# **Object Oriented Programming**

# Project Report

# **Submitted To:**

Ma'am Amber Madeeha Zeb

# **Submitted By:**

Manaal Waseem FA18-BCE-074

# Contents

| Project Description                  | 3  |
|--------------------------------------|----|
| Vorking                              |    |
| Outputs                              |    |
| Concepts being used in this Project: |    |
|                                      |    |
| APPENDIX                             | 29 |

# **Project: Bookshop Management System**

# **Project Description:**

This is a Bookshop Management System which is made for carrying out the following tasks:

- Record keeping of books in shop
- Selling books from shop
- > Searching books in the shop
- Maintaining the cashbox

This project comprises the following classes and a main ( ) function:

- > Admin
- Animation
- Books
- Cashbox
- Controller

Following is the detail of functionality of each class respectively.

# \* Admin

```
1 #ifndef ADMIN H
2 #define ADMIN H
4 #include <string>
6 using namespace std;
8 class Admin
10 public:
11 ///Default constructor
12 Admin();
14 ///Parameterized constructor
15 Admin(string log, string pass);
17 ///Returns login value
18 string get login() const;
20 ///Returns password value
21 string get_pswd() const;
24 string login; ///variable to hold login value
25 string pswd; ///variable to hold password value
27
28 #endif
```

# **\*** Animation

```
1 #ifndef ANIMATION H
2 #define ANIMATION H
4 #include <string>
5 using namespace std;
7 class Animation
8 {
9 public:
10 ///Display phrase with borders
11 static void frame phrase(string, int, int count = 4);
12
13 ///Draw horizontal borders
14 static void draw border (int, int, int count = 4);
15
16 ///Exit from the system
17 static int exit();
18
19 ///System initialization
20 static void initializing();
21
22 ///Colorizes text
23 static void colorize text(string, int);
25 ///Display two points of border
26 static void points();
27 };
28
29 #endif
```

# **❖** Books

```
1 #ifndef BOOKS H
2 #define BOOKS H
4 #include <string>
5 #include <iostream>
6 #include <fstream>
7 #include <sstream>
8 #include <conio.h>
9 #include <iomanip>
10 #include <time.h>
11 #include <Windows.h>
12
13 using namespace std;
14
15 class Book
16 {
17 protected:
18 ///variables for all books types
19 string title;
20 float price;
21 int id;
22 string release date;
23 string genre;
24 string language;
25 string edition;
26 string writer;
27 float rating;
28 public:
29 ///functions set/get for all books types
30 virtual void add book()=0;
31 string get title();
```

```
32 void set_title();
33 void set_id(int);
34 int get_id();
35 int get id no update();
36 void set price();
37 float get_price();
38 void set_genre();
39 string get genre();
40 void set release date();
41 void set language();
42 void set edition();
43 void set writer();
44 void set_rating();
45 float get_rating();
46 string get_release_date();
47 string get_language();
48 string get_edition();
49 string get writer();
50 virtual ~Book()
51 {
52
53 }
54 };
55
56
57 class Paper book : public Book
58 ///Paper book is publicly derived class of Book class
59 {
60 protected:
61 string publisher;
62 int pages;
63 public:
64
65 void set_publisher();
66 void set_pages();
67 string get publisher();
68 virtual void add book();
69 int get pages();
70 ~Paper book()
71 {
72
73 }
74 };
76 class Audio cd : public Book
77 ///Audio cd book is publicly derived class of Book class
78 {
79 protected:
80 string listening_length;
81 string voice;
82 public:
83 virtual void add book();
84 void set voice();
85 void set listening length();
86 string get listening length();
87 string get_voice();
88 ~Audio cd()
89 {
90
91 }
92 };
93
94 class Digital book : public Book
95 ///Digital book is publicly derived class of Book class
96 {
97 public:
98 virtual void add_book();
99 ~Digital book()
```

```
100 {
101
102 }
103 };
104
105 #endif
```

# Cashbox

```
1 #ifndef CASHBOX H
2 #define CASHBOX H
4 #include <fstream>
5 #include <string>
6 #include <iostream>
8 using namespace std;
10 class Cashbox
11 {
12 private:
13 float cash; ///variable for cash
14 public:
16 ///Default constructor
17 Cashbox();
18
19 ///Parameterized constructor
20 Cashbox (float money);
22 ///Returns the value of cash in cashbox
23 float get cash()
25 return cash;
26
27
28 ///operator overloading in order to show the amount of money in cashbox
from object itself
29 friend void operator << (ostream &out, Cashbox cash)
31 ifstream money;
32
33 money.open("cashbox.dat", ios::in | ios::binary);
34 money.read((char*)&cash, sizeof(cash));
35 cout << cash.get_cash();
36 money.close();
37 }
38
39 //operator overloading in oder to add the price of book into cashbox.dat
40 friend void operator>>(istream &in, float price of book)
41 {
42 Cashbox cash;
43 fstream money;
44 money.open("cashbox.dat", ios::in | ios::out | ios::binary);
45 money.read((char*)&cash, sizeof(cash));
46 float update = cash.get cash() + price of book;
47 Cashbox new cash (update);
48 money.write ((char*) & new cash, sizeof (new cash));
49 money.close();
50 cout << new_cash.get_cash();</pre>
53 ///Reads and displays the value of cash from cashbox.dat
54 static void check cashbox();
55
```

```
56 ///Takes the price of book as argument and increases the cash amount -
inside the file cashbox.dat
57 static void update_cashbox(float);
58 };
59
60 #endif
```

# \* Controller

```
1 #ifndef CONTROLLER H
2 #define CONTROLLER H
4 #include "string"
6 using namespace std;
8 class Controller
10 protected:
11
12 public:
13 ///Display main menu of project
14 void main menu();
16 ///Display menu to add book
17 void add book menu();
18
19 ///Function to log out
20 void log_out_admin();
22 ///Display control panel
23 void control panel();
25 ///Display log-in screen and get login and password from user
26 void log in();
28 ///Display information about project developer
29 void about();
31 ///Sell book by id and delete it from record
32 void sell book();
33
34 ///Find book by id
35 void find book();
36
37 ///Display cash box
38 void display_cashbox();
39
40 private:
41 ///Display main menu screen
42 void main menu screen();
44 ///Display login screen
45 void login screen();
46
47 ///Take login from user
48 void login user (string &lgn);
50 ///Take password from user
51 void password user(string &psw);
53 ///Display wrong login screen
54 void wrong login screen();
55
56 ///Display control panel screen
57 void control panel screen();
```

```
59 ///Display add book menu screen
60 void add book menu screen();
62 ///Display paper book
63 void display paper book(ifstream& f1, float& price);
64 void display_paper_book(ifstream& f1);
66 ///Display Audio cd book
67 void display Audio cd book (ifstream& f2, float& price);
68 void display Audio cd book (ifstream & f2);
69
70 ///Display Digital book
71 void display_digital_book(ifstream& f3, float& price);
72 void display_digital_book(ifstream& f3);
73
74 ///Display book not found screen
75 void display not found screen();
76 };
77
78
79 #endif
```

# \* main ()

```
1 #include "Admin.h"
2 #include "Controller.h"
3 #include "fstream"
5 using namespace std;
7 int main()
8
9 Admin admin("MANAAL", "OOP"); ///Instance of Admin created and initialized
10
11 /// admin login and password written in file
12 ofstream Administration;
13 Administration.open("Admin.txt", ios::out);
14 Administration << admin.get login() << endl;
15 Administration << admin.get pswd();
16 Administration.close();
17
18 Controller bookShop; ///Instance of Controller created
19 bookShop.main menu();
20 return 0;
21 }
```

# Working:

In this program, the user is firstly presented with three options that are:

- > Login
- > About
- > Exit

# • Login:

The Login option allows user to enter the Bookshop Management System as admin. After Login, user is presented with Control Panel which allows user to perform following five functionalities:

- ➤ Add New Book
- > Sell Book
- Search Book
- Check Cashbox
- ➤ Log out

#### Add New Book:

This option allows user to add three types of new books i.e. paper book, audio book and digital book.

#### Sell Book:

This option allows user to sell a book identified via. Book Id. This functionality deletes the record of that particular book from the database (the name of the folder where files are stored) and add its price in the cash box.

#### Search Book:

This option allows user to search a book through its Id.

#### Check Cashbox:

This option allows user to check the total amount of cash in cashbox.

#### Log out

This option allows user to logout from control panel. Once the user is logout main menu appears.

