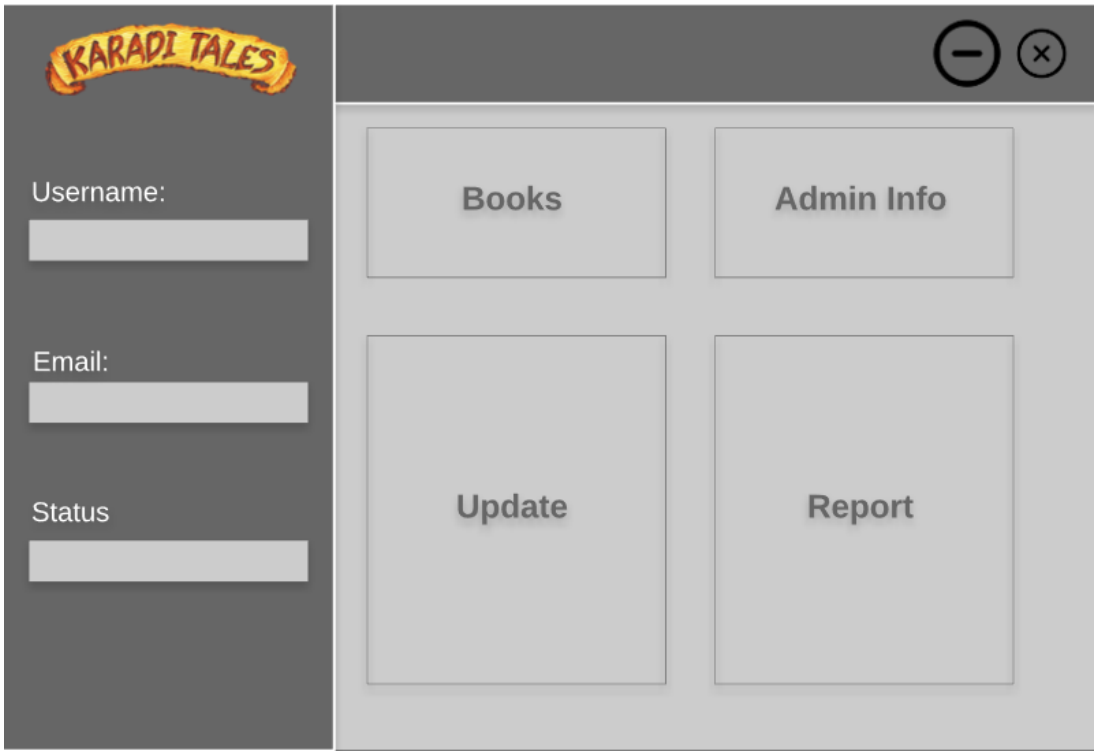
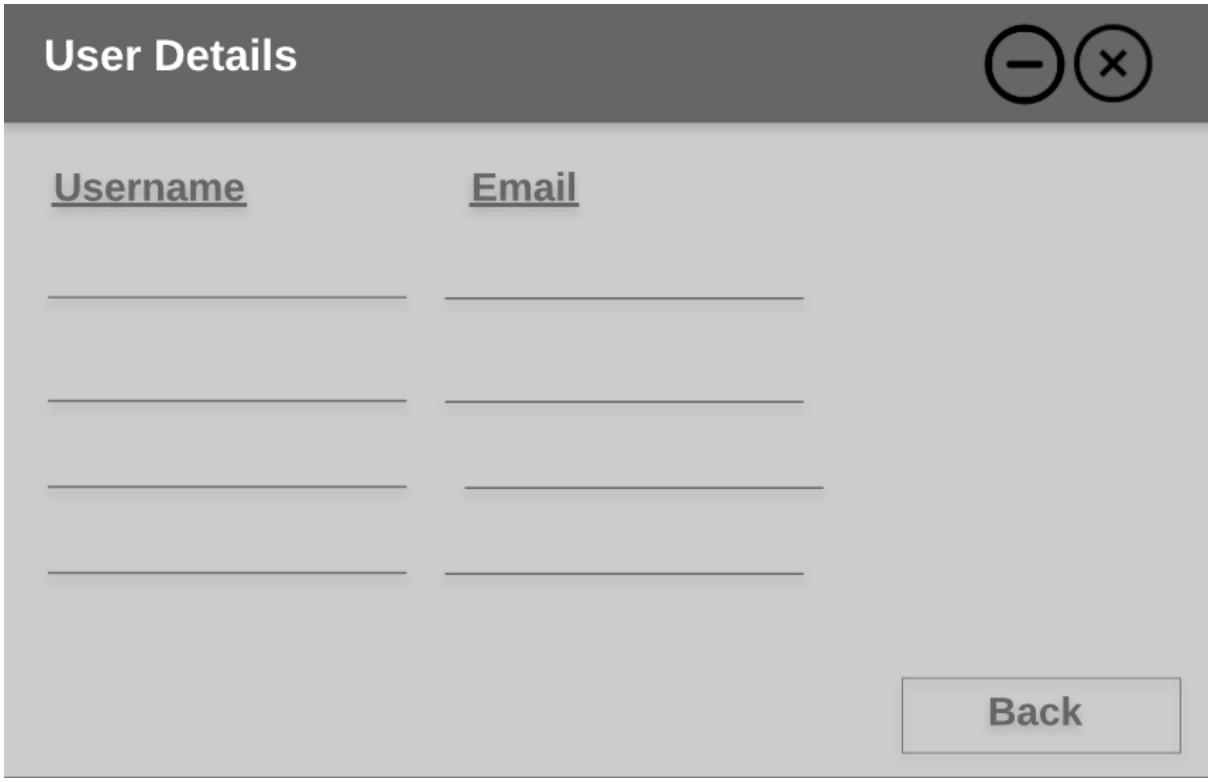
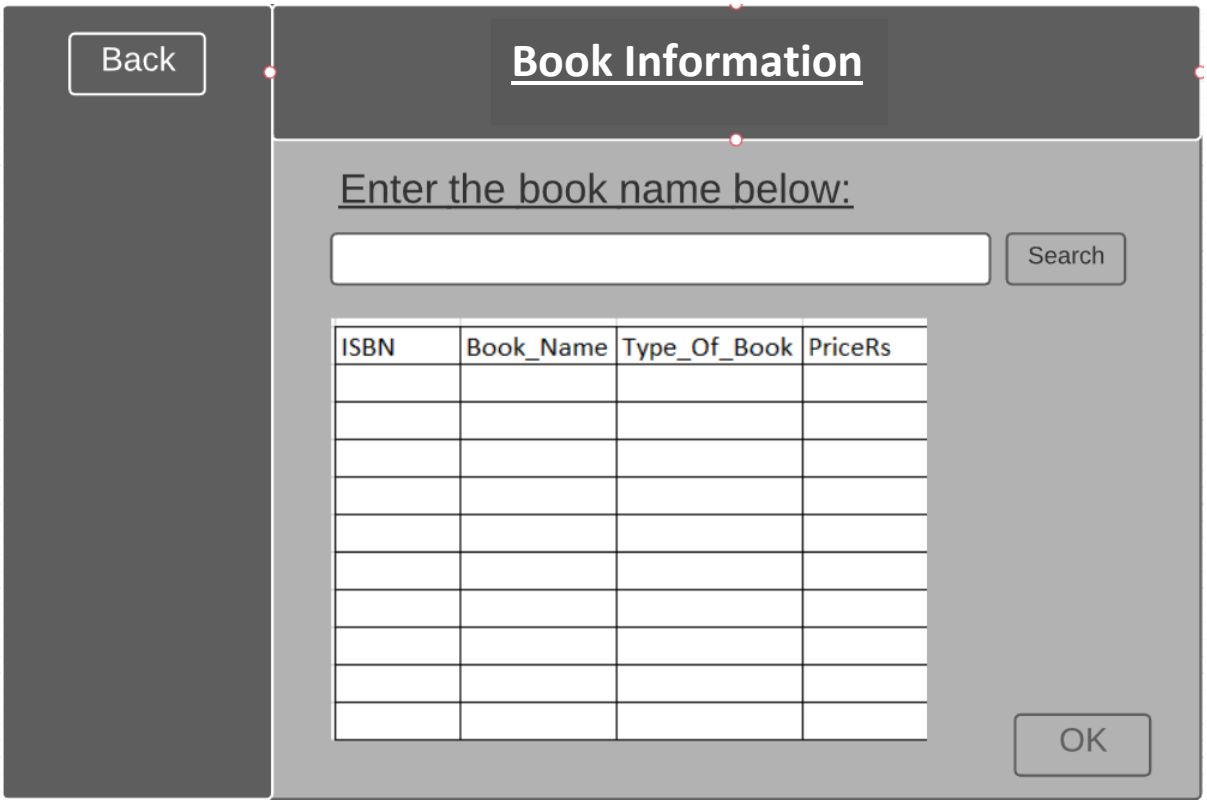
Criterion B: Design

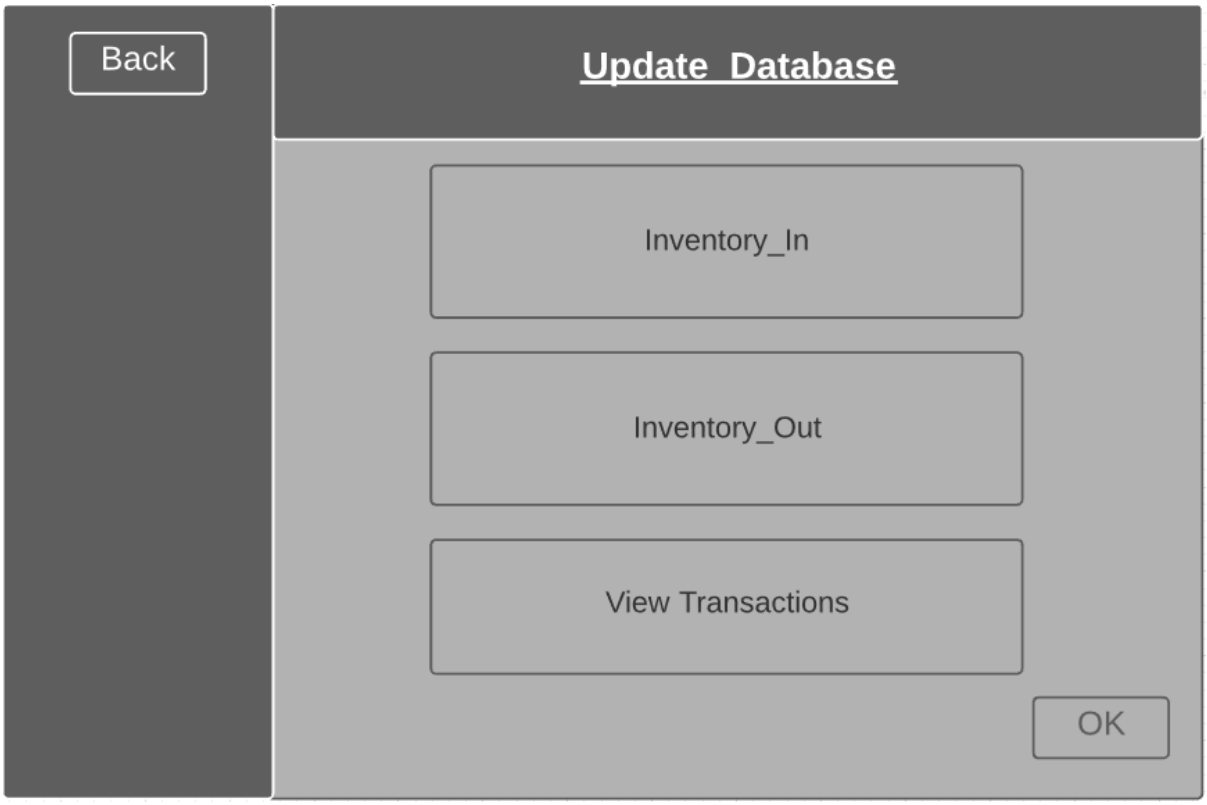
Design Overview[1]:

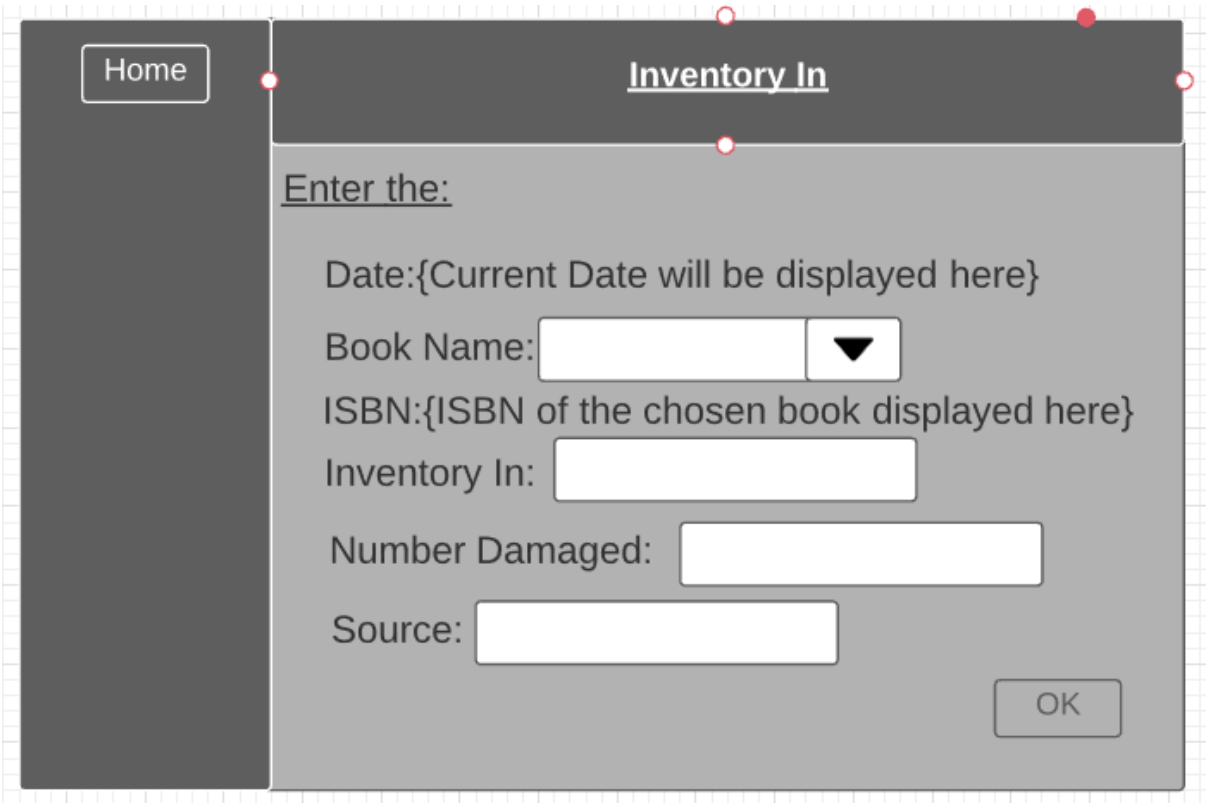


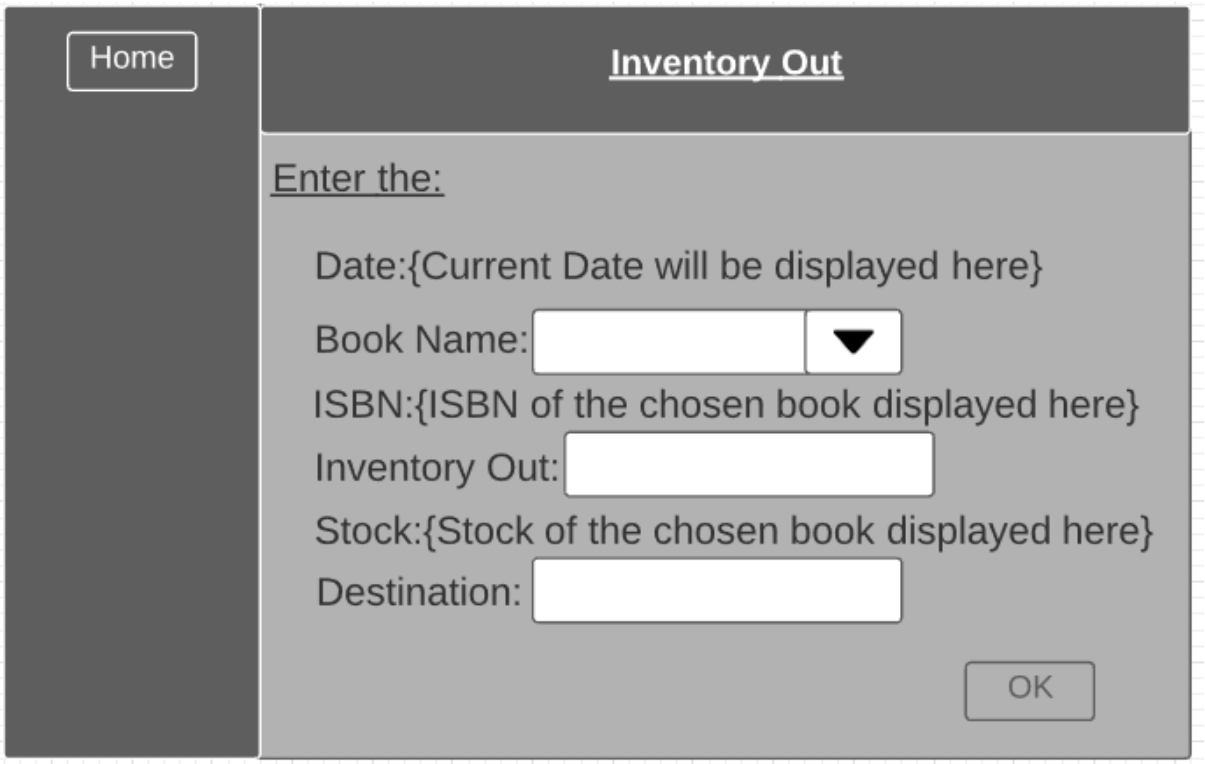
[1] This is the final design. The earlier versions are included in the Appendix_Feedback

