / Cengage Learning Sales: 167.4 Bank Account: 4272970418139153
Name: Random House Sales: 165.9 Bank Account: 4123857668548987
Name: Knopf Books for Young Readers Sales: 159.7 Bank Account: 4366555183926404
Name: Elsevier Sales: 156.1 Bank Account: 4169965293787648
Name: McGraw Hill Professional Sales: 148.7 Bank Account: 4993301449221730
Name: Stone Bridge Press Inc. Sales: 147.6 Bank Account: 4979444834082072
Name: Dark Horse Comics Sales: 136.5 Bank Account: 4719337284295058
Name: Da Capo Press Sales: 135.5 Bank Account: 4462797588243849
Name: Profile Books Sales: 128.2 Bank Account: 4752266276717808
Back to Previous Page

Below is the diagram of the application's architecture:



Bonus Features

approximate search for books: user can search book by title or author by entering any parts of the words. For example: if user search "enny", the result will include authors named "Jenny, Benny…".

GitHub Repository

https://github.com/catherine28shiro/3005project

Appendix I

My availability is 2pm-5pm 12th 2022

Appendix II

The password and username of the online database: https://www.elephantsql.com/
username: catherineli3@cmail.carleton.ca
psw: Database3005!

The source of the book data: https://www.kaggle.com/datasets/bilalyussef/google-books-dataset