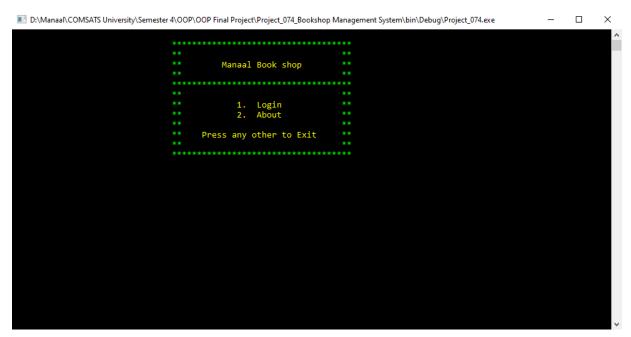
#### • About:

About option presents user with the information of the developer of this Bookshop Management System.

#### • Exit:

Exit option allows user to exit the Bookshop Management System.

# **Outputs**:



#### Main Menu

```
D:\Manaa\\COMSATS University\Semester 4\00P\00P Final Project_074_Bookshop Management System\bin\Debug\Project_074.exe  

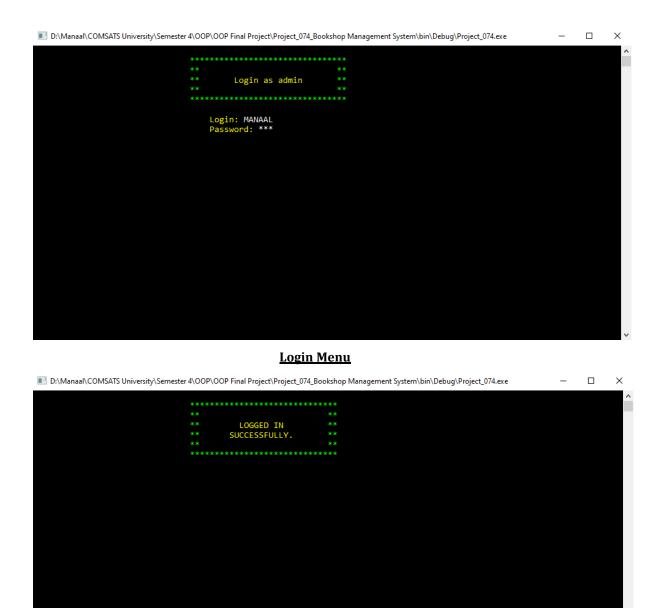
Final Project

Course Title: Object Oriented Programming
Teacher: Amber Madeeha Zeb

Made by:

Manaal Waseem - FA18-BCE-074
```

**About** 



**Login Success Window** 

#### **Control Panel**

```
D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe — X

TYPE OF BOOK

1. Paper book
2. Audio book
3. Digital book
4. Any other key to GO BACK
4. Any o
```

**Add New Book** 

#### **Add New Paper Book**

**Add New Audio Book** 

```
System needs ID of book to sell

Sell by ID: 22
```

Sell Book



**Book Sold** 

```
**D:\Manaal\COMSATS University\Semester 4\00P\00P Final Project\Project_074.Bookshop Management System\bin\Debug\Project_074.exe" - X

System needs ID of book to sell

Sell by ID: 8
```

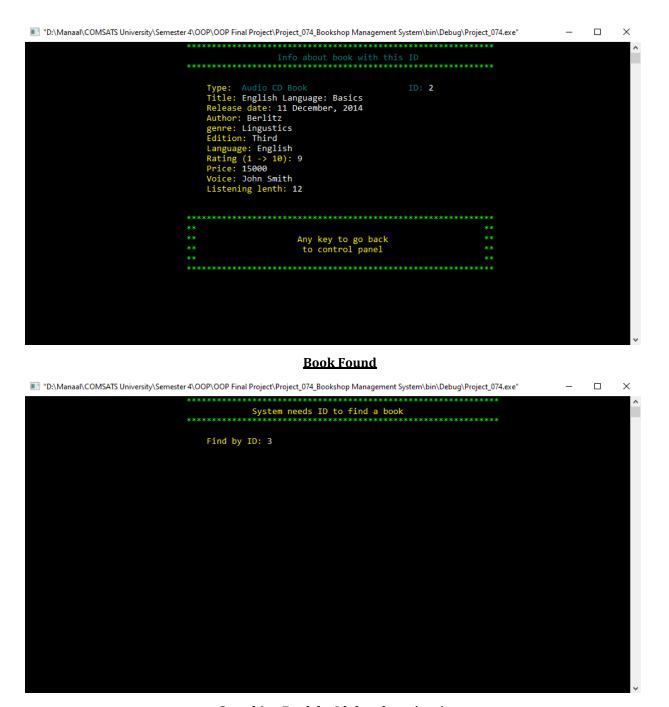
Sell Book

CASE: Book at this Id doesn't exist

**Book not Found** 



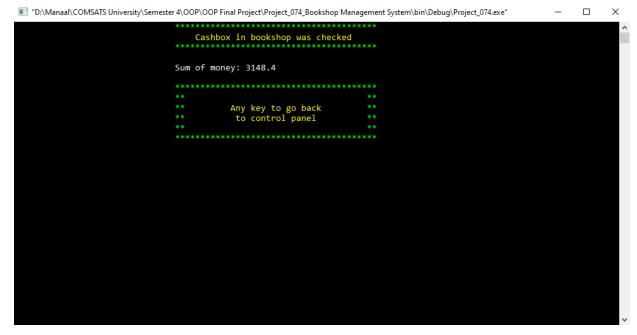
Search Book



Searching Book by Id that doesn't exist



#### **Book not Found**



**Check Cashbox** 

```
■ "D\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe" — 

Exiting from Bookshop Management System

Process returned 1 (0x1) execution time: 214.188 s

Press any key to continue.
```

**Exiting from Database** 

# **Concepts being used in this Project:**

# **Default Constructor:**

#### In **Admin.cpp**

```
///Default constructor
Admin::Admin()
{
    login = "Default login";
    pswd = "Default password";
};
```

# **Parameterized Constructor:**

#### In **Admin.cpp**

```
///Parameterized constructor
Admin::Admin(string log, string pass)
{
    login = log;
    pswd = pass;
}
```

# **Constructor with default parameters:**

#### In **Animation.h**

```
class Animation
{
public:
    ///Display phrase with borders
    static void frame_phrase(string, int, int count = 4);

    ///Draw horizontal borders
    static void draw_border(int, int, int count = 4);
```

## **Data Hiding:**

#### In **Cashbox.h**

```
class Cashbox
{
private:
    float cash; ///variable for cash
```

# **Constant function:**

#### In **Admin.cpp**

```
///Returns login value
string Admin::get_login() const
{
    return login;
}

///Returns password value
string Admin::get_pswd() const
{
    return pswd;
}
```

## **Inline Function:**

#### In Cashbox.h

#### **Static function:**

#### In **Animation.h**

```
class Animation
{
  public:
    ///Display phrase with borders
    static void frame_phrase(string, int, int count = 4);

  ///Draw horizontal borders
  static void draw_border(int, int, int count = 4);

  ///Exit from the system
  static int exit();

  ///System initialization
  static void initializing();

  ///Colorizes text
  static void colorize_text(string, int);

  ///Display two points of border
  static void points();
};
```

#### Pass by reference:

#### In Controller.h

```
///Take login from user
void login_user(string &lgn);

///Take password from user
void password_user(string &psw);
```

## File Handling:

#### In main.cpp

Write in text file.

```
/// admin login and password written in file
ofstream Administration;
Administration.open("Admin.txt", ios::out);
Administration << admin.get_login() << endl;
Administration << admin.get_pswd();
Administration.close();</pre>
```

#### In Controller.cpp

Read from text file.

```
///Reading login and password from file
string login, passwd;
ifstream Administration;
Administration.open("Admin.txt", ios::in);
Administration >> login;
Administration >> passwd;
Administration.close();
Admin admin(login, passwd);
```

#### In Cashbox.cpp

Read from and write in a binary file.

```
///Takes the price of book as argument and increases the cash amount - inside the file cashbox.dat
void Cashbox::update_cashbox(float price_of_book)
{
    Cashbox cash;
    ifstream ifile;
    ofstream ofile;

    ifile.open("cashbox.dat", ios::binary);
    ifile.read((char*)&cash, sizeof(cash));
    float update = cash.get_cash() + price_of_book;
    ifile.close();

    ofile.open("cashbox.dat", ios::out | ios::binary);
    Cashbox new_cash(update);
    ofile.write((char*)&new_cash, sizeof(new_cash));
    ofile.close();
}
```