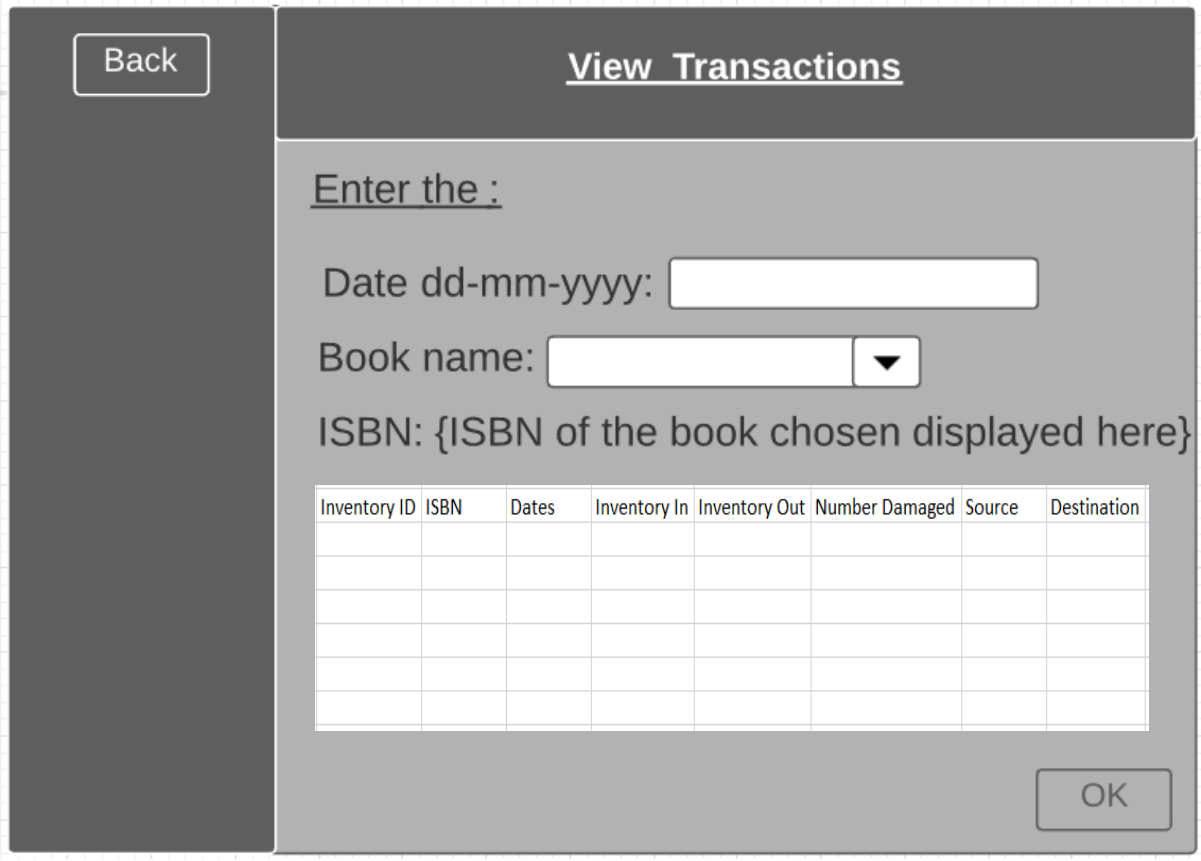

User Details
Username
Email
Back

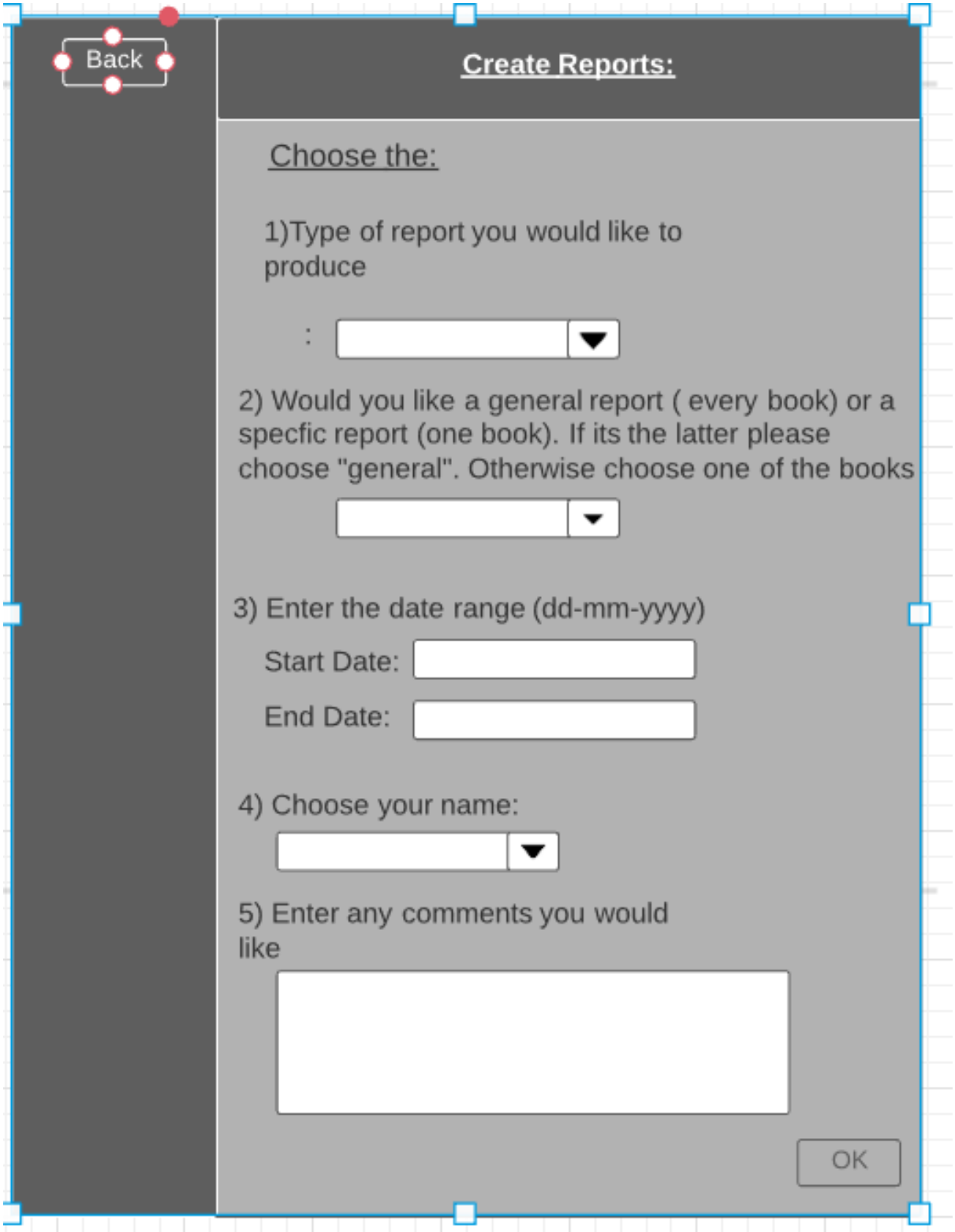

Back
Book Information
Enter the book name below:
Search
ISBN
Book_Name
Type_Of_Book
PriceRs
OK


Back
Update_Database
Inventory_In
Inventory_Out
View Transactions
OK


Home
Inventory In
Enter the:
Date:{Current Date will be displayed here}
Book Name:
ISBN:{ISBN of the chosen book displayed here}
Inventory In:
Number Damaged:
Source:
OK


Home
Inventory Out
Enter the:
Date:{Current Date will be displayed here}
Book Name:
ISBN:{ISBN of the chosen book displayed here}
Inventory Out:
Stock:{Stock of the chosen book displayed here}
Destination:
OK


Back
View Transactions
Enter the :
Date dd-mm-yyyy:
Book name:
ISBN: {ISBN of the book chosen displayed here}
Inventory ID
ISBN
Dates
Inventory In
Inventory Out
Number Damaged
Source
Destination
OK


Back
Create Reports:
Choose the:
1)Type of report you would like to produce
:
2) Would you like a general report (every book) or a specfic report (one book). If its the latter please choose "general". Otherwise choose one of the books
3) Enter the date range (dd-mm-yyyy)
Start Date:
End Date:
4) Choose your name:
5) Enter any comments you would like
OK

Back

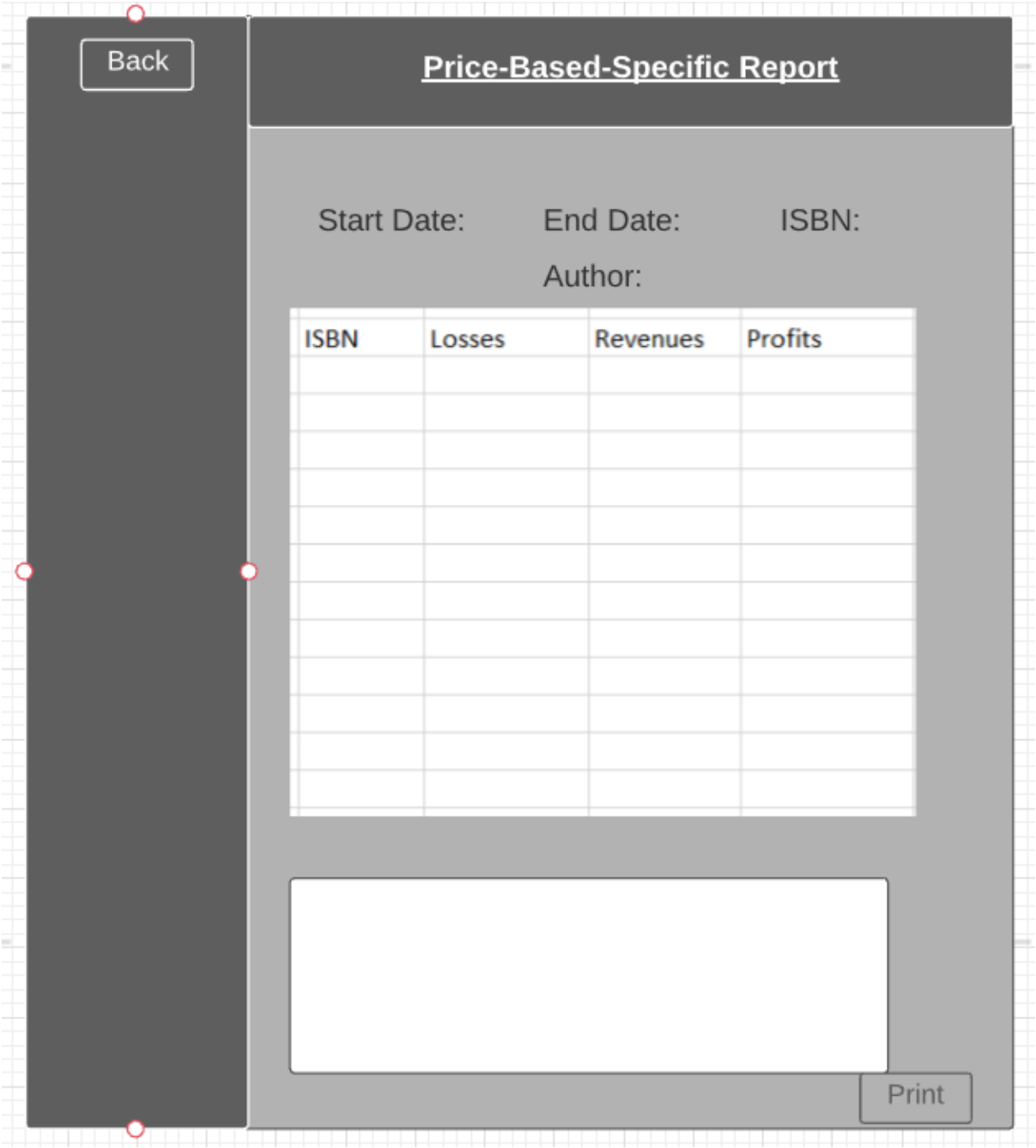
Stock-Based-Specific Report

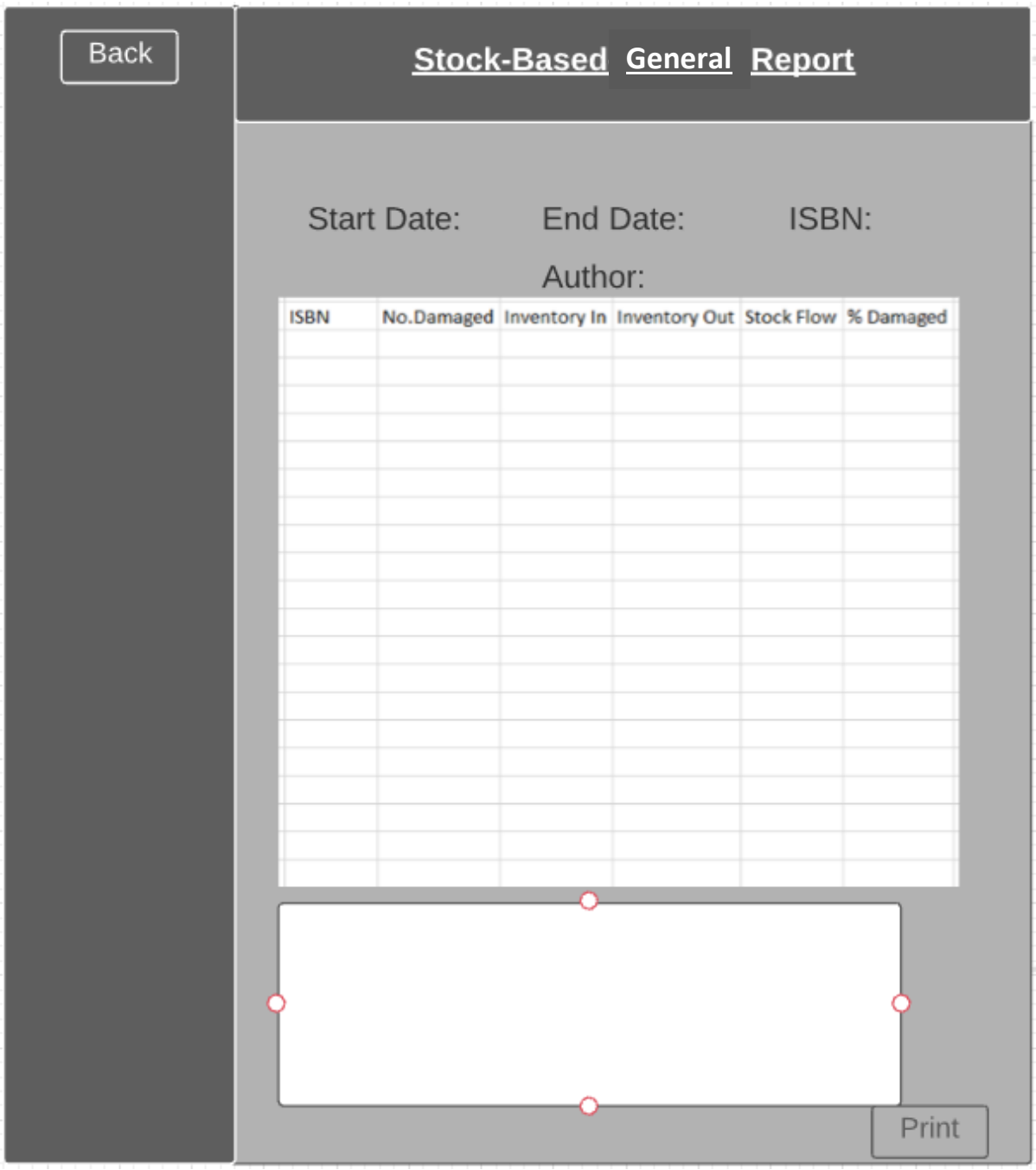
Start Date: End Date: ISBN:

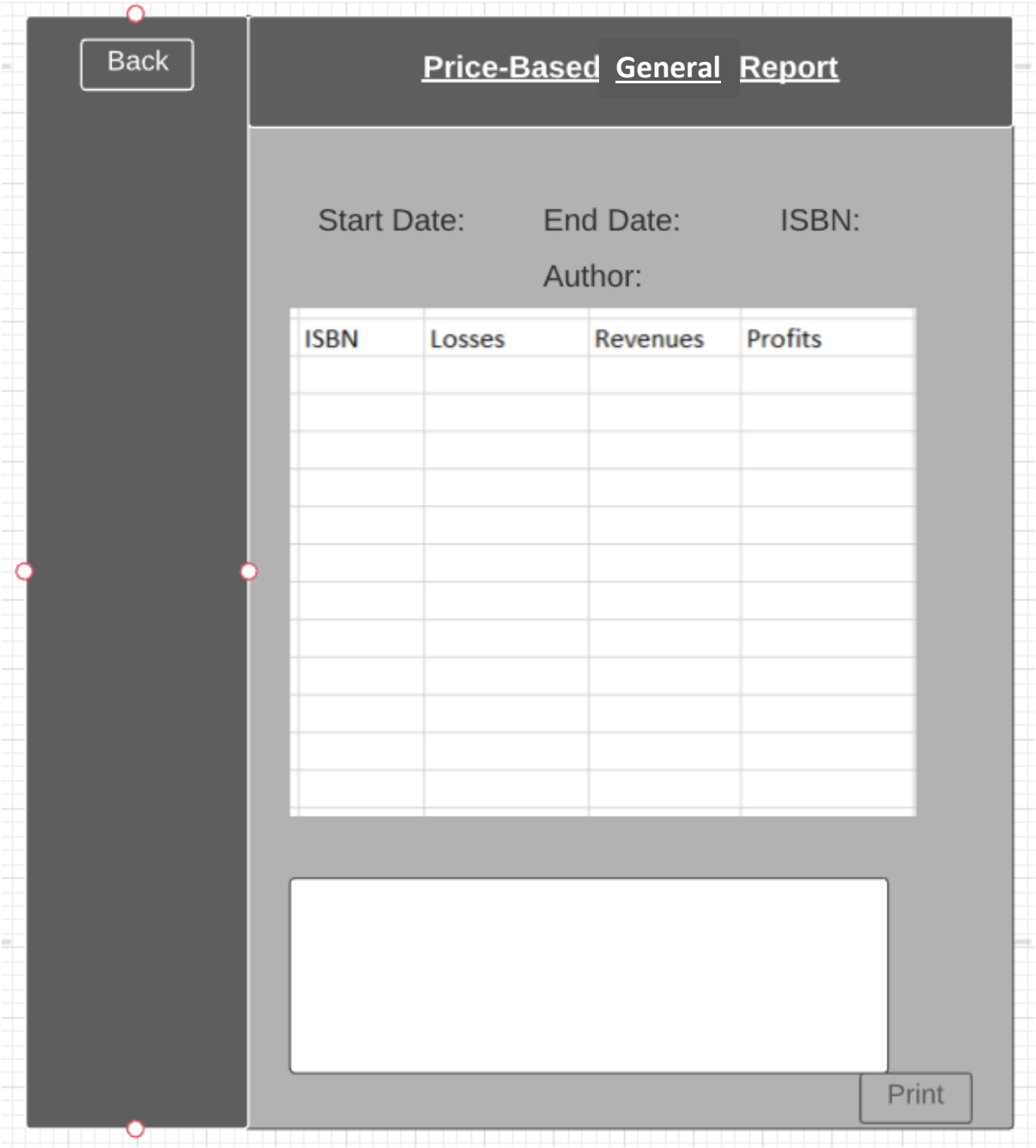
Author:

ISBN	No.Damaged	Inventory In	Inventory Out	Stock Flow	% Damaged

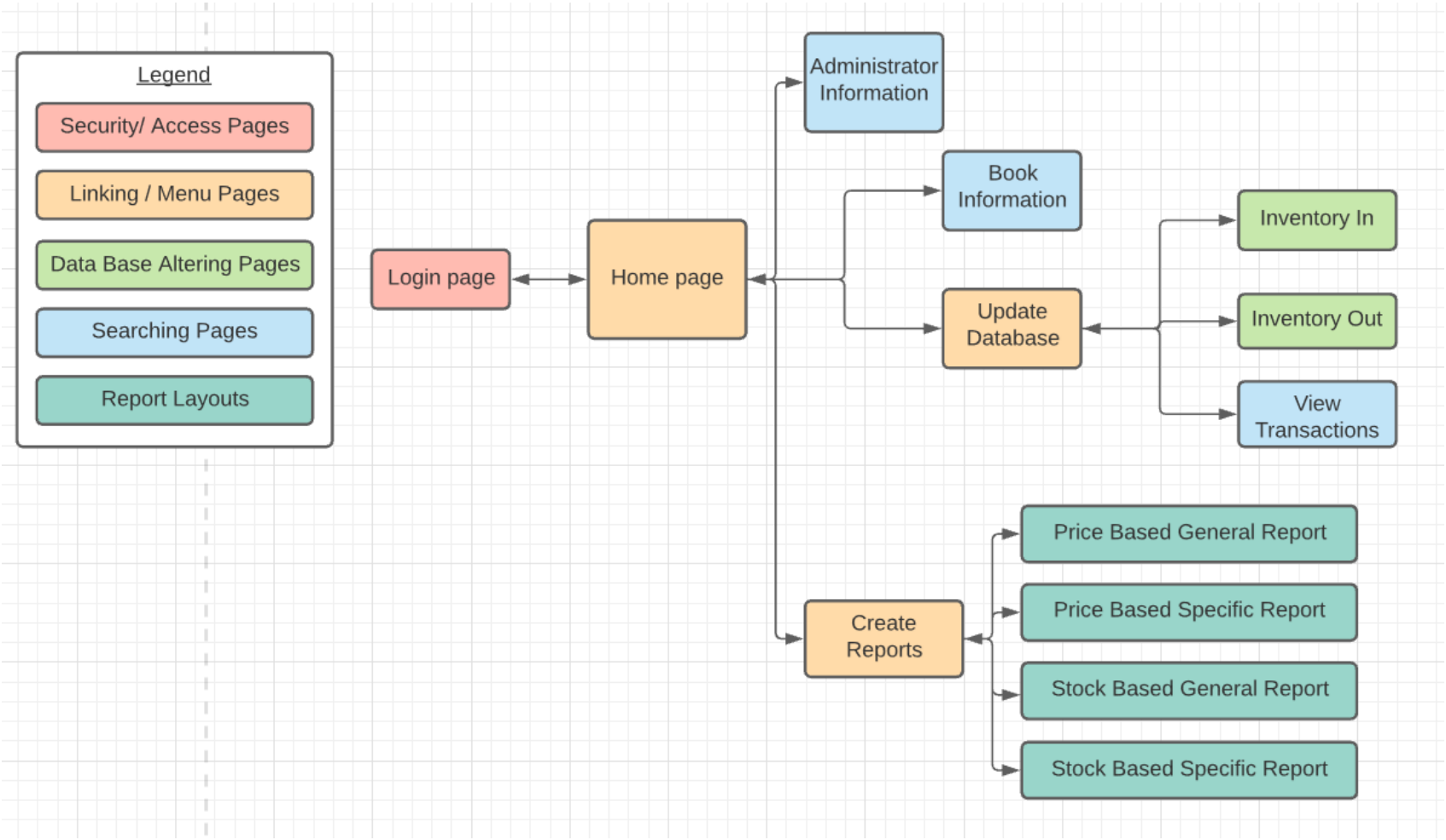
Print


Back
Price-Based-Specific Report
Start Date:
End Date:
ISBN:
Author:
ISBN
Losses
Revenues
Profits
Print


Back
Stock-Based General Report
Start Date:
End Date:
ISBN:
Author:
ISBN
No.Damaged
Inventory In
Inventory Out
Stock Flow
% Damaged
Print


Back
Price-Based General Report
Start Date:
End Date:
ISBN:
Author:
ISBN
Losses
Revenues
Profits
Print

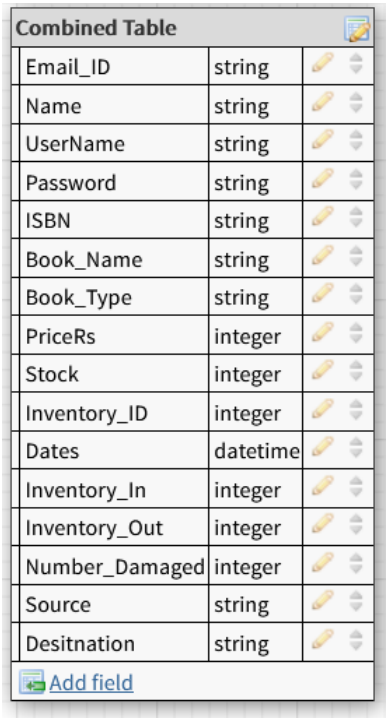
Design Layout[2]:



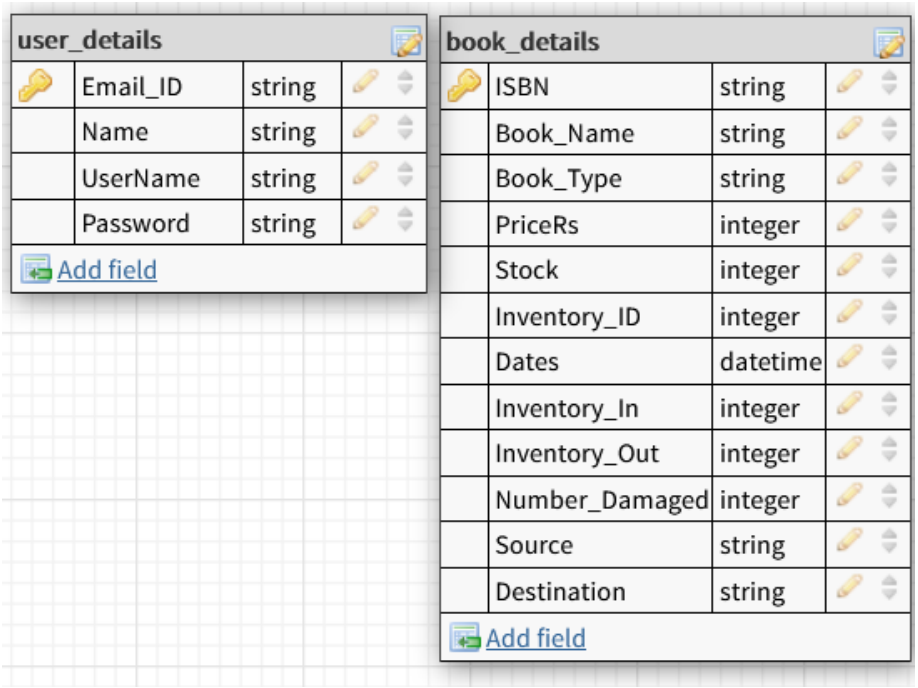
[2] The website "*Lucidchart*" was used to make the UML Diagrams, the design layout, and design overview-https://www.lucidchart.com/pages/

Database Normalization[3]:

Draft 1: [All attributes recorded as captured from details shared by Mr XX]

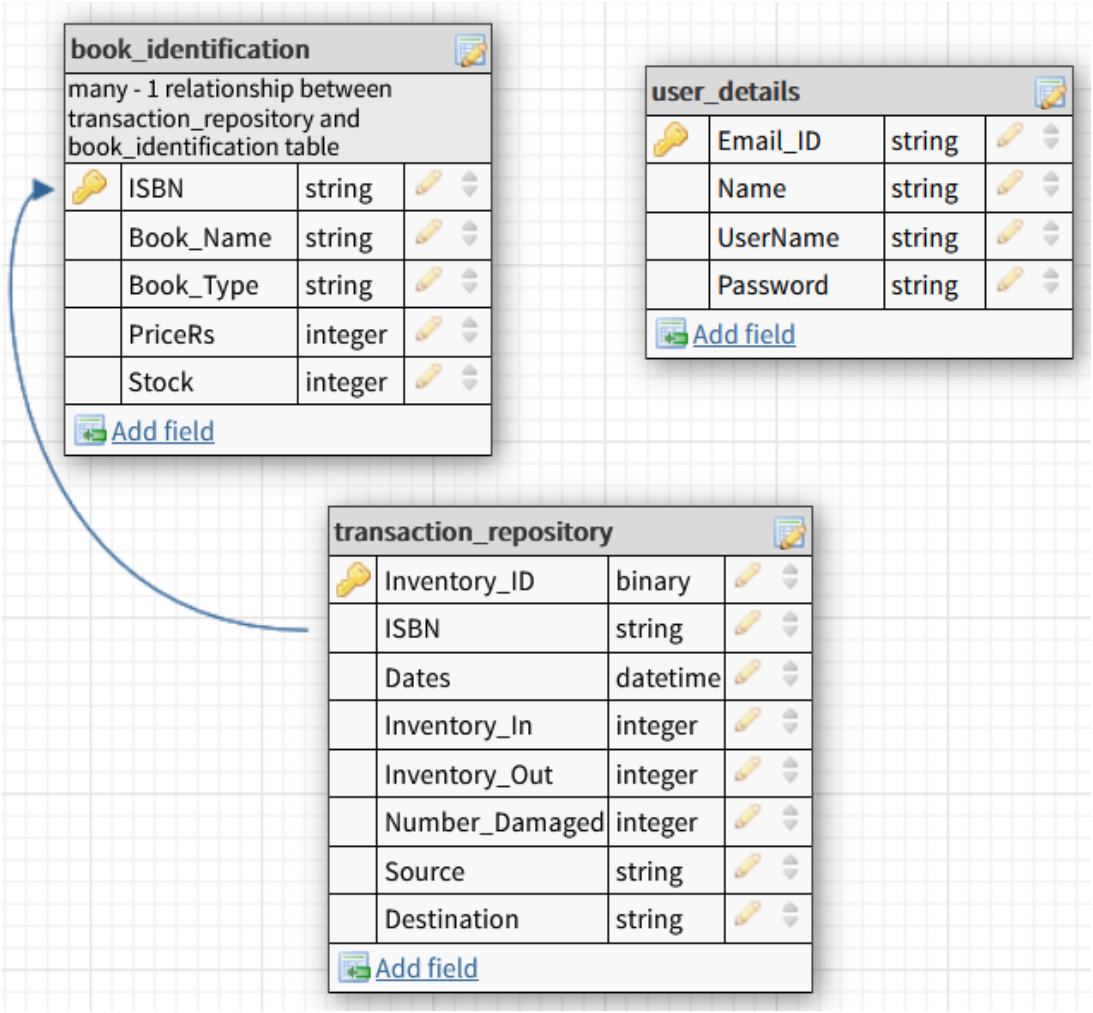


Draft 2:

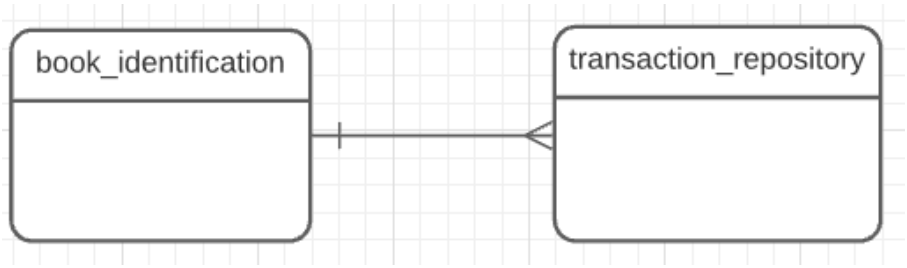


[3]The website "*DB Designer*" was used to the data flow diagrams above- https://www.dbdesigner.net/

Draft 3:

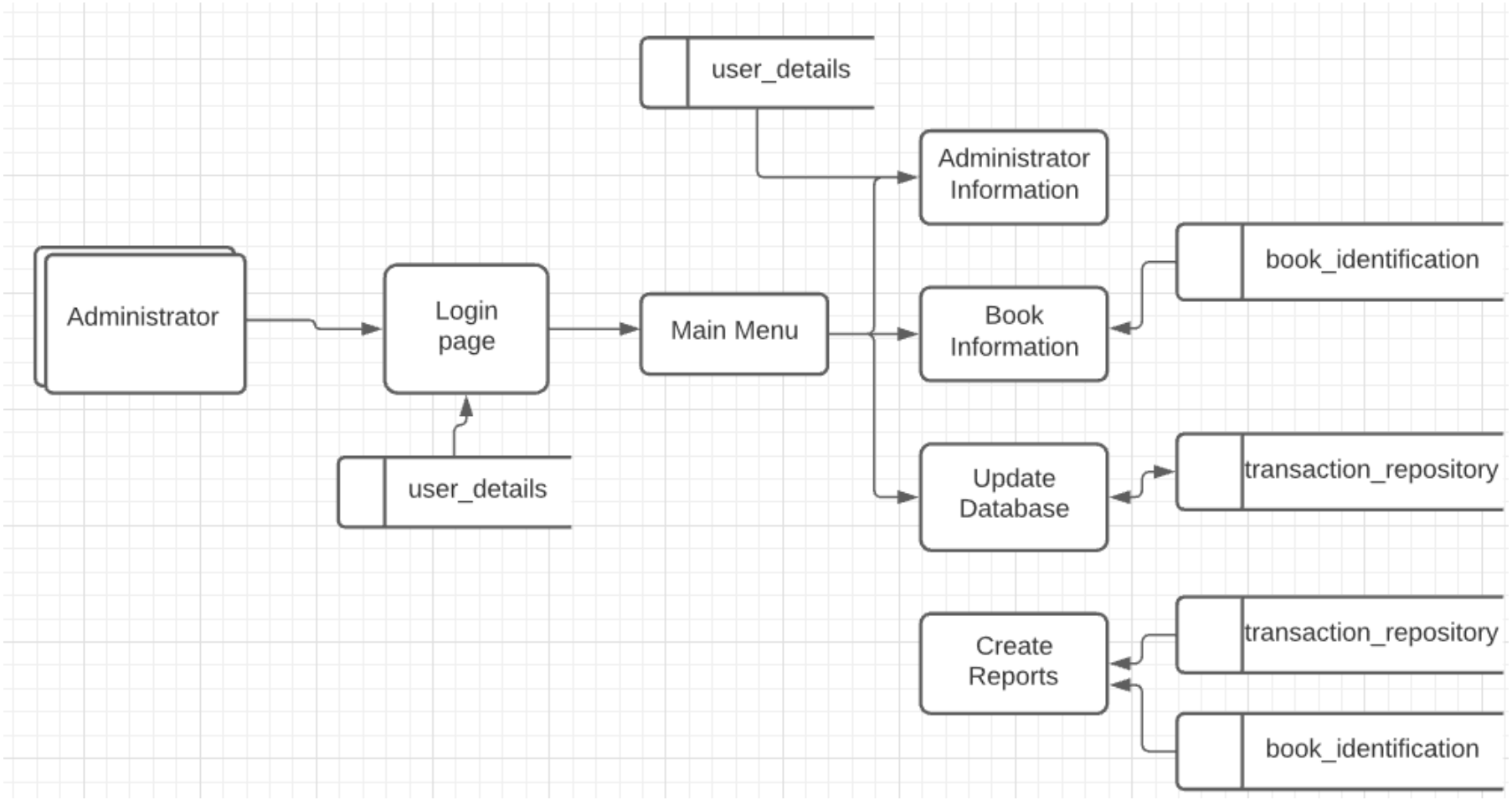


Entity Relationship Diagram

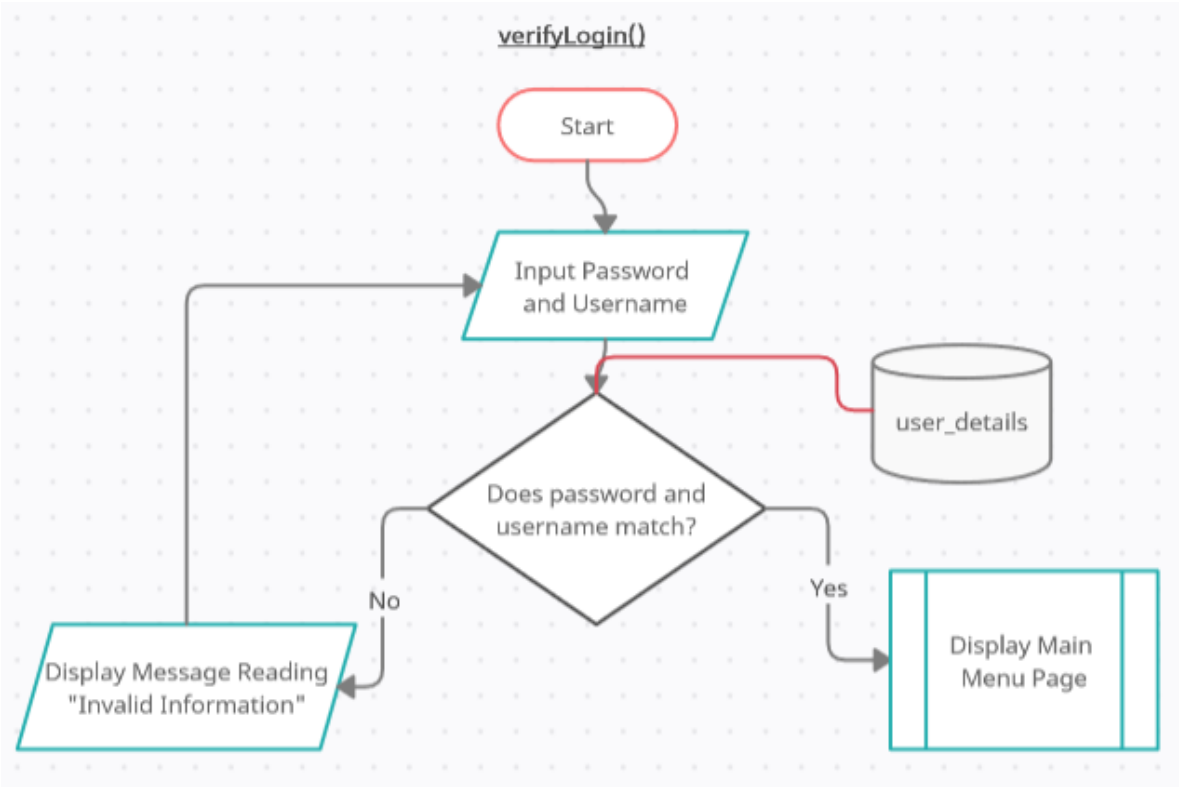


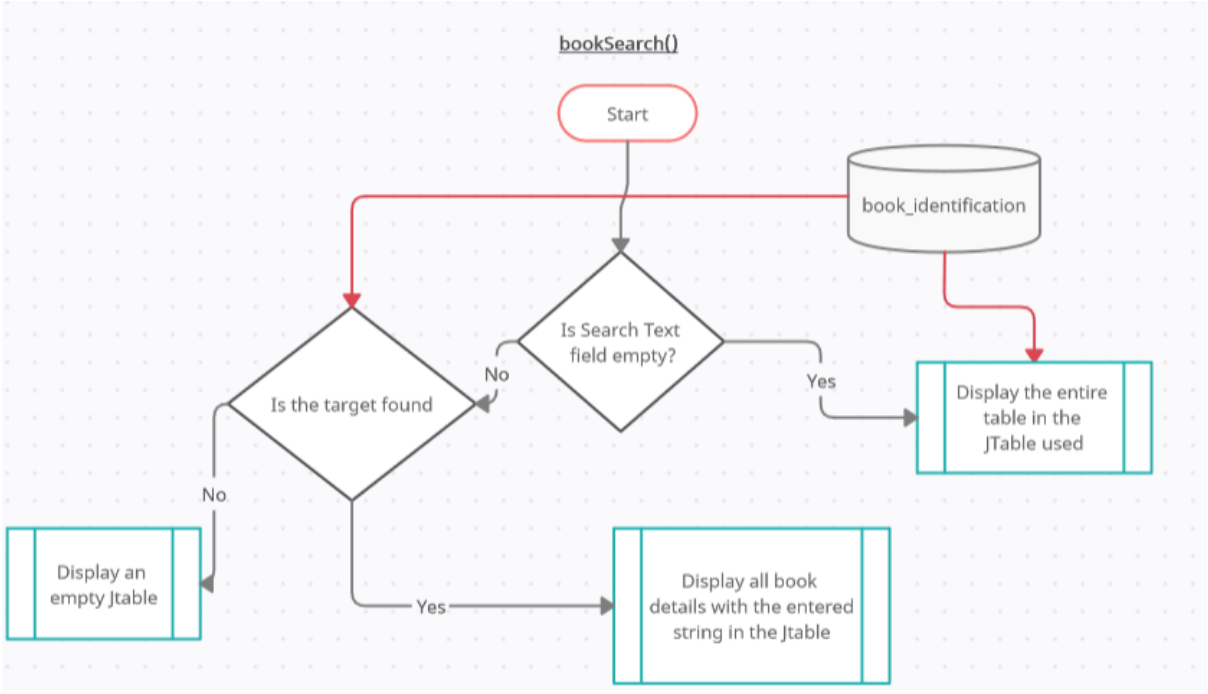
There is a 1 to many relation between book_identification and transaction_repository. One ISBN from book_identification can have many ingoing or outgoing transactions stored in the transaction_repository table.

Data Flow Diagrams:

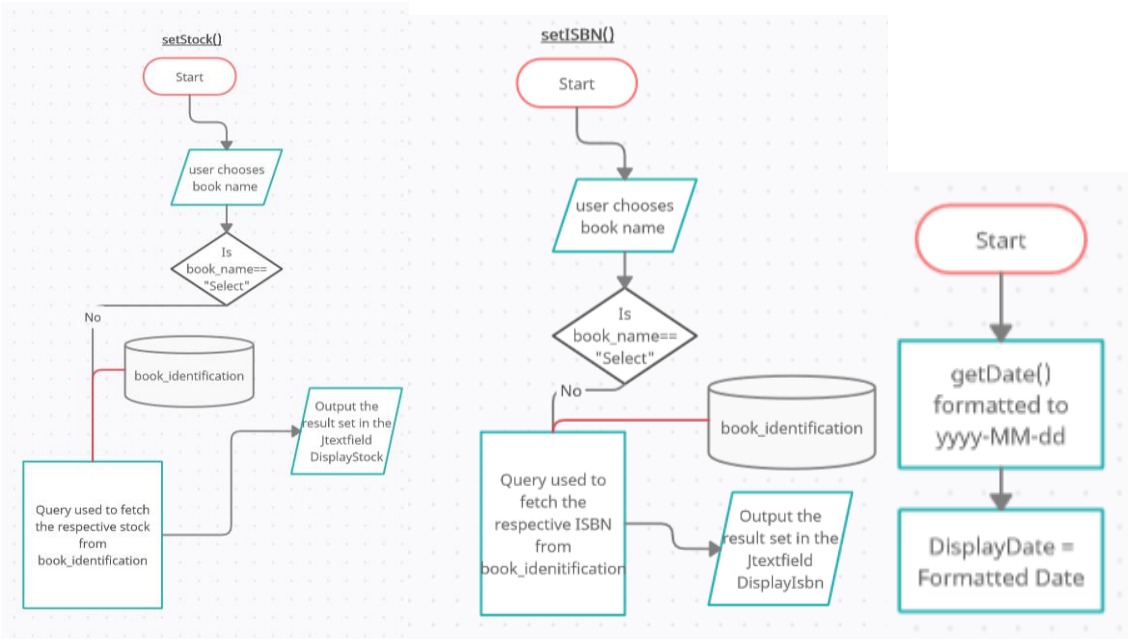


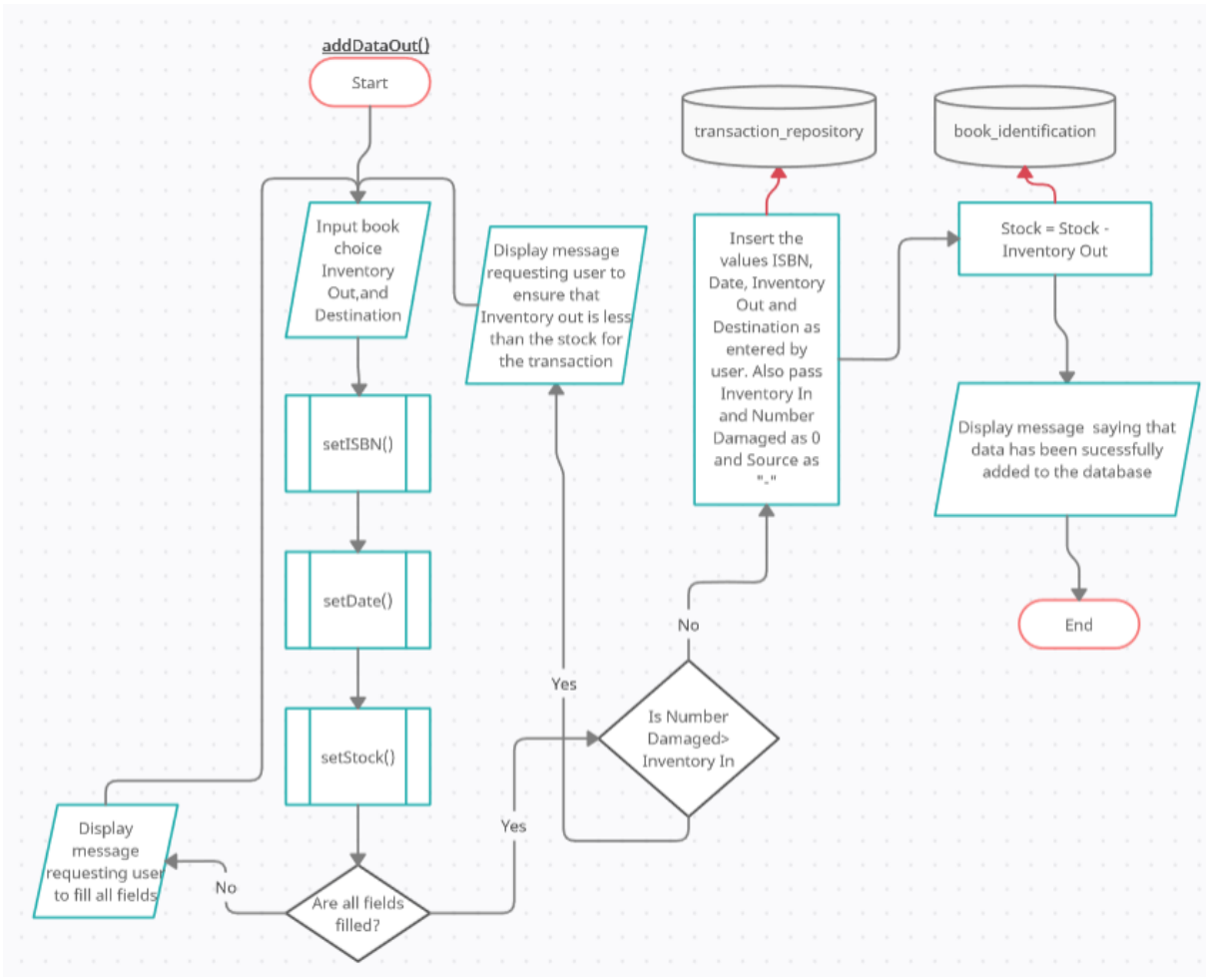
Flowcharts for functions[4]:

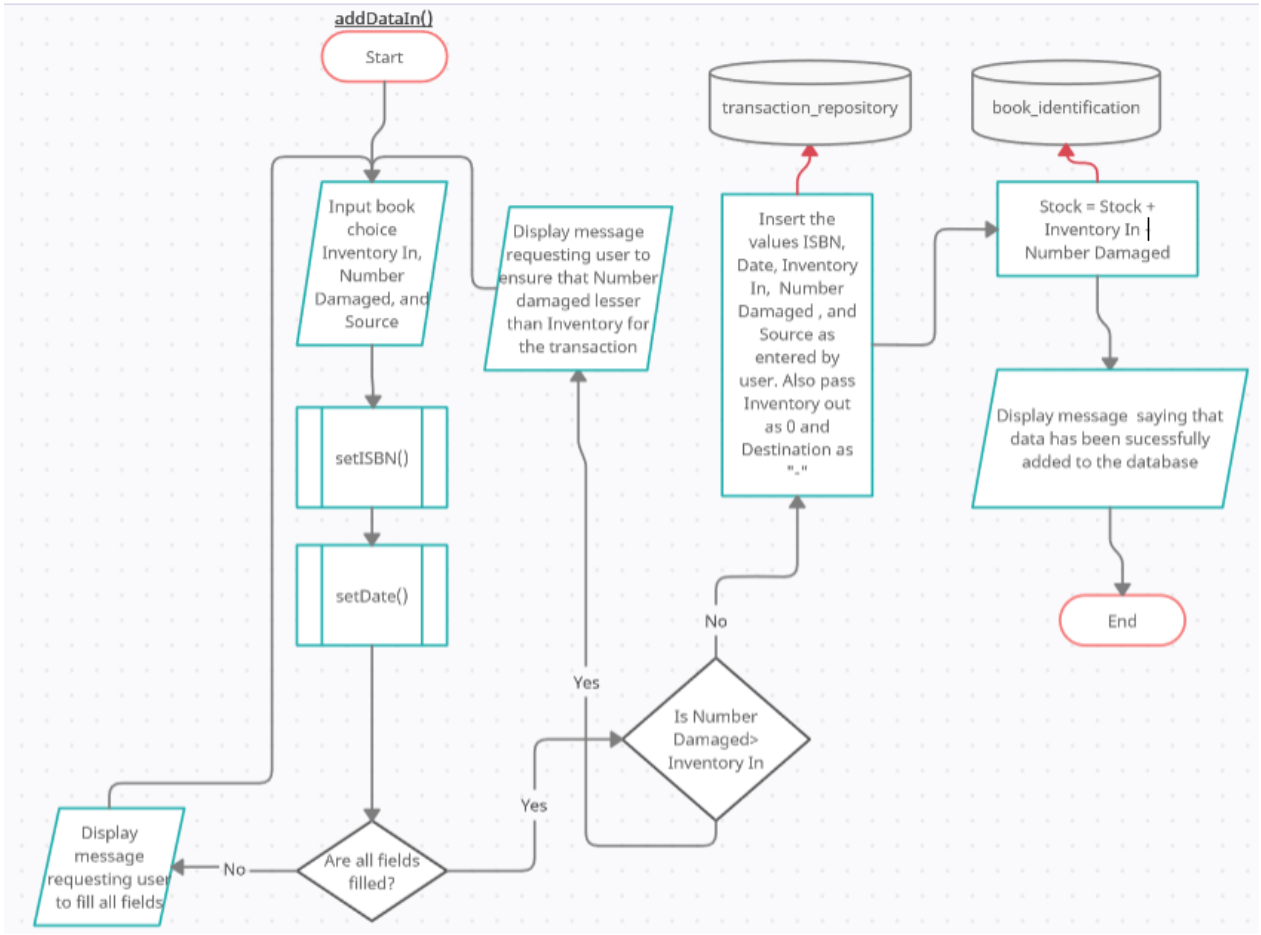


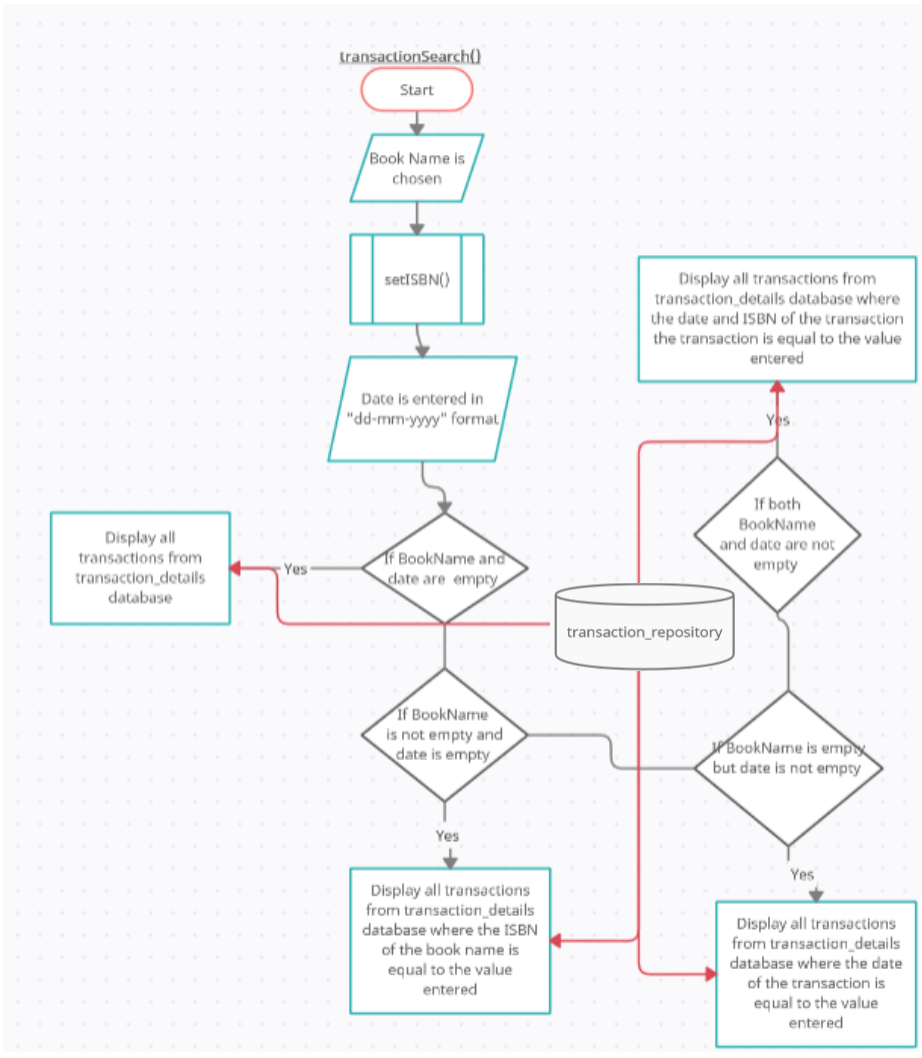


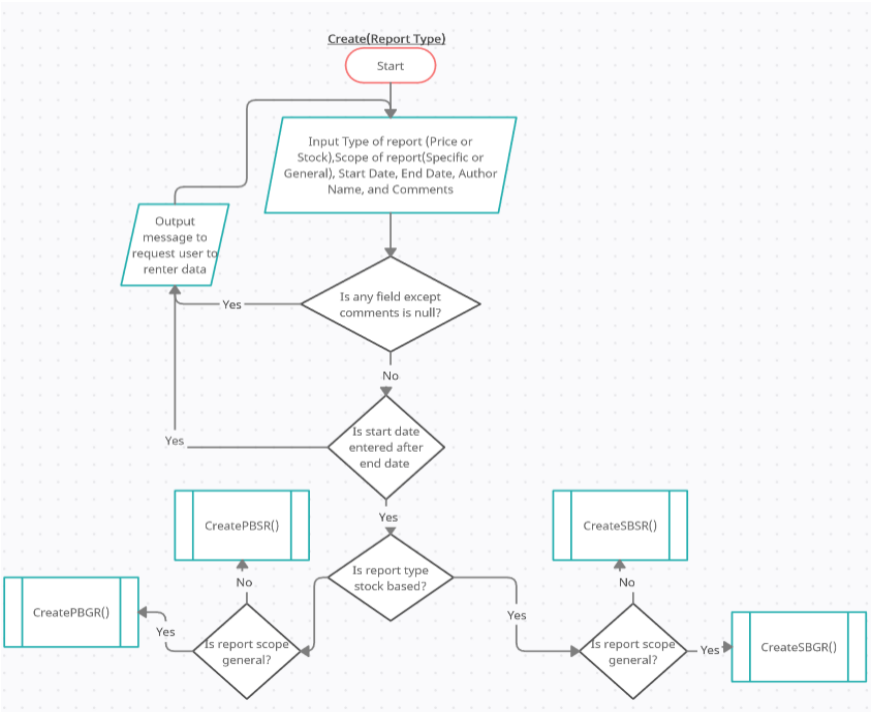
[4] *The website "Creately" was used to the function flowcharts - https://creately.com/*

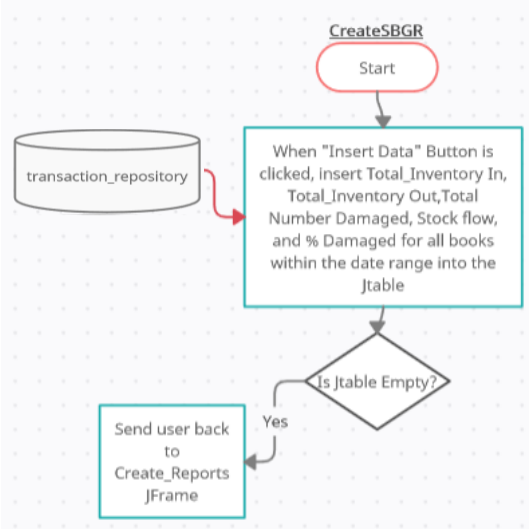

setStock()
Start
user chooses book name
Is book_name== "Select"
No
book_identification
Output the result set in the Jtextfield DisplayStock
Query used to fetch the respective stock from book_identification
setISBN()
Start
user chooses book name
Is book_name== "Select"
No
book_identification
Query used to fetch the respective ISBN from book_idenitification
Output the result set in the Jtextfield DisplayIsbn
Start
getDate() formatted to yyyy-MM-dd
DisplayDate = Formatted Date

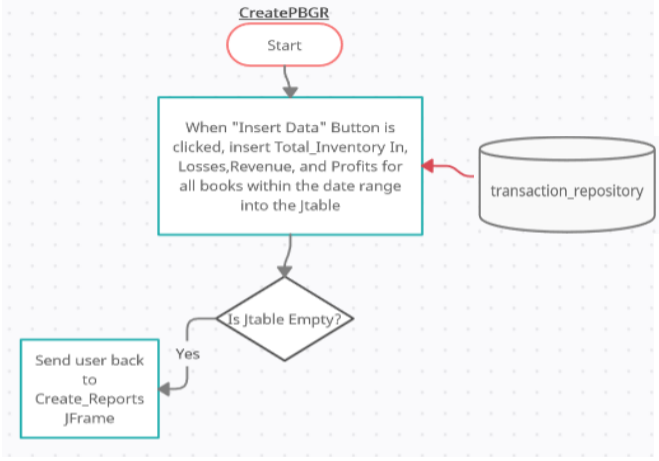

addDataOut()
Start
Input book choice Inventory Out,and Destination
setISBN()
setDate()
setStock()
Are all fields filled?
No
Display message requesting user to fill all fields
Yes
Display message requesting user to ensure that Inventory out is less than the stock for the transaction
Yes
Is Number Damaged> Inventory In
No
Insert the values ISBN, Date, Inventory Out and Destination as entered by user. Also pass Inventory In and Number Damaged as 0 and Source as "-"
transaction_repository
book_identification
Stock = Stock - Inventory Out
Display message saying that data has been sucessfully added to the database
End

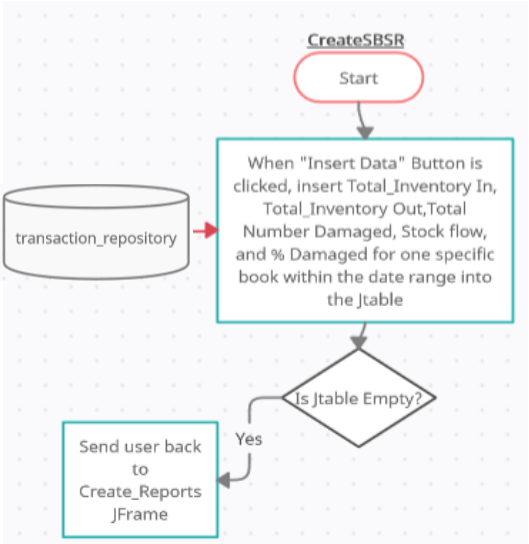

addDataIn()
Start
Input book choice Inventory In, Number Damaged, and Source
setISBN()
setDate()
Are all fields filled?
No
Display message requesting user to fill all fields
Yes
Display message requesting user to ensure that Number damaged lesser than Inventory for the transaction
Yes
Is Number Damaged> Inventory In
No
Insert the values ISBN, Date, Inventory In, Number Damaged , and Source as entered by user. Also pass Inventory out as 0 and Destination as "-"
transaction_repository
Stock = Stock + Inventory In - Number Damaged
book_identification
Display message saying that data has been sucessfully added to the database
End

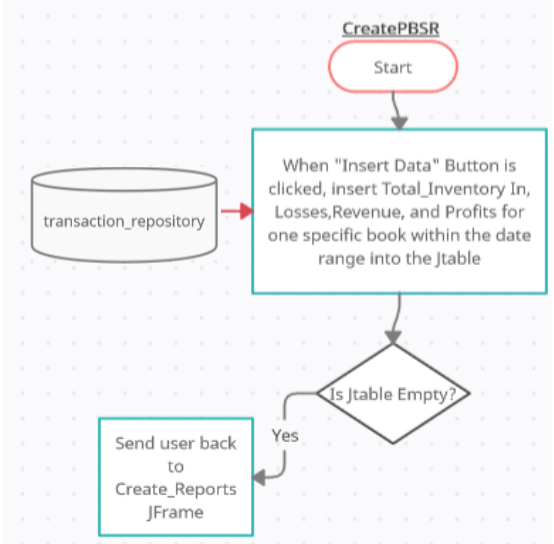

transactionSearch()
Start
Book Name is chosen
setISBN()
Date is entered in "dd-mm-yyyy" format
If BookName and date are empty
Yes
Display all transactions from transaction_details database
If BookName is not empty and date is empty
Yes
Display all transactions from transaction_details database where the ISBN of the book name is equal to the value entered
If both BookName and date are not empty
Yes
Display all transactions from transaction_details database where the date and ISBN of the transaction the transaction is equal to the value entered
If BookName is empty but date is not empty
Yes
Display all transactions from transaction_details database where the date of the transaction is equal to the value entered
transaction_repository


Create(Report Type)
Start
Input Type of report (Price or Stock),Scope of report(Specific or General), Start Date, End Date, Author Name, and Comments
Output message to request user to renter data
Is any field except comments is null?
Yes
No
Is start date entered after end date
Yes
Yes
CreatePBSR()
CreateSBSR()
Is report type stock based?
No
No
CreatePBGR()
Yes
Is report scope general?
Yes
Is report scope general?
Yes
CreateSBGR()


CreateSBGR
Start
transaction_repository
When "Insert Data" Button is clicked, insert Total_Inventory In, Total_Inventory Out,Total Number Damaged, Stock flow, and % Damaged for all books within the date range into the Jtable
Is Jtable Empty?
Yes
Send user back to Create_Reports JFrame

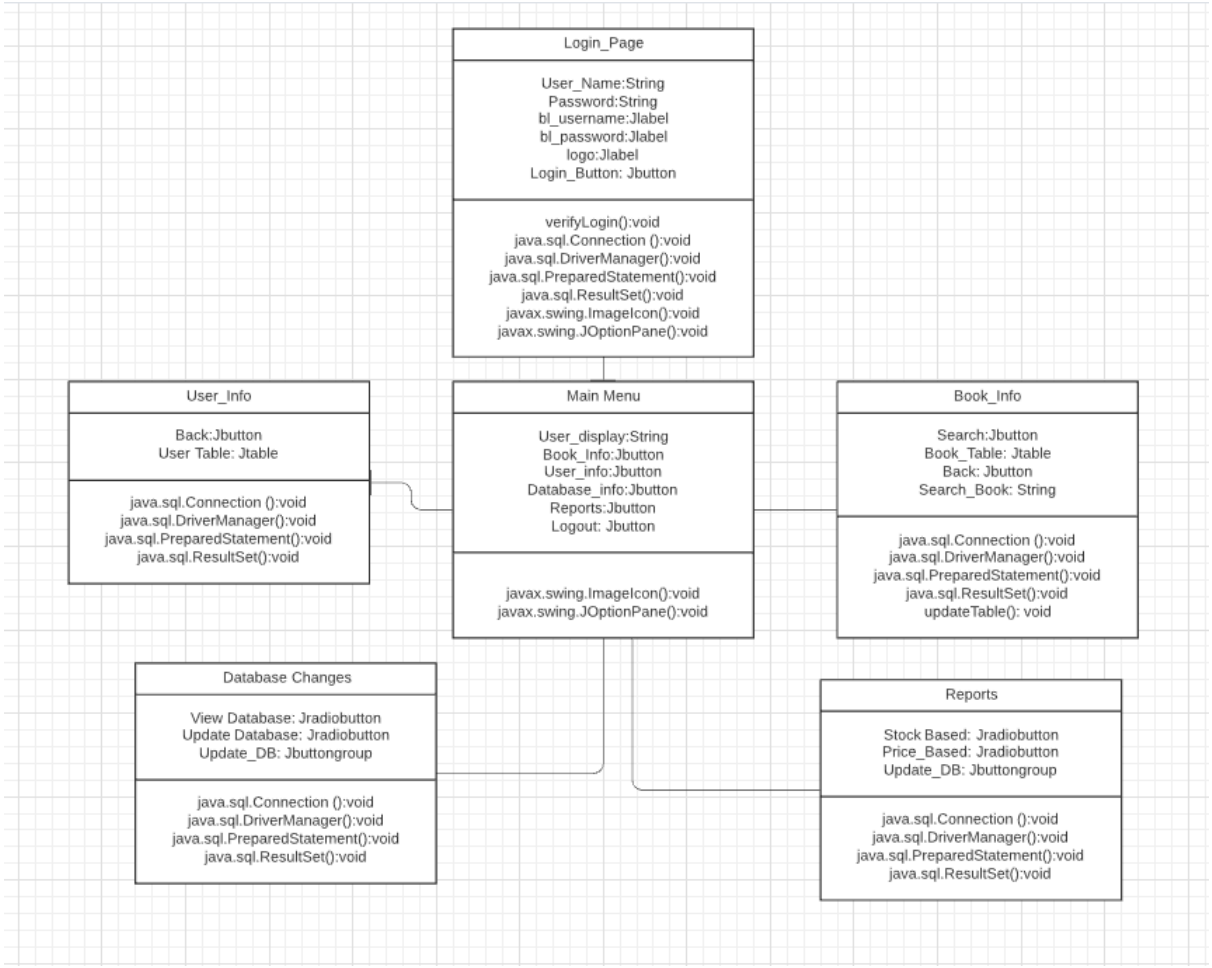

CreatePBGR
Start
When "Insert Data" Button is clicked, insert Total_Inventory In, Losses,Revenue, and Profits for all books within the date range into the Jtable
transaction_repository
Is Jtable Empty?
Yes
Send user back to Create_Reports JFrame


CreateSBSR
Start
transaction_repository
When "Insert Data" Button is clicked, insert Total_Inventory In, Total_Inventory Out,Total Number Damaged, Stock flow, and % Damaged for one specific book within the date range into the Jtable
Is Jtable Empty?
Yes
Send user back to Create_Reports JFrame

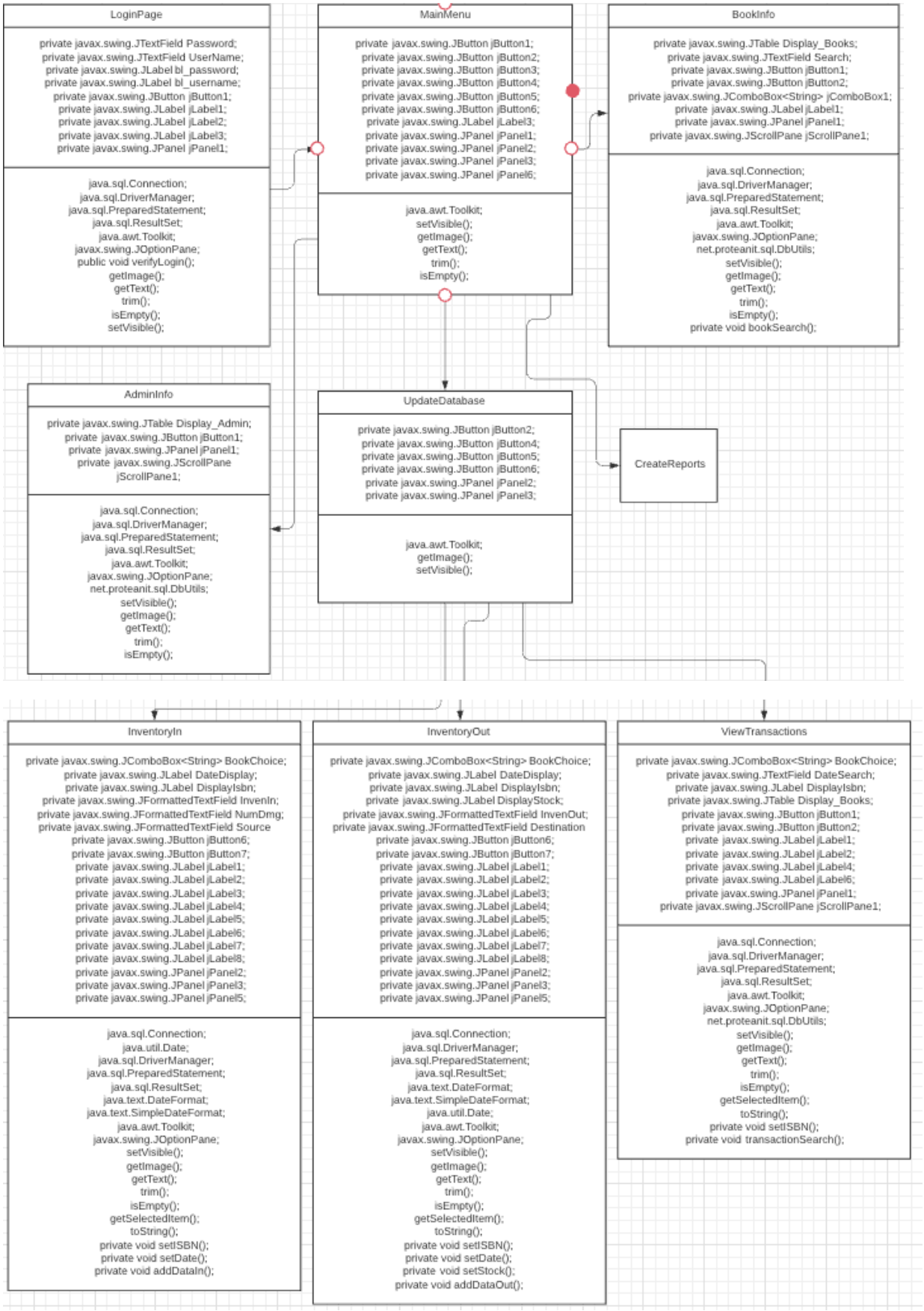

CreatePBSR
Start
transaction_repository
When "Insert Data" Button is clicked, insert Total_Inventory In, Losses,Revenue, and Profits for one specific book within the date range into the Jtable
Is Jtable Empty?
Yes
Send user back to Create_Reports JFrame