# **Exception Handling:**

#### In **Controller.cpp**

```
try
{
    if (id < 0)
        throw (id);
} //end of try block
catch (int thrownValue) // start of catch block
{
    cout << endl;
    Animation::colorize_text("Exception caught. Id cannot be negative", 4);
    cout << endl;
    Sleep(2000);
    find_book();
} // end of catch block</pre>
```

# **Polymorphism:**

#### In **Controller.cpp**

```
void Controller::add_book_menu()
   Book* b;
   add book menu screen();
   while (true)
       char ch = getch();
       switch (ch)
           case '1':
               b = new Paper_book;
               b->add book();
               delete b;
               break;
           case '2':
               b = new Audio_cd;
               b->add_book();
               delete b;
               break:
           case '3':
               b = new Digital_book;
               b->add_book();
               delete b;
               break;
```

## **Operator overloading and friend function:**

#### In Cashbox.h

```
//operator overloading in order to show the amount of money in cashbox from object itself
friend void operator << (ostream &out, Cashbox cash)
   ifstream money;
   money.open("cashbox.dat", ios::in | ios::binary);
   money.read((char*)&cash, sizeof(cash));
   cout << cash.get_cash();</pre>
   money.close();
//operator overloading in oder to add the price of book into cashbox dat
friend void operator>>(istream &in, float price of book)
   Cashbox cash;
   fstream money;
   money.open("cashbox.dat", ios::in | ios::out | ios::binary);
   money.read((char*)&cash, sizeof(cash));
   float update = cash.get_cash() + price_of_book;
   Cashbox new_cash(update);
   money.write((char*)&new_cash, sizeof(new_cash));
   money.close();
   cout << new_cash.get_cash();</pre>
```

#### Inheritance:

#### In Books.h

```
class Book

class Paper_book: public Book

//Paper_book is publicly derived class of Book class

class Audio_cd: public Book

//Audio_cd_book is publicly derived class of Book class

class Digital_book: public Book

//Digital_book is publicly derived class of Book class

class Digital_book : public Book

//Digital_book is publicly derived class of Book class

class Digital_book is publicly derived class of Book class
```

# **Abstract Base Class**

```
class Book
protected:
   //variables for all books types
  string title;
  float price;
  int id;
  string release_date;
  string genre;
   string language;
   string edition;
   string writer;
   float rating;
public:
   //functions set/get for all books types
   virtual void add book()=0;
   string get title();
   void set title();
   void set id(int);
   int get id();
```

# **Virtual Destructor:**

#### In **Books.h**

```
string get_release_date();
string get_language();
string get_edition();
string get_writer();
virtual ~Book()
{
};
```

# **Function Overloading:**

## In **Controller.h**

```
///Display paper_book
void display_paper_book(ifstream& f1, float& price);
void display_paper_book(ifstream& f1);

///Display Audio_cd_book
void display_Audio_cd_book(ifstream& f2, float& price);
void display_Audio_cd_book(ifstream& f2);

///Display Digital_book
void display_digital_book(ifstream& f3, float& price);
void display_digital_book(ifstream& f3);
```

#### APPENDIX:

# Admin

```
#include "Admin.h"
 3
   ///Default constructor
   Admin::Admin()
 4
 5
 6
        login = "Default login";
 7
       pswd = "Default password";
 8
10 ///Parameterized constructor
11 Admin::Admin(string log, string pass)
12
       login = log;
13
14
       pswd = pass;
15 }
16
17 ///Returns login value
18 string Admin::get login() const
19
20
       return login;
21 }
22
23 ///Returns password value
24 string Admin::get_pswd() const
25 {
       return pswd;
26
27
```

# Animation

```
#include "Animation.h"
2 #include <Windows.h>
3
   #include <iostream>
   #include <string>
6 using namespace std;
8 ///Display two points of border
9 void Animation::points()
10 {
       colorize text("*", 10); ///prints * in Light Green color
11
       colorize_text("*", 10);
12
13 }
14
15 ///Exit from the system
16 int Animation::exit()
17 {
18
           system("cls");
19
           string s;
20
           s = "Exiting from Bookshop Management System";
21
           cout << endl;
           colorize text(s, 10);
22
23
           cout << endl;
           cout << endl << endl;</pre>
24
25
           return 0;
26 }
2.7
28 ///System initialization
29
  void Animation::initializing()
30 {
31
       int color = 10;
32
       cout << endl;</pre>
       draw border(30, color);
33
                                                ", color);
34
       frame phrase ("
       frame_phrase("
                                                ", color);
35
                             LOGGED IN
                                                ", color);
        frame_phrase("
36
                           SUCCESSFULLY.
        frame_phrase("
                                                ", color);
37
        draw border (30, color);
38
```

```
Sleep(1000);
                          ///wait for 1000 milliseconds
        system("cls");
40
41
42
43
44
   ///Display phrase with borders
45 void Animation::frame phrase(string phrase, int color, int count)
46
47
        string ch = "*";
48
        for (int i = 0; i < count; i++)</pre>
49
50
            cout << "\t";</pre>
51
52
       colorize_text(ch, color);
        colorize_text(ch, color);
colorize_text(phrase, 14);
53
54
55
        colorize_text(ch, color);
        colorize_text(ch, color);
56
57
        cout << endl;
58 }
59
60
   ///Draw horizontal borders
   void Animation::draw_border(int amount, int color, int count)
61
62
        string g = "*";
63
64
        for (int i = 0; i < count; i++)</pre>
65
            cout << "\t";</pre>
66
67
        for (int i = 0; i < amount; i++)</pre>
68
69
70
            colorize_text(g, color);
71
72
        cout << endl;</pre>
73 }
74
   ///Colorizes text
7.5
  void Animation::colorize text(string text, int color)
77
78
        int color white = 15;
79
        HANDLE hConsole = GetStdHandle(STD OUTPUT HANDLE); //Get a handle to the standard
output
80
        SetConsoleTextAttribute(hConsole, color);
        cout << text;</pre>
81
82
        SetConsoleTextAttribute(hConsole, color white);
83
84
85
```

# Books

```
#include "Books.h'
    #include "Animation.h"
 3 #include "Controller.h"
 4
 6 using namespace std;
 8
        string Book::get_title()
 9
10
            return title;
11
12
        void Book::set_id(int i)
13
14
            id = i;
15
16
17
18
        float Book::get price()
19
20
            return price;
21
2.2
23
        ///Returns id with incrementing
```

```
24
         int Book::get id()
25
 26
             string file = "Update ID.txt";
27
28
             ifstream last id;
29
             last_id.open(file.c_str());
 30
             last id >> id;
             last_id.close();
 31
 32
 33
             ofstream update id;
             update_id.open(file.c_str());
 34
 35
             update_id << (id + 1);
             update id.close();
36
 37
 38
             return id;
39
 40
 41
         ///Returns id whithout incrementing
 42
         int Book::get id no update()
43
 44
             return id;
 45
 46
 47
         float Book::get rating()
48
 49
             return rating;
50
 51
 52
         string Book::get release date()
 53
 54
             return release date;
 55
 56
57
         string Book::get_language()
 58
 59
             return language;
 60
 61
 62
         string Book::get edition()
 63
             return edition;
 64
 65
 66
         string Book::get genre()
 67
 68
 69
             return genre;
 70
 71
 72
         string Book::get writer()
 73
 74
             return writer;
 75
 76
 77
         void Book::set title()
 78
 79
             string t;
80
             getline(cin, t);
81
             title = t;
82
83
 84
         void Book::set price()
8.5
 86
             float t = -1;
             while (t < 0 || !(cin.good()) || t > 10000000)
87
88
89
                      cin.clear();
                      cin.ignore(100, '\n');
90
                      Animation::colorize_text("\t\t\t\ Price: ", 14);
 91
 92
                      cin >> t;
 93
                      price = t;
 94
 95
96
97
         void Book::set_release_date()
 98
99
             string t;
100
             getline(cin, t);
```

```
101
             release date = t;
102
103
104
         void Book::set language()
105
106
             string t;
107
             Animation::colorize text("\t\t\t\ Language: ", 14);
108
             cin >> t;
109
             language = t;
110
111
112
         void Book::set edition()
113
114
             string t;
115
             getline(cin, t);
             edition = t;
116
117
118
119
         void Book::set writer()
120
121
             string t;
122
             getline(cin, t);
123
             writer = t;
124
125
126
         void Book::set rating()
127
128
             float t = - 1;
129
             while (t < 1 || t > 10 || !(cin.good()))
130
131
                 cin.clear();
                 cin.ignore(100, '\n');
132
133
                 Animation::colorize text("\t\t\t\ Rating (1 \rightarrow 10): ", 14);
134
                 cin >> t;
135
                 rating = t;
136
137
138
139
140
         void Book::set genre()
141
142
             string t;
143
            getline(cin, t);
144
             genre = t;
145
146
147
         ///Paper book functions
148
149
         string Paper book::get publisher()
150
151
             return publisher;
152
153
154
         int Paper book::get pages()
155
156
             return pages;
157
158
159
         void Paper book::set publisher()
160
             Animation::colorize_text("\t\t\t\ Publisher: ", 14);
161
162
             cin >> publisher;
163
164
165
         void Paper_book::set_pages()
166
167
             int t = -1;
             while (t < 0 || !(cin.good()))</pre>
168
169
170
                 cin.clear();
                 cin.ignore(100, '\n');
171
172
                 Animation::colorize text("\t\t\t\ Volume in pages: ", 14);
173
                 cin >> t;
174
                 pages = t;
175
176
177
```

```
///Audio cd functions
179
         string Audio cd::get listening length()
180
181
             return listening length;
182
183
184
         string Audio cd::get voice()
185
186
             return voice;
187
188
189
         void Audio cd::set voice()
190
191
             cin.ignore(100, '\n');
192
             getline(cin, voice);
193
194
195
         void Audio cd::set listening length()
196
197
             getline (cin, listening length);
198
199
         ///Implementation of pure virtual function by derived classes
2.00
201
         void Paper book::add book()
202
203
             system("cls");
204
             Paper book pb;
205
             int color yellow = 14;
206
             int color = 10;
             cout << endl;
207
208
209
             Animation::draw border(61, color);
210
             Animation::colorize text("\t\t\t\t
                                                                  Fill the form to add book
n", 07);
211
            Animation::draw border(61, color);
212
             cout << endl;</pre>
             Animation::colorize_text("\n\t\t\t\ Title: ", color yellow);
213
214
            pb.set title();
            Animation::colorize text("\t\t\t
                                                  Release date: ", color yellow);
215
216
            pb.set release date();
217
            Animation::colorize text("\t\t\t
                                                   Author: ", color yellow);
218
            pb.set writer();
219
            Animation::colorize_text("\t\t\t\t
                                                   genre: ", color yellow);
             pb.set_genre();
220
221
            Animation::colorize text("\t\t\t\ Edition: ", color yellow);
222
            pb.set_edition();
223
            pb.set language();
            pb.set rating();
224
225
            pb.set_price();
226
             pb.set_publisher();
227
            pb.set_pages();
            Animation::colorize_text("\t\t\t ID of this book in the system: ", 3);
228
229
             cout << pb.get id();</pre>
230
            cout << endl;</pre>
231
            cout << endl;</pre>
2.32
             ofstream pb write;
233
             pb write.open("DataBase/Paper Books/book - " + to string(pb.get id no update())
+ ".info");
234
             pb_write << pb.get_id_no_update() << endl;</pre>
235
             pb_write << pb.get_title() << endl;</pre>
             pb write << pb.get_release_date() << endl;</pre>
236
237
             pb write << pb.get writer() << endl;
             pb_write << pb.get_genre() << endl;</pre>
238
239
             pb_write << pb.get_edition() << endl;</pre>
240
             pb write << pb.get language() << endl;</pre>
241
            pb write << pb.get rating() << endl;</pre>
242
             pb_write << pb.get_price() << endl;</pre>
243
             pb write << pb.get publisher() << endl;</pre>
             pb write << pb.get_pages() << endl;</pre>
244
245
             pb_write.close();
246
             Animation::draw border(61, color);
247
248
             Animation::frame phrase("
", color);
249
             Animation::frame phrase(" New paper book has been successfully added to database
", color);
```

```
Animation::frame phrase("
", color);
251
             Animation::frame phrase("
                                                    -> Press any key to go back <-
", color);
252
             Animation::frame phrase("
", color);
253
             Animation::draw border(61, color);
254
             cout << endl;</pre>
255
256
            getch();
             system("cls");
2.57
258
             Controller c;
259
             c.control panel();
260
        };
261
262
         void Audio cd::add book()
263
2.64
             system("cls");
265
             Audio cd pb;
266
            fstream books;
267
            books.open("Audio.dat", ios::binary | ios::app);
268
             int color yellow = 14;
269
             int color = 10;
            cout << endl;</pre>
2.70
271
            Animation::draw border(61, color);
272
             Animation::colorize text("\t\t\t\t
                                                                   Fill the form to add
book
             \n", 07);
273
            Animation::draw border(61, color);
274
             cout << endl;</pre>
2.75
             Animation::colorize text("\t\t\t Title: ", color yellow);
276
            pb.set title();
            Animation::colorize_text("\t\t\t Release date: ", color_yellow);
277
            pb.set_release_date();
Animation::colorize_text("\t\t\t Author: ", color_yellow);
278
279
280
            pb.set writer();
281
            Animation::colorize_text("\t\t\t
                                                   genre: ", color_yellow);
             pb.set_genre();
282
283
            Animation::colorize text("\t\t\t\ Edition: ", color yellow);
284
            pb.set_edition();
            pb.set_language();
285
286
             pb.set rating();
287
             pb.set price();
             Animation::colorize_text("\t\t\t Voice: ", color_yellow);
288
             pb.set_voice();
289
290
             Animation::colorize text("\t\t\t\ Listening length (hours): ",
color_yellow);
            pb.set listening length();
292
           Animation::colorize text("\t\t\t ID of this book in the system: ", 3);
293
             cout << pb.get_id();</pre>
294
             cout << endl;</pre>
295
             cout << endl;</pre>
296
             Animation::draw border(61, color);
297
             Animation::frame phrase("
", color);
298
            Animation::frame phrase(" New audio book has been successfully added to
database ", color);
299
             Animation::frame phrase("
", color);
300
             Animation::frame phrase("
                                                    -> Press any key to go back <-
", color);
301
             Animation::frame phrase("
", color);
302
             Animation::draw border(61, color);
303
             cout << endl;</pre>
304
305
             ofstream pb write;
            pb write.open("DataBase/Audio CD/book - " + to string(pb.get id no update()))
306
+ ".info");
307
             pb write << pb.get id no update() << endl;</pre>
             pb_write << pb.get_title() << endl;</pre>
308
309
             pb_write << pb.get_release_date() << endl;</pre>
             pb_write << pb.get_writer() << endl;</pre>
310
311
             pb_write << pb.get_genre() << endl;</pre>
             pb_write << pb.get_edition() << endl;
pb_write << pb.get_language() << endl;</pre>
312
313
             pb write << pb.get rating() << endl;</pre>
314
```

```
315
             pb write << pb.get price() << endl;</pre>
316
             pb write << pb.get voice() << endl;</pre>
317
             pb write << pb.get listening length() << endl;</pre>
318
             pb write.close();
319
320
             books.write((char*)&pb, sizeof(pb));
321
             books.close();
322
323
             getch();
324
             system("cls");
325
             Controller c;
326
             c.control panel();
327
         };
328
329
         void Digital book::add book()
330
331
             system("cls");
332
             Digital book pb;
333
             fstream books;
             books.open("Digital.dat", ios::binary | ios::in | ios::app);
334
335
            books.open("Digital.dat", ios::binary | ios::app);
336
             int color yellow = 14;
337
             int color = 10;
338
             cout << endl;</pre>
339
             Animation::draw border(61, color);
340
             Animation::colorize text("\t\t\t\t
                                                                    Fill the form to add book
n", 07);
341
             Animation::draw border(61, color);
342
             cout << endl;</pre>
343
             Animation::colorize text("\t\t\t
                                                    Title: ", color yellow);
344
             pb.set title();
345
            Animation::colorize_text("\t\t\t
                                                   Release date: ", color yellow);
346
             pb.set release date();
             Animation::colorize_text("\t\t\t
                                                   Author: ", color_yellow);
347
348
            pb.set writer();
349
             Animation::colorize text("\t\t\t
                                                    genre: ", color yellow);
             pb.set_genre();
350
351
             Animation::colorize text("\t\t\t
                                                   Edition: ", color yellow);
352
            pb.set_edition();
353
             pb.set_language();
354
             pb.set rating();
355
             pb.set price();
             Animation::colorize_text("\t\t" ID of this book in the system: ", 3);
356
             cout << pb.get id();</pre>
357
358
             cout << endl;</pre>
             cout << endl;</pre>
359
360
             Animation::draw border(61, color);
             Animation::frame phrase("
", color);
362
             Animation::frame_phrase(" New digit-book has been successfully added to database
", color);
363
             Animation::frame phrase("
", color);
364
             Animation::frame_phrase("
                                                   -> Press any key to go back <-
", color);
365
             Animation::frame phrase("
", color);
366
             Animation::draw_border(61, color);
367
             cout << endl;</pre>
368
369
             ofstream pb write;
             pb write.open("DataBase/Digital Books/book - " +
370
to_string(pb.get_id_no_update()) + ".info");
             pb_write << pb.get_id_no_update() << endl;</pre>
371
372
             pb_write << pb.get_title() << endl;</pre>
373
             pb write << pb.get release date() << endl;</pre>
374
             pb_write << pb.get_writer() << endl;</pre>
375
             pb_write << pb.get_genre() << endl;</pre>
             pb write << pb.get edition() << endl;
376
377
             pb_write << pb.get_language() << endl;</pre>
378
             pb write << pb.get rating() << endl;</pre>
             pb_write << pb.get_price() << endl;</pre>
379
380
             pb write.close();
381
             books.write((char*)&pb, sizeof(pb));
382
383
             books.close();
```