UML diagrams:

Draft 1:



Draft 2:


LoginPage
private javax.swing.JTextField Password;
private javax.swing.JTextField UserName;
private javax.swing.JLabel bl_password;
private javax.swing.JLabel bl_username;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel1;
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.awt.Toolkit;
javax.swing.JOptionPane;
public void verifyLogin();
getImage();
getText();
trim();
isEmpty();
setVisible();
MainMenu
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel6;
java.awt.Toolkit;
setVisible();
getImage();
getText();
trim();
isEmpty();
BookInfo
private javax.swing.JTable Display_Books;
private javax.swing.JTextField Search;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JComboBox<String> jComboBox1;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.awt.Toolkit;
javax.swing.JOptionPane;
net.proteanit.sql.DbUtils;
setVisible();
getImage();
getText();
trim();
isEmpty();
private void bookSearch();
AdminInfo
private javax.swing.JTable Display_Admin;
private javax.swing.JButton jButton1;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.awt.Toolkit;
javax.swing.JOptionPane;
net.proteanit.sql.DbUtils;
setVisible();
getImage();
getText();
trim();
isEmpty();
UpdateDatabase
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
java.awt.Toolkit;
getImage();
setVisible();
CreateReports
InventoryIn
private javax.swing.JComboBox<String> BookChoice;
private javax.swing.JLabel DateDisplay;
private javax.swing.JLabel DisplayIsbn;
private javax.swing.JFormattedTextField InvenIn;
private javax.swing.JFormattedTextField NumDmg;
private javax.swing.JFormattedTextField Source
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel5;
java.sql.Connection;
java.util.Date;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.text.DateFormat;
java.text.SimpleDateFormat;
java.awt.Toolkit;
javax.swing.JOptionPane;
setVisible();
getImage();
getText();
trim();
isEmpty();
getSelectedItem();
toString();
private void setISBN();
private void setDate();
private void addDataIn();
InventoryOut
private javax.swing.JComboBox<String> BookChoice;
private javax.swing.JLabel DateDisplay;
private javax.swing.JLabel DisplayIsbn;
private javax.swing.JLabel DisplayStock;
private javax.swing.JFormattedTextField InvenOut;
private javax.swing.JFormattedTextField Destination
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel5;
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.text.DateFormat;
java.text.SimpleDateFormat;
java.util.Date;
java.awt.Toolkit;
javax.swing.JOptionPane;
setVisible();
getImage();
getText();
trim();
isEmpty();
getSelectedItem();
toString();
private void setISBN();
private void setDate();
private void setStock();
private void addDataOut();
ViewTransactions
private javax.swing.JComboBox<String> BookChoice;
private javax.swing.JTextField DateSearch;
private javax.swing.JLabel DisplayIsbn;
private javax.swing.JTable Display_Books;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.awt.Toolkit;
javax.swing.JOptionPane;
net.proteanit.sql.DbUtils;
setVisible();
getImage();
getText();
trim();
isEmpty();
getSelectedItem();
toString();
private void setISBN();
private void transactionSearch();

Create_Reports
private javax.swing.JComboBox<String> BookChoice;
public static javax.swing.JTextArea Comments;
public static javax.swing.JLabel DisplayIsbn;
public static javax.swing.JFormattedTextField EndDate;
public static javax.swing.JComboBox<String> NameChoice;
private javax.swing.JComboBox<String> ReportChoice;
public static javax.swing.JFormattedTextField StartDate;
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel5;
private javax.swing.JScrollPane jScrollPane1;
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
javax.swing.JOptionPane;
java.text.ParseException;
java.text.SimpleDateFormat;
java.util.Date;
java.awt.Toolkit;
getImage();
java.util.logging.Level;
java.util.logging.Logger;
setVisible();
getText();
setText();
setISBN();
StockBasedGeneralReport
private javax.swing.JLabel DateDisplay;
public static javax.swing.JTextArea DisplayComments;
public static javax.swing.JLabel DisplayEndDate;
public static javax.swing.JLabel DisplayStartDate;
private javax.swing.JTable Display_Report;
public static javax.swing.JLabel Name;
private javax.swing.JButton PrintButton;
private javax.swing.JPanel Report_Print;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton4;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
java.awt.Graphics;
java.awt.Graphics2D;
java.awt.print.PageFormat;
java.awt.print.Printable;
java.awt.print.PrinterException;
java.awt.print.PrinterJob;
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.text.DateFormat;
java.text.SimpleDateFormat;
java.util.Date;
javax.swing.JOptionPane;
net.proteanit.sql.DbUtils;
java.awt.Toolkit;
getImage();
getText();
setVisible();
setDate();
CreateSBGR();
StockBasedSpecificReport
private javax.swing.JLabel DateDisplay;
public static javax.swing.JTextArea DisplayComments;
public static javax.swing.JLabel DisplayEndDate;
public static javax.swing.JLabel DisplayStartDate;
private javax.swing.JTable Display_Report;
public static javax.swing.JLabel Name;
private javax.swing.JButton PrintButton;
private javax.swing.JPanel Report_Print;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton4;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
java.awt.Graphics;
java.awt.Graphics2D;
java.awt.print.PageFormat;
java.awt.print.Printable;
java.awt.print.PrinterException;
java.awt.print.PrinterJob;
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.text.DateFormat;
java.text.SimpleDateFormat;
java.util.Date;
javax.swing.JOptionPane;
net.proteanit.sql.DbUtils;
java.awt.Toolkit;
getImage();
getText();
setVisible();
setDate();
CreateSBSR();

PriceBasedSpecficReport
private javax.swing.JLabel DateDisplay;
public static javax.swing.JTextArea DisplayComments;
public static javax.swing.JLabel DisplayEndDate;
public static javax.swing.JLabel DisplayStartDate;
private javax.swing.JTable Display_Report;
public static javax.swing.JLabel Name;
private javax.swing.JButton PrintButton;
private javax.swing.JPanel Report_Print;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton4;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
java.awt.Graphics;
java.awt.Graphics2D;
java.awt.print.PageFormat;
java.awt.print.Printable;
java.awt.print.PrinterException;
java.awt.print.PrinterJob;
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.text.DateFormat;
java.text.SimpleDateFormat;
java.util.Date;
javax.swing.JOptionPane;
net.proteanit.sql.DbUtils;
java.awt.Toolkit;
getImage();
getText();
setVisible();
setDate();
CreatePBGR();
PriceBasedGeneralReport
private javax.swing.JLabel DateDisplay;
public static javax.swing.JTextArea DisplayComments;
public static javax.swing.JLabel DisplayEndDate;
public static javax.swing.JLabel DisplayStartDate;
private javax.swing.JTable Display_Report;
public static javax.swing.JLabel Name;
private javax.swing.JButton PrintButton;
private javax.swing.JPanel Report_Print;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton4;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
java.awt.Graphics;
java.awt.Graphics2D;
java.awt.print.PageFormat;
java.awt.print.Printable;
java.awt.print.PrinterException;
java.awt.print.PrinterJob;
java.sql.Connection;
java.sql.DriverManager;
java.sql.PreparedStatement;
java.sql.ResultSet;
java.text.DateFormat;
java.text.SimpleDateFormat;
java.util.Date;
javax.swing.JOptionPane;
net.proteanit.sql.DbUtils;
java.awt.Toolkit;
getImage();
getText();
setVisible();
setDate();
CreatePBSR();

Test Plan:

Accepted outcomes:

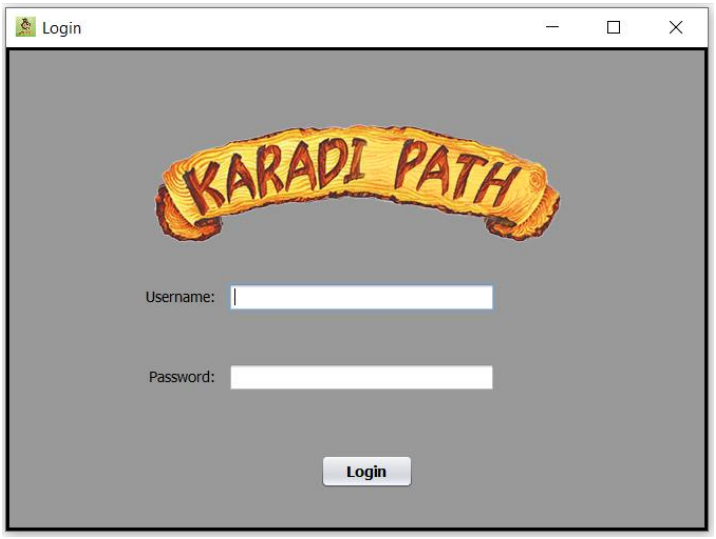
<table>
<tr><td>Program</td><td>Accepted Data</td></tr>
<tr><td>Login Page</td><td>Any of the pairs in the below table must be matched (UserName and Password)</td></tr>
<tr><td>Test Plan 1</td><td rowspan="5"><table><tr><th>Name</th><th>Email_ID</th><th>UserName</th><th>Password</th></tr><tr><td>Abubucker. M</td><td>abubucker.m@karadipath.com</td><td>abubucker</td><td>abubucker@123</td></tr><tr><td>Gopinath. D</td><td>gopinath.d@karadipath.com</td><td>gopinath</td><td>gopinath@123</td></tr><tr><td>Kripakaran. P</td><td>kripakaran.p@karadipath.com</td><td>kripakaran</td><td>kripakaran@123</td></tr><tr><td>Sathiskumar. M</td><td>sathiskumar.m@karadipath.com</td><td>sathiskumar</td><td>sathiskumar@123</td></tr></table></td></tr>
<tr><td></td></tr>
<tr><td></td></tr>
<tr><td></td></tr>
<tr><td></td></tr>
<tr><td>Book Search Page</td><td>Any of the names of the 70 books (case sensitive) can be entered and the table has to reflect all the data on it</td></tr>
<tr><td>Test Plan 2</td><td>Example:" The Monkey King" will be accepted and the corresponding details will be returned</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Inventory_In Page</td><td>Inventory_In > Number Damaged, Book chosen combo box is not "select"</td></tr>
<tr><td>Test Plan 3</td><td>Inventory In: 68</td></tr>
<tr><td></td><td>Number Damaged: 5</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Inventory_Out Page</td><td>Inventory_Out < Stock, Book chosen combo box is not "select"</td></tr>
<tr><td>Test Plan 4</td><td>Stock: 365</td></tr>
<tr><td></td><td>Inventory_out: 240</td></tr>
<tr><td></td><td></td></tr>
<tr><td>View_Transactions</td><td>Date is in format of "yyyy--MM-dd"</td></tr>
<tr><td>Test Plan 5</td><td>Date: 2020-01-04 is accepted</td></tr>
<tr><td></td><td></td></tr>
<tr><td>All Report Sections</td><td>Start and End Date is in format of "yyyy--MM-dd"</td></tr>
<tr><td>Test Plan 6</td><td>Date: 2020-01-04 is accepted</td></tr>
<tr><td></td><td>Start Date should be before End date</td></tr>
<tr><td></td><td>Start Date = 2020-12-12, End date = 2020-12-14 will be accepted</td></tr>
<tr><td></td><td>None of the combo boxes are "select" and start date and end date are not empty</td></tr>
<tr><td></td><td>Report choice, Book Choice = "General", NameChoice= "Abubucker”,
Start Date = 2020-12-12,
End date = 2020-12-14 will be accepted</td></tr>
</table>

Rejected outcomes:

Program	**Rejected Data**
Login Page	**Any other input will be rejected {if fields are empty, users will be prompted to enter the data again}**
Test Plan 1	Example username = “xyz” and password = “xyz” will be rejected
Book Search Page	**Empty fields will return the entire book_identification table and anything with special characters is rejected**
Test Plan 2	*" " will return the entire table - "Let's" will return a null table- "the Monkey king" will return an empty table*
Inventory_In Page	**Any entry in which number damaged > Inventory_In is rejected (even empty fields)**
Test Plan 3	*Inventory In: 68 or empty*
	Number_Damaged: 72 or empty
Inventory_Out Page	**Any entry in which Inventory_Out> Stock is rejected (even empty fields)**
Test Plan 4	*Stock: 365 or empty*
	Inventory_out: 400 or empty
View_Transactions	**Any date not written in the format " yyyy-MM-dd" is rejected**
Test Plan 5	*Date: 04-01-2020 is not accepted*
All Report Sections	**Any date not written in the format "yyyy-MM-dd" is rejected**
Test Plan 6	*Date: 04-01-2020 is not accepted*
	Start Date is after End date
	Start Date = 2020-12-14, End date = 2020-12-12 will be rejected
	One of the combo boxes are "select" and start date or end date is empty

Final software layouts (Examples)

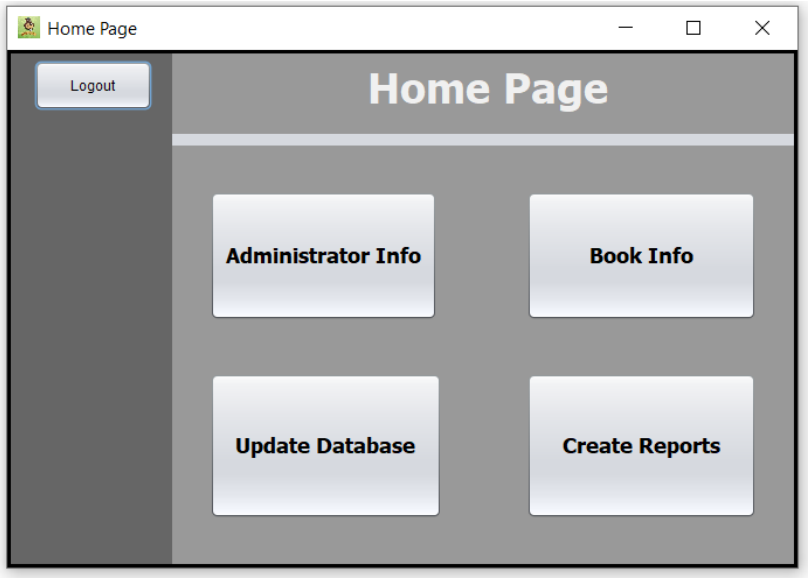
Login Page



- The administrator details, book details, and transaction repository tables will be loaded from MySQL schema when the program is opened.

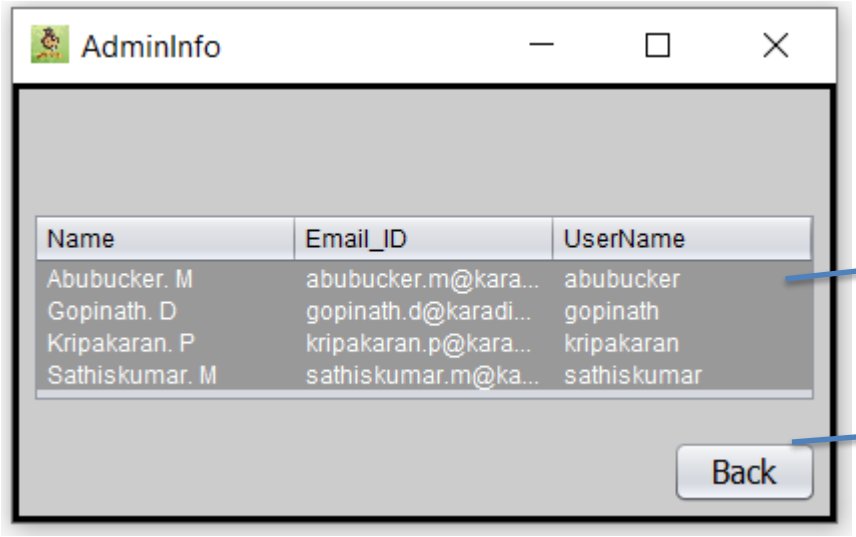
- Each administrator will have a username and password to enter the software

Home Page



- Administrators can view:
  - Administrator Info - shows a table containing the details of each administrator
  - Book Info - A repository of book details which can be filtered by entering a particular title's name
  - Update Database -administrators can add incoming and outgoing transactions into the database and view each of these transactions
  - Create Reports - administrators can create summaries of based on the price or the stock of a title or every title in the warehouse over a period of time

Administrator Information

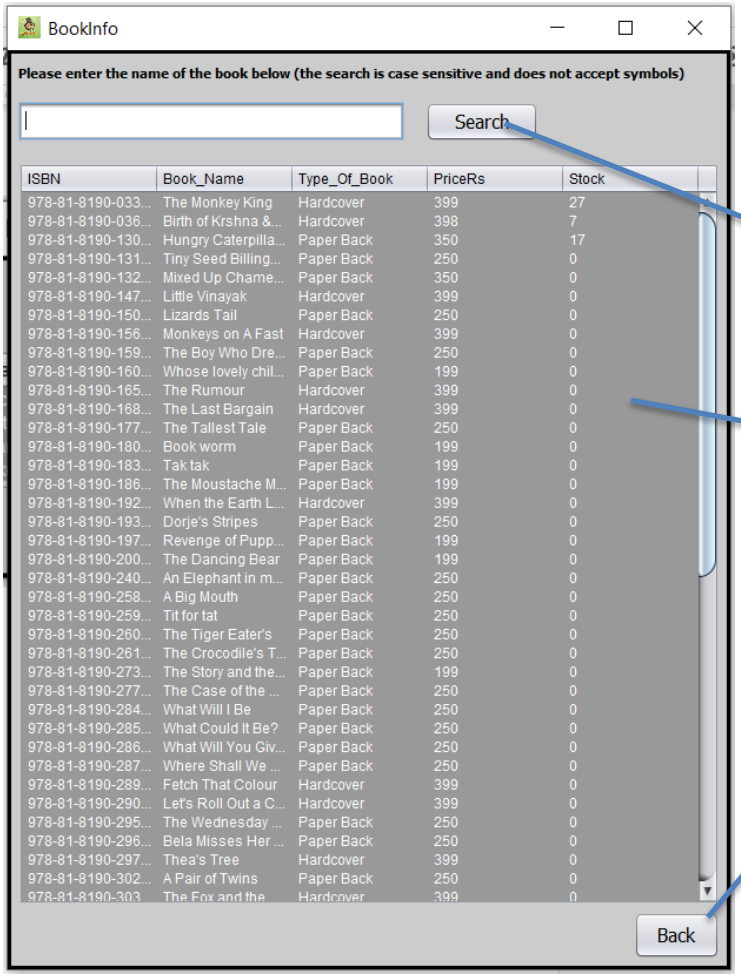


- Emails can be copied from this repository so that they can send and receive the reports they create