# Cashbox

```
#include "Cashbox.h"
   using namespace std;
5
    ///Default constructor
6
   Cashbox::Cashbox()
8
        cash = 0;
9
10
11
   ///Parameterized constructor
12
   Cashbox::Cashbox(float money)
13
14
        cash = money;
15 };
16
17
18
   ///Reads and displays the value of cash from cashbox.dat
19
   void Cashbox::check cashbox()
20
   {
21
        Cashbox cash;
22
       ifstream money;
23
       money.open("cashbox.dat", ios::binary);
24
25
       money.read((char*)&cash, sizeof(cash));
26
        cout << cash.get_cash();</pre>
2.7
       money.close();
28 }
29
30 ///Takes the price of book as argument and increases the cash amount - inside the
file cashbox.dat
31 void Cashbox::update_cashbox(float price_of_book)
32
33
        Cashbox cash;
34
35
        ifstream ifile;
36
       ofstream ofile;
37
38
        ifile.open("cashbox.dat", ios::binary);
39
        ifile.read((char*)&cash, sizeof(cash));
40
        float update = cash.get cash() + price of book;
41
       ifile.close();
42
43
        ofile.open("cashbox.dat", ios::out | ios::binary);
44
        Cashbox new_cash(update);
45
        ofile.write((char*)&new cash, sizeof(new cash));
46
        ofile.close();
47
```

# Controller

```
1  #include "Controller.h"
2  #include "Cashbox.h"
3  #include "Books.h"
4  #include "Animation.h"
5  #include "Admin.h"
6
7  #include <iostream>
8  #include <fstream>
9  #include <string>
10  #include <Windows.h>
11  #include <conio.h>
```

```
13
    using namespace std;
14
15
   ///Display main menu of project
16
    void Controller::main menu()
17
18
        main menu screen();
19
        while (true)
20
21
            char ch = getch();
2.2
23
            switch (ch)
24
                case '1':
2.5
26
                    system("cls");
27
                    log in();
28
                    break;
29
                case '2':
30
                    system("cls");
31
                    about();
32
                    break;
33
                default:
34
                    system("cls");
                    Animation::exit();
36
                    exit(1);
37
38
39
   }
40
   ///Display log-in screen and get login and password from user
41
   void Controller::log in()
43
44
        string lqn, psw;
45
46
        login screen();
47
        Animation::colorize_text("\t\t\t Login: ", 14);
48
49
        login user(lgn);
50
51
        Animation::colorize text("\r\t\t\t Password: ", 14);
52
        password user (psw);
53
54
        ///Reading login and password from file
        string login, passwd;
55
        ifstream Administration;
56
        Administration.open("Admin.txt", ios::in);
58
       Administration >> login;
59
        Administration >> passwd;
60
        Administration.close();
61
        Admin admin (login, passwd);
62
        ///Checking if the login or password wrong
63
64
        if (lgn != admin.get login() || psw != admin.get pswd())
65
66
            system("cls");
67
            wrong_login_screen();
68
            Sleep (1000);
            system("cls");
69
70
            log_in();
71
72
73
       ///Go to next menu
74
        system("cls");
75
        Animation::initializing();
76
        control_panel();
77
78
79
80
   ///Display control panel
81
    void Controller::control_panel()
82
83
        control_panel_screen();
84
85
        while (true)
86
87
            char ch = getch();
```

```
switch (ch)
 89
                  case '1':
 90
 91
                      add book menu();
 92
                      break;
 93
                 case '2':
 94
                      sell book();
 95
                     break;
 96
                 case '3':
 97
                     find book();
 98
                     break;
99
                 case '4':
                     display_cashbox();
100
101
                     break;
102
                 case '0':
                     log_out_admin();
103
104
                     break;
105
106
107
108
109
110 void Controller::log_out_admin()
111
112
         system("cls");
113
114
         int color = 10;
115
116
         cout << endl;</pre>
117
         Animation::draw border(30, color);
118
         Animation::frame_phrase("
                                                               ", color);
                                                              ", color);
", color);
", color);
         Animation::frame_phrase("
119
                                         Are you sure?
120
         Animation::frame phrase("
                                       Y -> Yes | N <- No
         Animation::frame_phrase("
121
122
         Animation::draw border(30, color);
123
124
         while (true)
125
126
             char ch = getch();
127
             switch (ch)
128
129
                 case 'Y':
130
                 case 'y':
                    system("cls");
131
132
                     main_menu();
133
134
                 case 'N':
                 case 'n':
135
136
                   system("cls");
137
                      control_panel();
138
                     break;
139
140
141 }
142
143 //add book menu
144 void Controller::add book menu()
145 {
         Book* b;
146
147
148
         add_book_menu_screen();
149
         while (true)
150
             char ch = getch();
151
152
             switch (ch)
153
154
                  case '1':
                     b = new Paper book;
155
156
                     b->add book();
157
                      delete b;
158
                     break;
159
                 case '2':
160
                     b = new Audio cd;
161
                      b->add book();
162
                      delete b;
                     break;
```

```
164
                  case
                     b = new Digital_book;
165
166
                      b->add book();
167
                      delete b;
168
                      break;
169
                  default:
170
                      control panel();
171
172
173
174 }
175
176 ///Sell book by id and delete it from record
177      void Controller::sell book()
178 {
         system("cls");
179
180
         int color = 10;
181
         int color lightblue = 14;
182
         float price;
183
         int id = -1;
184
         Animation::draw border(62, color);
185
         Animation::colorize_text("\t\t\t\t
                                                         System needs ID of book to sell
186
\n", color lightblue);
187
         Animation::draw border(62, color);
188
         cout << endl;</pre>
189
190
         while (id < 0 )
191
192
             Animation::colorize text("\t\t\t Sell by ID: ", color lightblue);
193
             cin >> id;
194
195
         system("cls");
196
197
         string n1 = "DataBase/Paper Books/book - " + to string(id) + ".info";
         string n2 = "DataBase/Audio CD/book - " + to_string(id) + ".info";
string n3 = "DataBase/Digital Books/book - " + to_string(id) + ".info";
198
199
200
201
         ifstream f1;
202
         fl.open(n1, ios::in);
203
204
         if (f1.is open())
205
             display_paper_book(f1, price);
206
         ifstream f2;
2.07
208
         f2.open(n2, ios::in);
209
210
         if (f2.is open())
211
             display Audio cd book(f2, price);
212
213
         ifstream f3;
214
         f3.open(n3, ios::in);
215
         if (f3.is open())
216
             display digital book(f3, price);
217
218
         if (f3.is_open() | f1.is_open() | f2.is_open())
219
220
             Animation::draw border(61, color);
221
             Animation::frame_phrase("
", color);
222
             Animation::frame phrase("
                                          This action will delete book from database
", color);
223
             Animation::frame_phrase("
                                                         Are you sure about that?
", color);
224
             Animation::frame_phrase("
", color);
225
                                                           Y <- Yes | NO -> N
             Animation::frame phrase("
", color);
226
             Animation::frame phrase("
", color);
227
             Animation::draw border(61, color);
228
229
             while (true)
230
231
                  char ch = getch();
232
                  switch (ch)
```

```
233
                     case 'Y':
2.34
                     case 'y':
235
236
                         system("cls");
237
                         if (f1.is open())
238
239
                             f1.close();
240
                             remove(n1.c str());
2.41
242
                         if (f2.is open())
243
244
                              f2.close();
245
                             remove(n2.c str());
246
                         if (f3.is_open())
247
248
249
                             f3.close();
250
                             remove(n3.c_str());
251
2.52
253
                         Cashbox::update cashbox(price);
254
                         system("cls");
255
256
                        Animation::draw border(61, color);
257
                         Animation::frame phrase("
", color);
258
                                                       Book was deleted from
                        Animation::frame phrase("
                    ", color);
database
259
                        Animation::frame phrase("
", color);
260
                         Animation::draw border(61, color);
                         Sleep(1500); //wait for 1500 milliseconds
261
262
                         control panel();
2.63
                        break;
264
                    case 'N':
265
                     case 'n':
                        system("cls");
266
267
                         control panel();
268
                        break;
269
                     default:
270
                         system("cls");
271
                         control panel();
2.72
                 }//switch close
            }//while close
273
         }//if close
274
275
         else
276
            display_not_found_screen();
277 }//function close
278
279 ///Find book by id
280 void Controller::find book()
281 {
282
         system("cls");
283
        int color = 10;
284
         int color lightblue = 14;
285
        int id;
286
287
         Animation::draw border(62, color);
         Animation::colorize_text("\t\t\t\t
                                                      System needs ID to find a book
288
\n", color lightblue);
289
        Animation::draw border(62, color);
290
         cout << endl;</pre>
2.91
292
         Animation::colorize text("\t\t\t Find by ID: ", color lightblue);
293
         cin >> id;
294
295
         try
296
297
             if (id < 0)
298
                throw (id);
299
300
         catch (int thrownValue) // start of catch block
301
302
             cout << endl;</pre>
             Animation::colorize text("Exception caught. Id cannot be negative", 4);
303
304
             cout << endl;</pre>
```

```
305
              Sleep (2000);
306
             find book();
307
          } // end of catch block
308
309
         system("cls");
310
         string n1 = "DataBase/Paper Books/book - " + to_string(id) + ".info";
string n2 = "DataBase/Audio CD/book - " + to_string(id) + ".info";
311
312
         string n3 = "DataBase/Digital Books/book - " + to_string(id) + ".info";
313
314
315
         ifstream f1;
316
         fl.open(nl, ios::in);
317
318
         if (f1.is_open())
319
             display paper book(f1);
320
321
         ifstream f2;
322
         f2.open(n2, ios::in);
323
324
         if (f2.is open())
325
             display Audio cd book(f2);
326
327
         ifstream f3;
328
         f3.open(n3, ios::in);
329
         if (f3.is open())
330
             display digital book(f3);
331
332
         if (f3.is open() | f1.is open() | f2.is open())
333
334
              Animation::draw border(61, color);
335
             Animation::frame phrase("
", color);
336
             Animation::frame phrase("
                                                             Any key to go back
", color);
337
             Animation::frame phrase("
                                                              to control panel
", color);
338
             Animation::frame_phrase("
", color);
339
             Animation::draw border(61, color);
340
             f1.close();
341
             f2.close();
342
             f3.close();
343
344
             while (true)
345
346
                 getch();
                 system("cls");
347
348
                 control panel();
349
         }//if close
350
351
         else
352
             display not found screen();
353 }
354
355 ///Display information about project developer
356 void Controller::about()
357
358
         int color = 10;
359
360
         cout << endl;
361
         Animation::draw border(56, color);
362
         Animation::frame phrase("
color);
363
                                                        Final Project
         Animation::frame phrase("
color);
364
         Animation::frame phrase("
color);
365
         Animation::draw border (56, color);
366
         Animation::frame phrase("
color);
367
         Animation::frame phrase("
                                       Course Title: Object Oriented Programming
color);
368
         Animation::frame_phrase("
                                                   Teacher: Amber Madeeha Zeb
color);
369
         Animation::frame phrase("
color);
```

```
Animation::frame phrase("
                                                        Made by:
 , color);
371
        Animation::frame phrase("
", color);
372
         Animation::frame phrase("
                                               Manaal Waseem - FA18-BCE-074
", color);
373
        Animation::frame phrase("
", color);
374
         Animation::frame_phrase("
", color);
375
        Animation::draw border (56, color);
376
377
         while (true)
378
379
             getch();
             system("cls");
380
381
            main menu();
382
383 }
384
385 ///Display main menu screen
386 void Controller::main_menu_screen()
387
         int color = 10; ///Black background, Light green foreground
388
389
         cout << endl;
390
         Animation::draw border(36, color);
        Animation::frame_phrase("
                                                                    ", color);
391
                                                                   ", color);