Table containing Name, Email ID, and User name of each administrator

Takes admin back to home page

Book Information



- The stock, and other details, of any title can be found by entering the title's name in the search parameter

When button is clicked the program will fetch the data from the book_identification table where the Book_Name is LIKE the search parameter

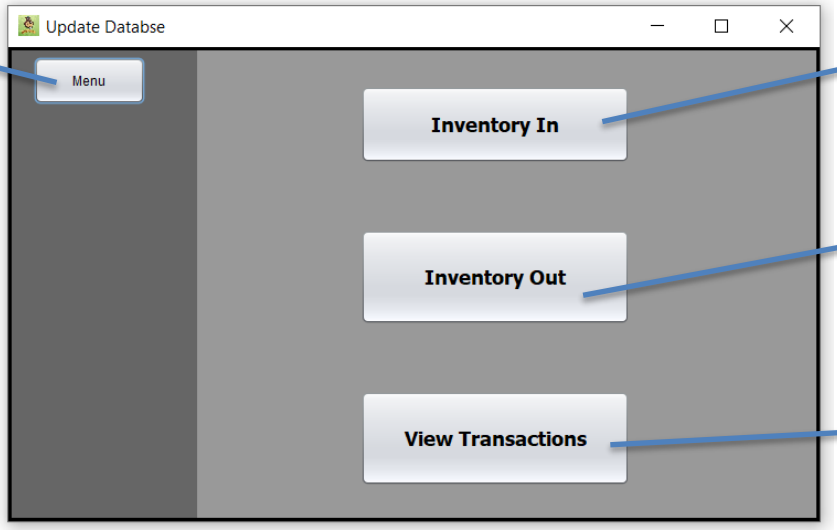
Scrollable table which displays the fetched data according to the search parameter

- For example, if the search parameter was equal to the word "The", the details of any title with the phrase "The" in it will be returned

Takes admin back to home page

Update Database

Takes admin back to home page



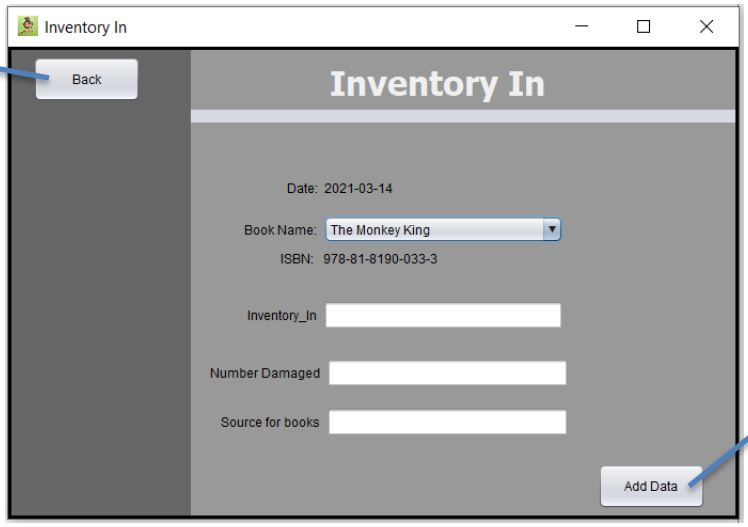
This page provides the functionality to enter the details of an incoming stock

This page provides the functionality to enter the details of an outgoing stock

This page allows the user to view and search the transaction repository with two search parameters

Inventory In

Takes admin back to update database page

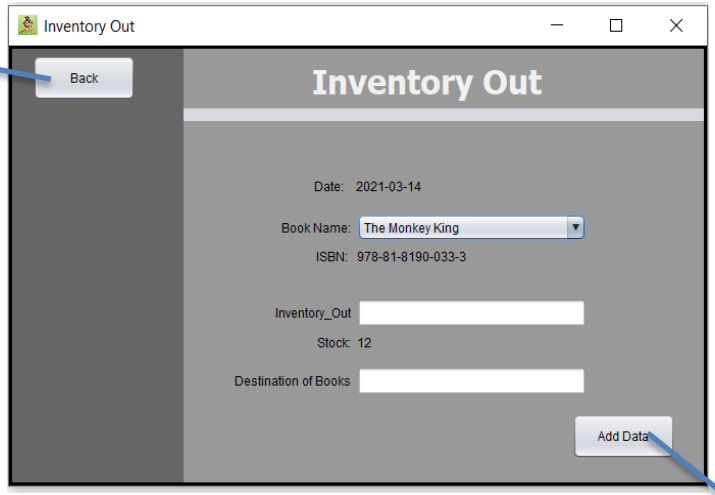


- Date is automatically captured when the page is opened.
- ISBN of the book is captured below the combo box (BookChoice) when a book is chosen

The entered Inventory In, Number Damaged, and source for books are added to the transaction_repository table and stock is updated in book_identification table provided the data entered follows test plan 3

Inventory Out

Takes admin back to update database page

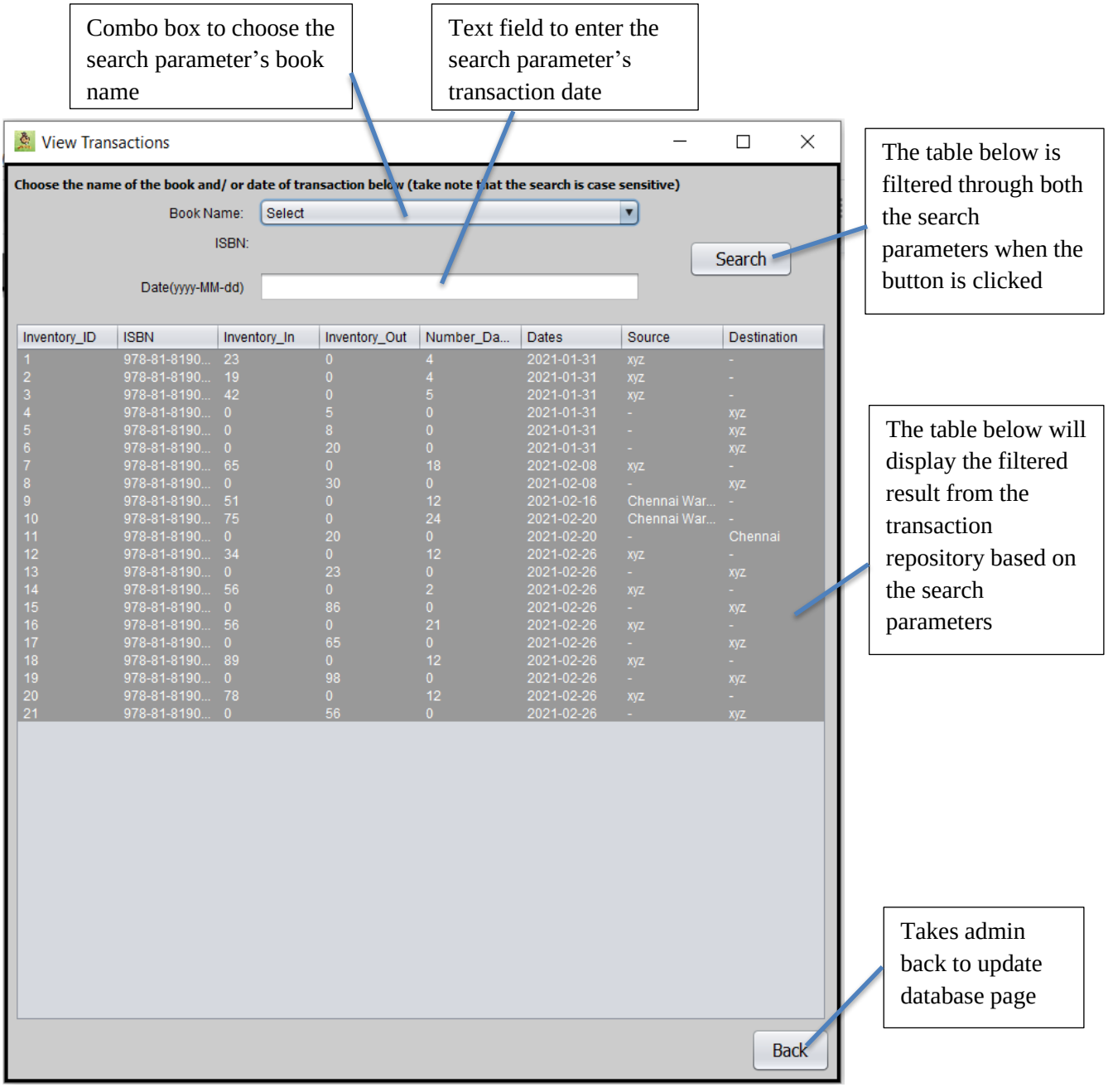


- Date is automatically captured when the page is opened.
- ISBN of the book is captured below the combo box (BookChoice) when a book is chosen and stock of the book is captured and displayed below the Inventory Out Text field

The entered Inventory Out, and Destination for books are added to the transaction_repository table and stock is updated in book_details table provided the data entered follows test plan 4

View Transactions

- This page provides the functionality to search the transaction repository with 2 different search parameters – namely the title name/ book name and or the date of the transaction
- Hence there are 3 possible searches in this context
    - Filtering the table to get the rows where the transactions' title is equal to the search parameter
    - Filtering the table to get the rows where the transactions' date is equal to the search parameter
    - Filtering the table to get the rows where the transactions' title is equal to the search parameter
- ISBN of the book is captured below the combo box (BookChoice) when a book is chosen

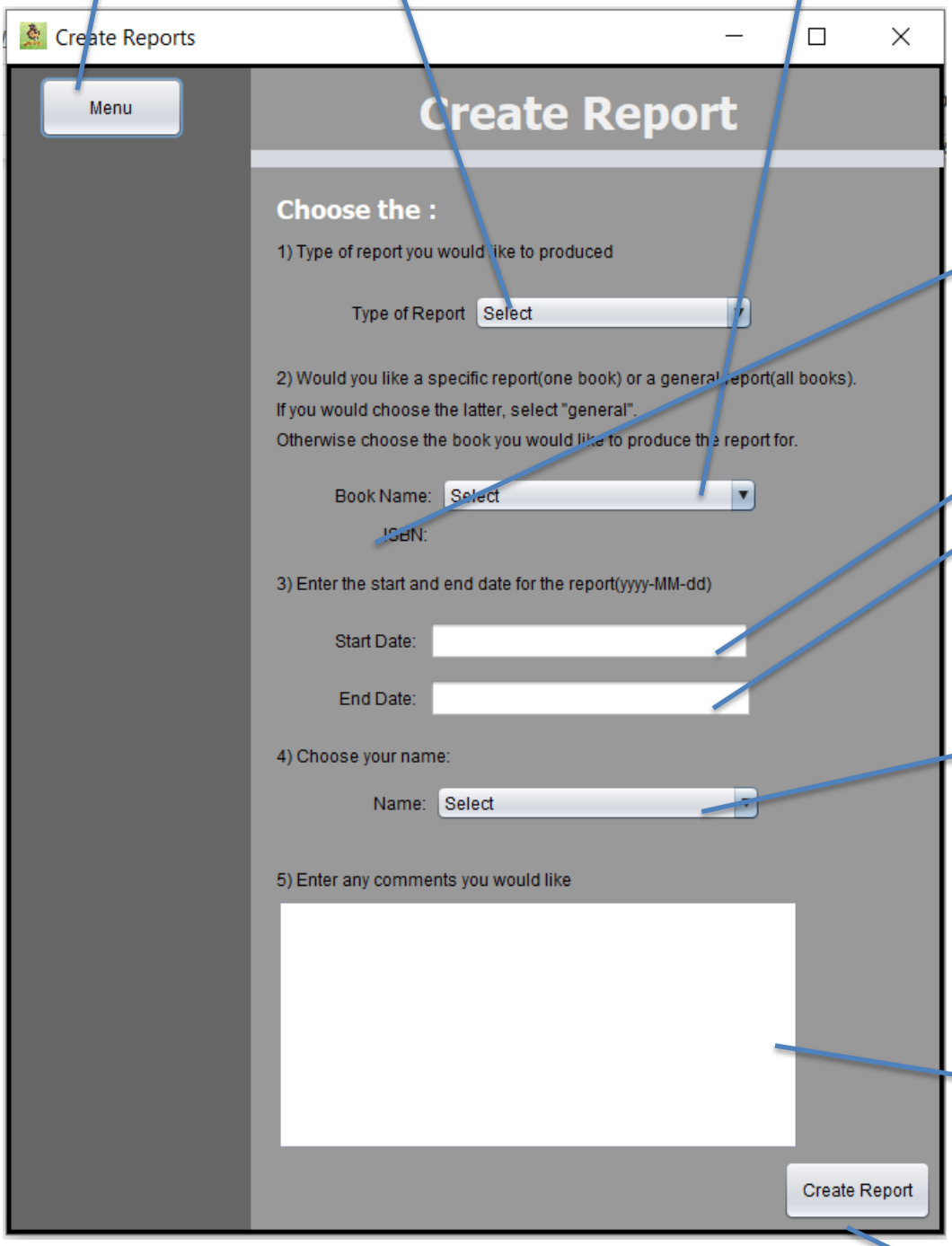


Create Report

Takes user back to the home/ menu page

Combo box to choose the type of report [stock- based or price-based]

Combo box to establish the scope of the report [General or any book can be chosen to make it specific]



ISBN captured and displayed based on the item selected in the combo box above

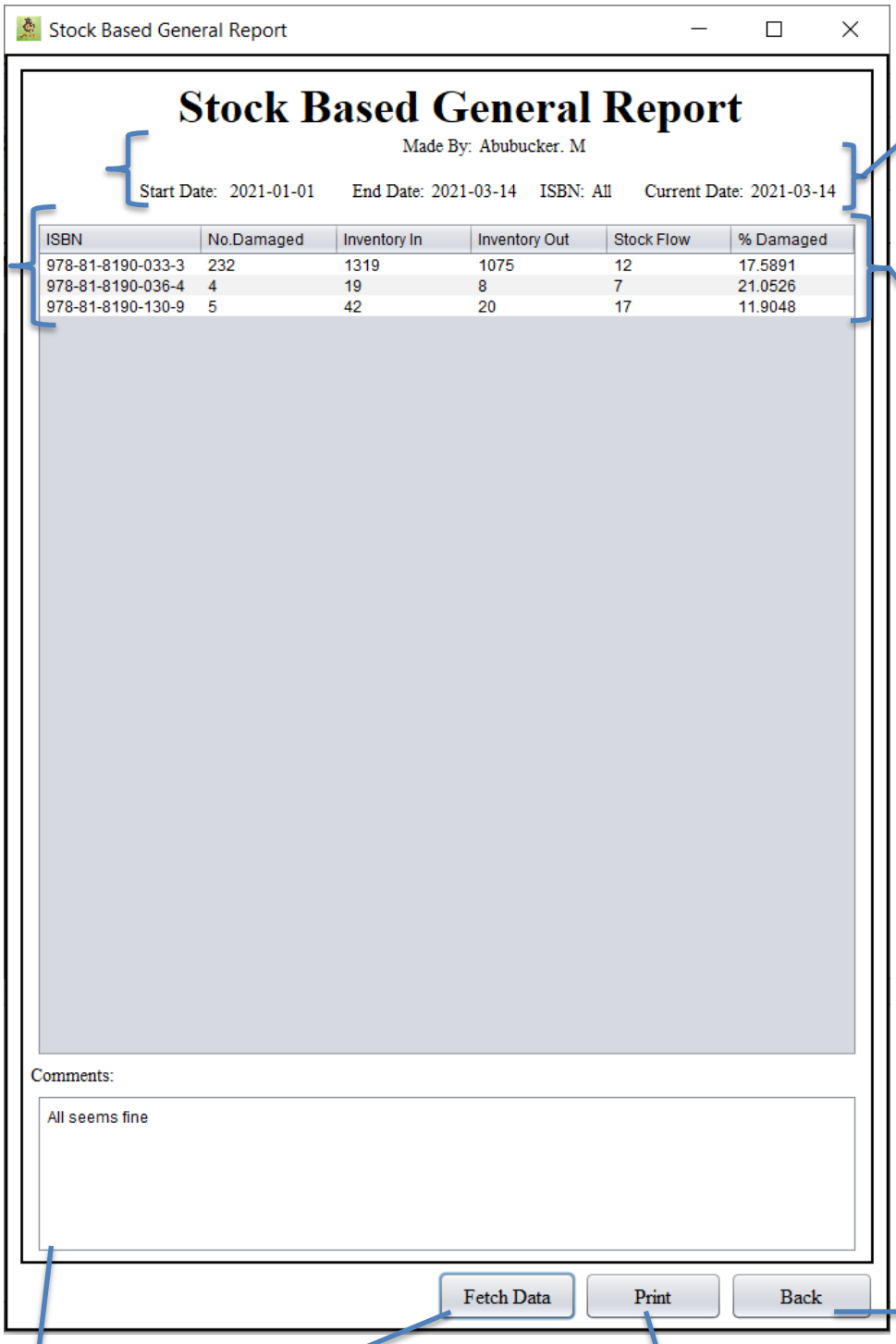
Text fields to enter the start and end date of the report, establish the time period of the report