         Animation::frame_phrase("
392
                                          Manaal Book shop
393
        Animation::frame_phrase("
                                                                   ", color);
394
         Animation::draw border(36, color);
395
         Animation::frame_phrase("
                                                                   ", color);
         Animation::frame_phrase("
                                                                   ", color);
396
                                              1. Login
                                                                   ", color);
397
         Animation::frame phrase("
                                              2. About
        Animation::frame_phrase("
                                                                   ", color);
398
                                                                   ", color);
         Animation::frame_phrase("
399
                                     Press any other to Exit
400
         Animation::frame_phrase("
                                                                   ", color);
401
         Animation::draw_border(36, color);
402
403
404 ///Display login screen
405 void Controller::login screen()
406 {
407
         int color = 10;
                            ///Black background, Light green foreground
408
        cout << endl;</pre>
409
410
        Animation::draw border (32, color);
        Animation::frame_phrase("
                                                               ", color);
411
         Animation::frame_phrase("
412
                                         Login as admin
                                                               ", color);
                                                               ", color);
413
         Animation::frame_phrase("
414
         Animation::draw_border(32, color);
415
         cout << endl;</pre>
416 }
417
418 ///Take login from user
419 void Controller::login user(string &lgn)
420 {
421
422
         while ((c = getch()) != '\r')
423
424
             if (c == 27) //ESC key
425
426
                 system("cls");
42.7
                 log_in();
428
429
             if (c == 8) //backspace
430
431
                 if (lgn.length() == 0)
432
                     continue;
                 cout << '\b' << " " << '\b';</pre>
433
                 lgn = lgn.substr(0, lgn.length() - 1);
434
435
                 continue;
436
437
             if (((c >= 48) && (c <= 57)) || ((c >= 65) && (c <= 90)) || ((c >= 97) &&
(c \le 122))
438
439
                 lgn += c;
```

```
cout << c;
441
442
443
         cout << endl:
444
445
446 ///Take password from user
447 void Controller::password user(string &psw)
448 {
449
         char c:
450
             while ((c = getch()) != '\r')
451
452
             if (c == 27) //ESC kev
453
454
                 system("cls");
455
                 log_in();
456
457
             if (c == 8) //backspace
458
459
                 if (psw.length() == 0)
460
                     continue;
                 cout << '\b' << " " << '\b';
461
                 psw = psw.substr(0, psw.length() - 1);
462
                 continue;
463
464
465
             if (((c >= 48) && (c <= 57)) || ((c >= 65) && (c <= 90)) || ((c >= 97) && (c
<= 122)))
466
467
                 psw += c;
cout << "*";</pre>
                                     ///Hiding the password with '*'
468
469
470
471 }
472
473 ///Display wrong login screen
474 void Controller::wrong_login_screen()
475
476
         int color = 10;  ///Black background, Light green foreground
477
478
         cout << endl:
479
         Animation::draw border (32, color);
480
         Animation::frame_phrase("
                                                                ", color);
         Animation::frame phrase(" Wrong login or password! ", color);
481
                                                               ", color);
         Animation::frame_phrase("
482
483
         Animation::draw_border(32, color);
484 }
485
486 ///Display control panel screen
487 void Controller::control_panel_screen()
488
489
         int color = 10;
         int color_lightblue = 14;
490
491
         int color white = 15;
492
493
         system("cls");
494
         cout << endl;</pre>
495
         Animation::draw border(39, color);
496
         Animation::frame phrase("
                                                                       ", color);
         Animation::frame_phrase("
                                                                       ", color);
497
                                            CONTROL PANEL
498
         Animation::frame phrase("
                                                                       ", color);
499
         Animation::draw border (39, color);
500
         Animation::frame_phrase("
                                                                       ", color);
501
         cout << "\t\t\t</pre>
502
         Animation::points();
         Animation::colorize_text("
                                         1.", color lightblue);
503
         Animation::colorize text(" Add new book ", color white);
504
505
         cout << "
506
         Animation::points();
507
         cout << endl;</pre>
         cout << "\t\t\t</pre>
508
509
         Animation::points();
         Animation::colorize text(" 2.", color lightblue);
510
         Animation::colorize_text(" Sell book ", color white);
511
512
         cout << "
         Animation::points();
513
514
         cout << endl;</pre>
```

```
515
         cout << "\t\t\t</pre>
516
        Animation::points();
        Animation::colorize text(" 3.", color lightblue);
517
        Animation::colorize_text(" Search book ", color_white);
518
519
        cout << " ";
520
        Animation::points();
521
        cout << endl;</pre>
522
        cout << "\t\t\t</pre>
523
        Animation::points();
                                      4.", color_lightblue);
        Animation::colorize text("
524
       Animation::colorize_text(" Check cashbox ", color_white);
525
526
        cout << " ";
       Animation::points();
527
        cout << " ";
528
529
        cout << endl;</pre>
        cout << "\t\t\t</pre>
530
531
       Animation::points();
532
        cout << "
                                                     · ;
533
        Animation::points();
534
        cout << endl;
535
       cout << "\t\t\t</pre>
536
        Animation::points();
        Animation::colorize_text(" 0.", color_lightblue);
537
       Animation::colorize_text(" Log out ", color_white);
538
       cout << " ";
539
540
        Animation::points();
       cout << endl;</pre>
541
       cout << "\t\t\t</pre>
542
543
       Animation::points();
544
        cout << '
545
       Animation::points();
546
        cout << endl;
547
        Animation::draw border(39, color);
548 }
549
550 ///Display cash box
551 void Controller::display_cashbox()
552 {
553
        Cashbox cash;
       int color = 10;
554
555
       system("cls");
556
          Animation::draw_border(40, color);
                    Animation::colorize text("\t\t
                                                                      Cashbox in
557
bookshop was checked
                             n'', 14);
558
                    Animation::draw_border(40, color);
559
                    cout << endl;</pre>
                    cout << "\t\t\t\sum of money: " << cash;</pre>
560
561
                    cout << endl;</pre>
562
                    cout << endl;</pre>
563
                    Animation::draw_border(40, color);
564
                    Animation::frame_phrase("
color);
                                                     Any key to go back
565
                    Animation::frame phrase("
color);
566
                    Animation::frame phrase("
                                                      to control panel
color);
567
                    Animation::frame phrase("
color);
568
                    Animation::draw_border(40, color);
569
                    while (true)
570
                    {
571
                        getch();
                       system("cls");
572
573
                        control panel();
574
575 }
576
577 ///Display add book menu screen
578 void Controller::add book menu screen()
579 {
580
         system("cls");
581
        int color = 10;
582
        cout << endl;</pre>
583
        Animation::draw border(30, color);
        Animation::frame_phrase("
                                                            ", color);
584
        Animation::frame phrase("
                                        TYPE OF BOOK
                                                           ", color);
585
```

```
586
         Animation::frame phrase("
                                                                ", color);
587
         Animation::draw border(30, color);
588
         Animation::frame phrase("
                                                                ", color);
                                                               ", color);
589
         Animation::frame_phrase("
                                         1. Paper book
         Animation::frame_phrase("
                                                                ", color);
590
                                         2. Audio book
                                                               ", color);
591
         Animation::frame_phrase("
                                         3. Digital book
                                                               ", color);
592
         Animation::frame_phrase("
         Animation::frame_pnrase("
Animation::frame_phrase(" Any other key to GO BACK ", color);
", color);
593
         Animation::frame_phrase("
594
595
         Animation::draw border(30, color);
596 }
597
598 ///Display paper book
599 void Controller::display paper book(ifstream& f1, float& price)
600 {
601
         string line;
602
         int color = 10;
         int color_lightblue = 14;
603
604
605
         Animation::draw_border(61, color);
         Animation::colorize_text("\t\t\t
606
                                                               Info about book with this ID
\n", 3);
607
         Animation::draw_border(61, color);
608
         cout << endl;</pre>
         Animation::colorize_text("\t\t\t
                                                Type: ", color lightblue);
609
         Animation::colorize_text(" Paper Book ", 3);
Animation::colorize_text("\t\t ID: ", 3);
610
611
612
         getline(f1, line);
613
         cout << line;</pre>
         cout << endl;
614
615
         Animation::colorize text("\t\t\t\ Title: ", color lightblue);
616
         getline(f1, line);
617
         cout << line << endl;</pre>
618
         Animation::colorize_text("\t\t\t Release date: ", color_lightblue);
619
         getline(f1, line);
620
         cout << line << endl;</pre>
621
         Animation::colorize text("\t\t\t
                                                 Author: ", color lightblue);
622
         getline(f1, line);
623
         cout << line << endl;</pre>
624
         Animation::colorize text("\t\t\t
                                                 genre: ", color lightblue);
625
         getline(f1, line);
626
         cout << line << endl;</pre>
         Animation::colorize text("\t\t\t
                                                 Edition: ", color lightblue);
62.7
628
         getline(f1, line);
629
         cout << line << endl;</pre>
630
         Animation::colorize text("\t\t\t
                                                 Language: ", color lightblue);
         getline(f1, line);
631
632
         cout << line << endl;</pre>
         Animation::colorize_text("\t\t\t Rating (1 -> 10): ", color_lightblue);
633
634
         getline(f1, line);
635
         cout << line << endl;</pre>
636
         Animation::colorize text("\t\t\t
                                                 Price: ", color lightblue);
637
         getline(f1, line);
638
         price = strtof((line).c_str(), 0);
         cout << line << endl;</pre>
639
         Animation::colorize_text("\t\t\t
640
                                                 Publisher: ", color_lightblue);
641
         getline(f1, line);
642
         cout << line << endl;</pre>
         Animation::colorize_text("\t\t\t Volume in pages: ", color lightblue);
643
644
         getline(f1, line);
         cout << line << endl;</pre>
645
646
         cout << endl;</pre>
         cout << endl;
647
648
649 ///Display Audio cd book
650 void Controller::display Audio cd book(ifstream& f2, float& price)
651 {
652
         int color = 10;
         int color lightblue = 14;
653
654
         string line;
655
         Animation::draw border(61, color);
656
657
         Animation::colorize_text("\t\t\t\t
                                                               Info about book with this ID
\n", 3);
658
         Animation::draw border(61, color);
         cout << endl;</pre>
659
```

```
Animation::colorize_text("\t\t\t Type: ",
Animation::colorize_text(" Audio CD Book", 3);
660
                                                 Type: ", color lightblue);
661
         Animation::colorize text("\t\t ID: ", 03);
662
663
         getline(f2, line);
664
         cout << line;</pre>
665
         cout << endl;</pre>
666
         Animation::colorize text("\t\t\t
                                                 Title: ", color lightblue);
667
         getline (f2, line);
         cout << line << endl;</pre>
668
         Animation::colorize text("\t\t\t
                                                 Release date: ", color lightblue);
669
670
         getline(f2, line);
671
         cout << line << endl;</pre>
672
         Animation::colorize text("\t\t\t
                                                 Author: ", color lightblue);
673
         getline(f2, line);
674
         cout << line << endl;</pre>
675
         Animation::colorize text("\t\t\t
                                                 genre: ", color lightblue);
676
         getline(f2, line);
677
         cout << line << endl;</pre>
678
         Animation::colorize text("\t\t\t
                                                 Edition: ", color lightblue);
679
         getline(f2, line);
680
         cout << line << endl;</pre>
         Animation::colorize_text("\t\t\t
681
                                                 Language: ", color lightblue);
682
         getline(f2, line);
         cout << line << endl;</pre>
683
684
         Animation::colorize text("\t\t\t
                                                 Rating (1 -> 10): ", color lightblue);
685
         getline(f2, line);
686
         cout << line << endl;</pre>
687
         Animation::colorize text("\t\t\t
                                                 Price: ", color lightblue);
688
         getline(f2, line);
         price = strtof((line).c_str(), 0);
689
690
         cout << line << endl;</pre>
691
         Animation::colorize_text("\t\t\t\t
                                                 Voice: ", color lightblue);
692
         getline(f2, line);
693
         cout << line << endl;</pre>
694
         Animation::colorize text("\t\t\t Listening lenth: ", color lightblue);
695
         getline(f2, line);
696
         cout << line << endl;</pre>
697
         cout << endl;</pre>
698
         cout << endl;</pre>
699
700
701 ///Display Digital_book
702 void Controller::display digital book(ifstream& f3, float& price)
703 {
704
         int color = 10;
705
         int color lightblue = 14;
706
         string line;
707
708
         Animation::draw border(61, color);
709
             Animation::colorize_text("\t\t\t
                                                                   Info about book with
                \n", 3);
this ID
710
             Animation::draw border(61, color);
711
             cout << endl;</pre>
712
             Animation::colorize text("\t\t\t Type: ", color lightblue);
             Animation::colorize text(" Digital Book ", 3);
713
             Animation::colorize_text("\t\t ID: ", 03);
714
715
             getline(f3, line);
716
             cout << line;</pre>
             cout << endl;</pre>
717
718
             Animation::colorize text("\t\t\t Title: ", color lightblue);
             getline(f3, line);
719
720
             cout << line << endl;</pre>
721
             Animation::colorize text("\t\t\t Release date: ", color lightblue);
722
             getline(f3, line);
723
             cout << line << endl;</pre>
724
             Animation::colorize text("\t\t\t
                                                     Author: ", color lightblue);
725
             getline(f3, line);
72.6
             cout << line << endl;</pre>
727
             Animation::colorize text("\t\t\t
                                                    genre: ", color lightblue);
             getline(f3, line);
728
729
             cout << line << endl;</pre>
730
             Animation::colorize text("\t\t\t Edition: ", color lightblue);
731
             getline(f3, line);
732
             cout << line << endl;</pre>
733
             Animation::colorize text("\t\t\t
                                                    Language: ", color lightblue);
734
             getline(f3, line);
```

```
735
             cout << line << endl;</pre>
736
             Animation::colorize text("\t\t\t Rating (1 -> 10): ", color lightblue);
737
             getline(f3, line);
738
             cout << line << endl;</pre>
739
             Animation::colorize text("\t\t\t
                                                   Price: ", color lightblue);
740
             getline(f3, line);
741
             price = strtof((line).c str(), 0);
742
             cout << line << endl;</pre>
             cout << endl;</pre>
743
744
             cout << endl;
745 }
746
747 ///Display book not found screen
748 void Controller::display_not_found_screen()
749 {
750
         int color = 10;
751
752
         system("cls");
753
         cout << endl;
754
         Animation::draw border(61, color);
755
         Animation::frame phrase("
", color);
756
                                           We could not found a book with this ID!
         Animation::frame phrase("
", color);
757
         Animation::frame phrase("
", color);
758
        Animation::draw border(61, color);
759
         Sleep(2000); //wait for 2000 milliseconds
760
         control panel();
761 }
762
763 ///Display paper book
764 void Controller::display paper book(ifstream& f1)
765
766
         int color = 10;
767
         int color lightblue = 14;
768
         string line;
769
770
         Animation::draw border(61, color);
                                                              Info about book with this ID
771
         Animation::colorize text("\t\t\t\t
\n", 3);
772
         Animation::draw border(61, color);
773
         cout << endl;
774
         Animation::colorize text("\t\t\t
                                               Type: ", color lightblue);
         Animation::colorize_text(" Paper Book ", 3);
Animation::colorize_text("\t\t ID: ", 3);
775
776
777
         getline(f1, line);
778
         cout << line;</pre>
779
         cout << endl;</pre>
780
         Animation::colorize_text("\t\t\t
                                                Title: ", color lightblue);
781
         getline(f1, line);
         cout << line << endl;</pre>
782
783
         Animation::colorize text("\t\t\t Release date: ", color lightblue);
         getline(f1, line);
784
785
         cout << line << endl;</pre>
         Animation::colorize_text("\t\t\t
786
                                                Author: ", color lightblue);
787
         getline(f1, line);
788
         cout << line << endl;</pre>
789
         Animation::colorize_text("\t\t\t
                                                genre: ", color lightblue);
790
         getline(f1, line);
791
         cout << line << endl;</pre>
792
         Animation::colorize text("\t\t\t
                                                Edition: ", color lightblue);
793
         getline(f1, line);
794
         cout << line << endl;</pre>
795
         Animation::colorize_text("\t\t\t
                                                Language: ", color lightblue);
796
         getline(f1, line);
797
         cout << line << endl;</pre>
798
         Animation::colorize text("\t\t\t
                                                Rating (1 -> 10): ", color lightblue);
799
         getline(f1, line);
800
         cout << line << endl;</pre>
801
         Animation::colorize text("\t\t\t
                                                Price: ", color lightblue);
802
         getline(f1, line);
803
         cout << line << endl;</pre>
804
         Animation::colorize text("\t\t\t Publisher: ", color lightblue);
805
         getline(f1, line);
         cout << line << endl;</pre>
806
```

```
807
         Animation::colorize text("\t\t\t
                                                  Volume in pages: ", color lightblue);
         getline(f1, line);
808
809
         cout << line << endl;</pre>
810
         cout << endl;</pre>
811
         cout << endl;</pre>
812
813
    ///Display Audio cd book
814
815 void Controller::display_Audio_cd_book(ifstream& f2)
816 {
817
         int color = 10;
818
         int color lightblue = 14;
819
         string line;
820
         Animation::draw_border(61, color);
Animation::colorize_text("\t\t\t\t
821
                                                                Info about book with this
822
ID
           \n", 3);
823
         Animation::draw border(61, color);
824
         cout << endl;
         Animation::colorize_text("\t\t\t Type: ", color lightblue);
825
         Animation::colorize text(" Audio CD Book ", 3);
826
         Animation::colorize_text("\t\t ID: ", 3);
827
828
         getline(f2, line);
829
         cout << line;</pre>
830
         cout << endl;</pre>
831
         Animation::colorize text("\t\t\t\ Title: ", color lightblue);
832
         getline(f2, line);
833
         cout << line << endl;</pre>
834
         Animation::colorize_text("\t\t\t Release date: ", color_lightblue);
835
         getline(f2, line);
836
         cout << line << endl;</pre>
837
         Animation::colorize_text("\t\t\t
                                                 Author: ", color lightblue);
838
         getline(f2, line);
         cout << line << endl;</pre>
839
840
         Animation::colorize text("\t\t\t
                                                 genre: ", color lightblue);
841
         getline(f2, line);
842
         cout << line << endl;</pre>
         Animation::colorize text("\t\t\t
843
                                                 Edition: ", color lightblue);
844
         getline(f2, line);
         cout << line << endl;</pre>
845
846
         Animation::colorize text("\t\t\t
                                                  Language: ", color lightblue);
847
         getline(f2, line);
         cout << line << endl;</pre>
848
849
         Animation::colorize text("\t\t\t
                                                 Rating (1 -> 10): ", color lightblue);
850
         getline(f2, line);
851
         cout << line << endl;</pre>
         Animation::colorize text("\t\t\t\t Price: ", color lightblue);
852
853
         getline(f2, line);
854
         cout << line << endl;</pre>
855
         Animation::colorize_text("\t\t\t
                                                  Voice: ", color_lightblue);
856
         getline(f2, line);
         cout << line << endl;</pre>
857
858
         Animation::colorize_text("\t\t\t Listening lenth: ", color_lightblue);
         getline(f2, line);
859
860
         cout << line << endl;</pre>
         cout << endl;</pre>
861
862
         cout << endl;</pre>
863 }
864
865 ///Display Digital book
866 void Controller::display_digital_book(ifstream& f3)
867 {
868
         int color = 10;
int color_lightblue = 14;
869
870
         string line;
871
         Animation::draw_border(61, color);
Animation::colorize_text("\t\t\t
872
873
                                                                Info about book with this
ΙD
          n'', 3);
874
         Animation::draw_border(61, color);
875
         cout << endl;</pre>
                                                 Type: ", color lightblue);
876
         Animation::colorize text("\t\t\t
         Animation::colorize_text(" Digital Book ", 3);
877
         Animation::colorize_text("\t\t\t\t
878
                                                 ID: ", 3);
         getline(f3, line);
879
880
         cout << line;</pre>
```

```
cout << endl;</pre>
882
         Animation::colorize_text("\t\t\t Title: ", color_lightblue);
883
         getline(f3, line);
884
         cout << line << endl;</pre>
         Animation::colorize text("\t\t\t
885
                                                 Release date: ", color lightblue);
886
         getline(f3, line);
887
         cout << line << endl;</pre>
888
         Animation::colorize text("\t\t\t
                                                 Author: ", color lightblue);
889
         getline(f3, line);
890
         cout << line << endl;</pre>
         Animation::colorize_text("\t\t\t
                                                 genre: ", color lightblue);
891
892
         getline(f3, line);
893
         cout << line << endl;</pre>
894
         Animation::colorize_text("\t\t\t
                                                 Edition: ", color_lightblue);
895
         getline(f3, line);
896
         cout << line << endl;</pre>
897
         Animation::colorize text("\t\t\t
                                                 Language: ", color lightblue);
898
         getline(f3, line);
899
         cout << line << endl;</pre>
900
         Animation::colorize_text("\t\t\t Rating (1 \rightarrow 10): ", color_lightblue);
901
         getline(f3, line);
902
         cout << line << endl;</pre>
         Animation::colorize_text("\t\t\t
                                                 Price: ", color_lightblue);
903
904
         getline(f3, line);
905
         cout << line << endl;</pre>
         cout << endl;</pre>
906
907
         cout << endl;</pre>
908
```

THE END