Combo box to choose the author of the report [Contains the full name of each of the administrators which are seen under the administration information section]

Text area to type comments on the report to be made if necessary

Will take the user to the appropriate report layout and pass the start date, end date, and author name to the report provided the data entered follows test plan 6

Stock Based General Report



Author, Start Date, End Date, and ISBN are taken from the Create Reports page and Current date is automatically captured by the software

- ISBN is directly retrieved from the transactions.
- Number Damaged is calculated by finding the sum of all the number damaged values entered for incoming transactions for one title
- Inventory In is calculated by finding the sum of all the inventory in values entered for incoming transactions for one title
- Inventory Out is calculated by finding the sum of all the inventory out values entered for outgoing transactions for one title
- Stock flow is calculated by finding the difference between Inventory In and Inventory Out
- % Damaged is calculated by finding the ratio of Number Damaged to Inventory In and multiplying it by 100

Comments are taken from the create reports page and can be edited here as well

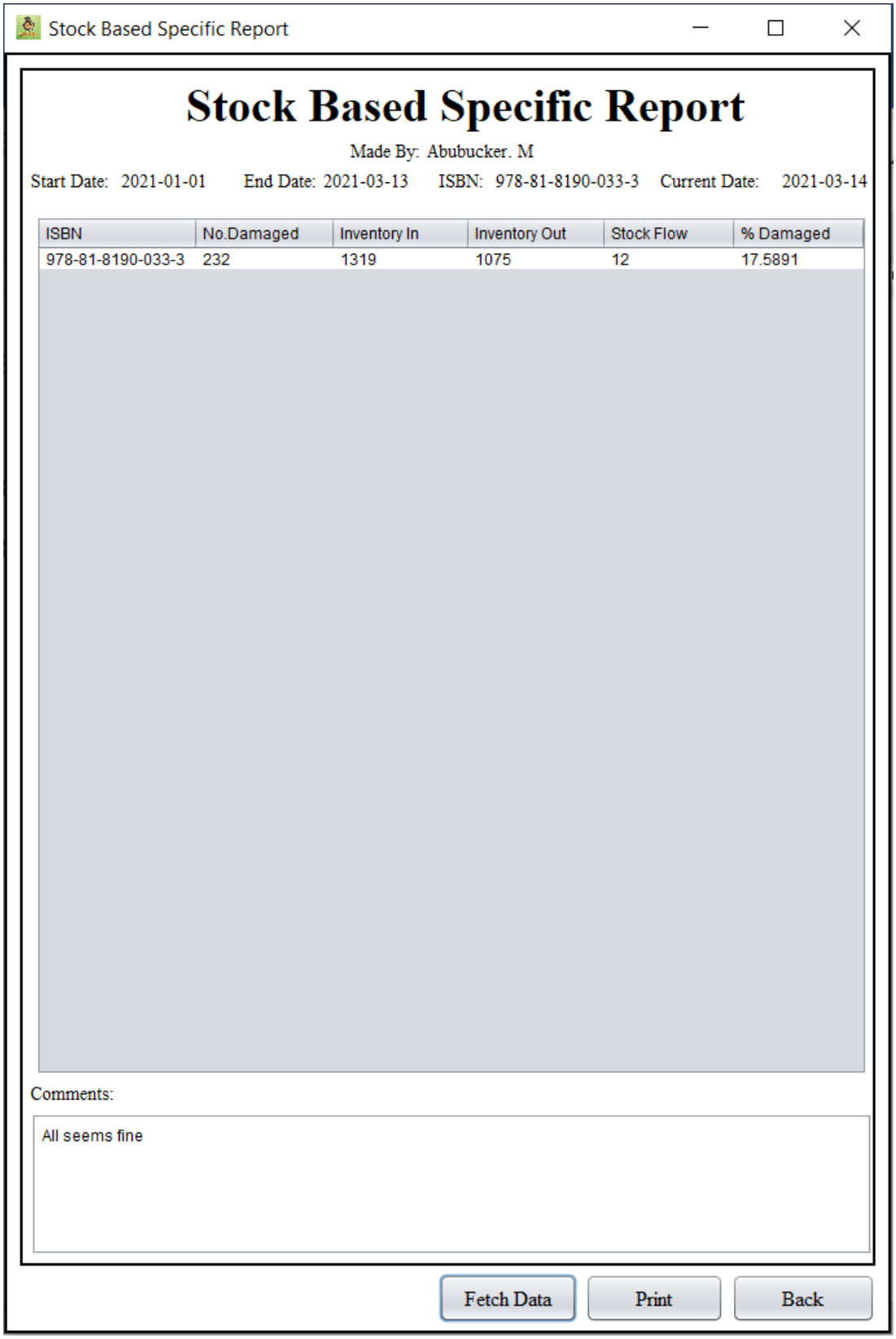
This button executes the MySQL query on multiple transactions and uses JOIN to order them in terms of the ISBN, captures the result set, and display it in the table above

This button prints or saves the panel with the data and the comments in an A4 size sheet or pdf

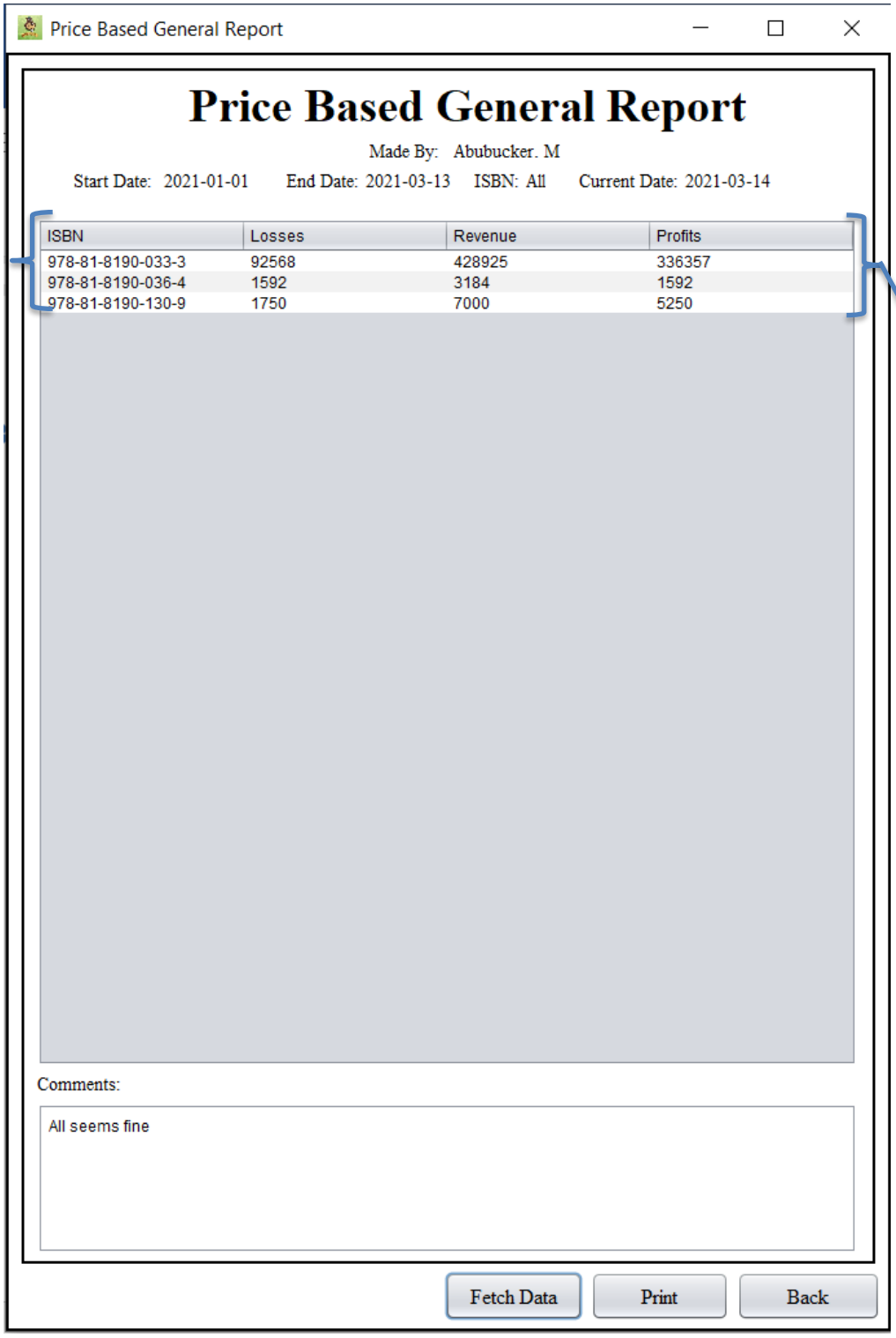
This button takes the user back to the create reports page to produce further reports

Stock Based Specific Report

- Same functionality as stock based general report except that it fetches the data only for one particular title



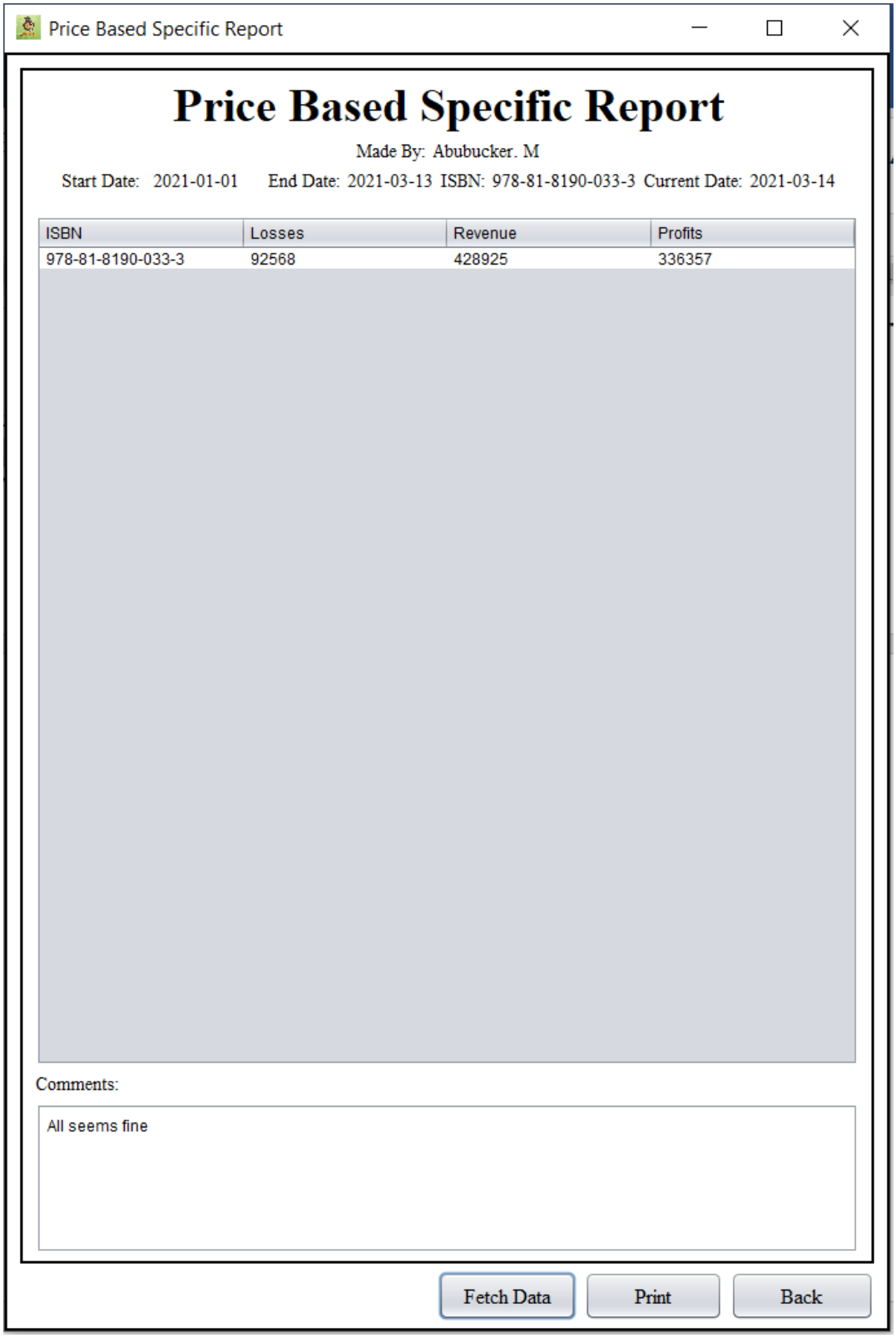
Price Based General Report



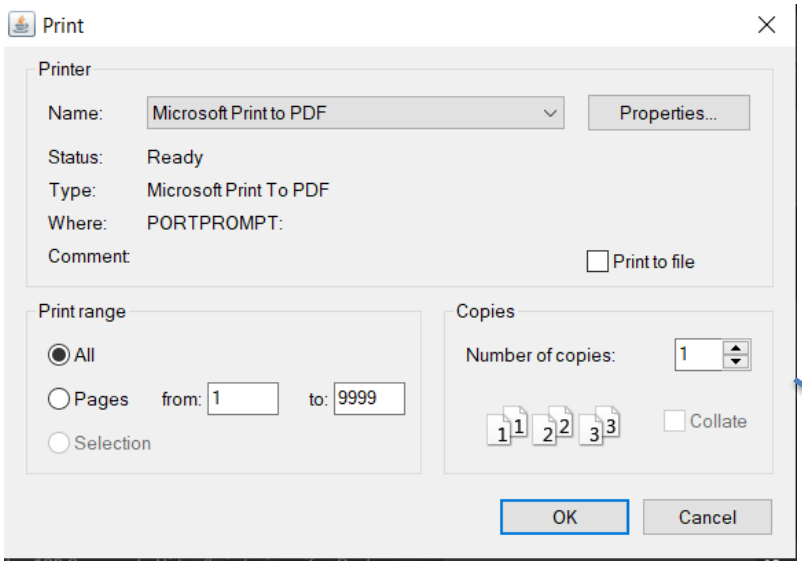
- ISBN is directly retrieved from the transactions.
- Losses are calculated by finding the sum of all the number damaged values entered for incoming transactions for one title and multiplying it by its Price (PriceRs from book_identification table)
- Revenue is calculated by finding the sum of all the inventory out values entered for outgoing transactions for one title and multiplying it by its Price (PriceRs from book_identification table)
- Profits are calculated by finding the difference between Revenue and Losses

Price Based Specific Report

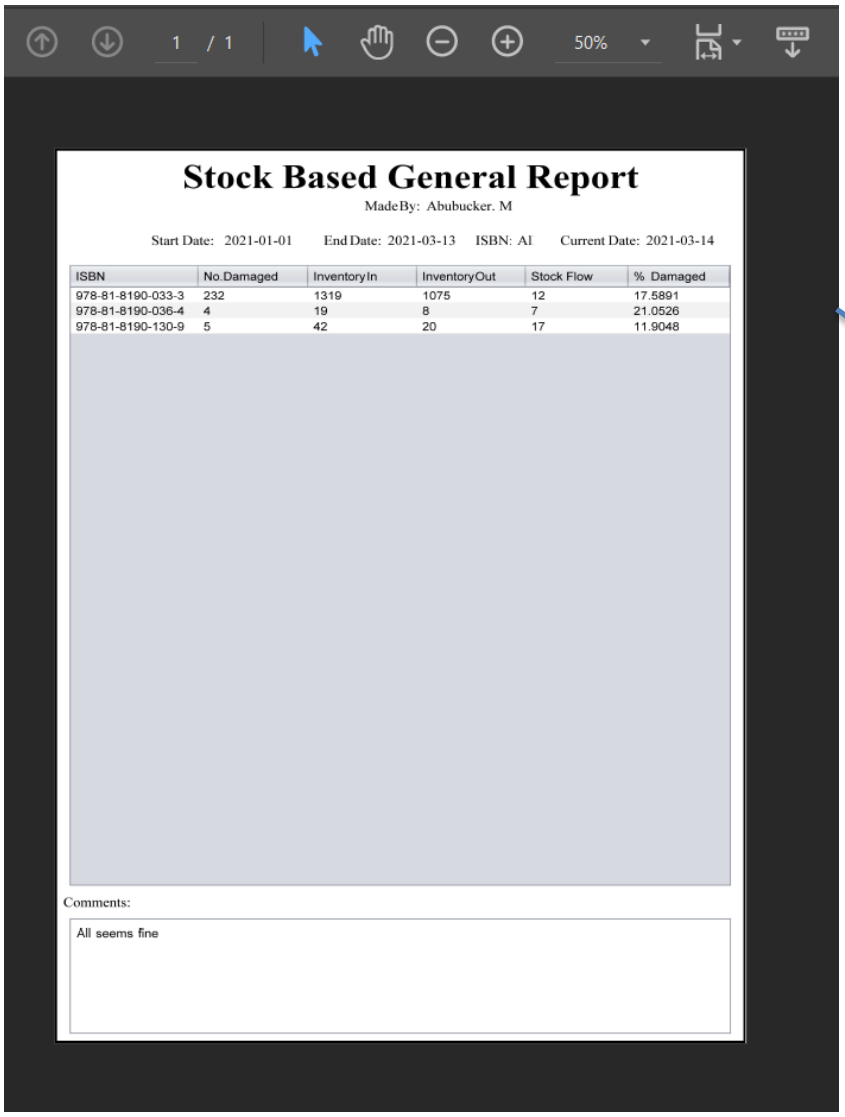
- Same functionality as price based general report except that it fetches the data only for one particular title



Printing Example (Stock based general report)



Print dialogue that is opened when "Print" button is clicked



Stock Based General Report opened in Adobe Acrobat scaled to the size of the A4 sheet in the form of a PDF