



דו"ח מסכם – פרויקט סוף תכנות מכוון אובייקטים אפליקציה לניהול השאלת ספרים דיגיטליים

מוגש ע"י: יובל המר, Y.H@e.braude.ac.il, 209158518

מוגש ע"י: עידו בן הרוש, Ido.ben.harosh@e.braude.ac.il, 316439116



בהנחיית: יעל ארז

תאריך: 28/02/25

ציון:



תוכן עניינים

3.....	תיאור מחלקות התוכנית:
3.....	מחלקת Library_App:
4.....	מחלקת DSM:
6.....	מחלקת User:
7.....	מחלקת Customer:
8.....	מחלקת Librarian:
8.....	מחלקת Book:
9.....	מחלקת Loan:
9.....	מחלקת LoanHistory:
10.....	מחלקת Reservation:
10.....	מחלקת Date:
11.....	מבנה ResCompare:
11.....	ממשק המשתמש:
12.....	דיאגרמת UML:
13.....	מבני הנתונים:
14.....	מקורות:
14.....	בעיות ידועות:
14.....	נספחים:
14.....	נספח 1:

תיאור מחלקות התוכנית:

מחלקת Library App:

המחלקה פועלת כממשק היישום הראשי עבור מערכת ניהול הספרייה. היא אחראית לאינטראקציה עם המשתמש (הן לקוחות והן ספרנים), טיפול באימות משתמשים, הצגת תפריטים למיניהן, עיבוד קלט משתמש והאצלת משימות הקשורות לנתונים למחלקת DSM (תפורט בהמשך).

:Data Members

Data Member	תיאור
DSM* m_ds	מצביע לאובייקט DSM (Data Structure Manager), המנהל את כל נתוני הספרייה, כולל ספרים, לקוחות, ספרנים, השאלות והזמנות.

:Function Members

להלן מספר פונקציות נבחרות הנמצאות במחלקה הנ"ל, בחלוקה לנושאים ופעולות:

1. אתחול המערכת וסגירתה:

Library App: בנאי האובייקט, דרכו יש קריאה לבנאי של DSM הקורא את כלל הנתונים מהקבצים ושם אותם בהתאם למבני הנתונים הרלוונטיים.
Deserialization: יציאה מהמערכת ושמירת כלל הנתונים והשינויים בקבצים. הקריאה מתבצעת על ידי ה-"הורס" (destructor) של המחלקה.

2. זיהוי התחברות המשתמש:

CheckCredentials: בדיקת התאמת נתוני הקלט על ידי המשתמש עם הנתונים המופיעים במבנה הנתונים הרלוונטי.

3. ניהול תפריט המשתמש:

Print*Menu:** הדפסת התפריט הפעולות הרלוונטי ל-*** (לקוח או ספרן) בהתאם לאימות הכניסה של המשתמש.
ParseChoice: חלוקת מחרוזת הקלט של המשתמש בבחירת פעולה מבין תפריט האפשרויות המוצע.

מחלקת DSM¹:

מחלקת זו מכילה את מבנה הנתונים של הספרייה. אחראית על ניהול וארגון כל מבני הליבה של הנתונים המשמשים במערכת ניהול ספרייה. הוא פועל כמוקד מרכזי, מטפל בספרים, לקוחות, ספרנים ומסווג את הספרים לפי מספר קטגוריות (שם ספר, כמות העותקים, זמן השאלה וכדומה). מחלקה זו משמשת עמוד השדרה לכל הגישה לנתונים, העדכונים ואינטראקציות המערכת, ומבטיחה עקביות נתונים תוך ביצוע פעולות שונות.

:Data Members

תיאור	Data Member
נתיב לקבצי קלט/פלט המשמשים לשמירה/טעינת נתונים.	<code>string m_path</code>
משמש לאתחול מגבלת ההשאלות המקסימלית בכל המערכת.	<code>const Customer Null</code>
מחסנית של הקצאות דינאמיות עבור מצביעים לספרים המשויכים למספר קטגוריות.	<code>stack<Book*> m_MulCategories</code>
מאחסנת ספרנים לפי תעודת הזהות שלהם.	<code>unordered_map<string, Librarian> m_librarians</code>
מאחסנת לקוחות לפי תעודת הזהות שלהם.	<code>unordered_map<string, Customer> m_Customers</code>
מאחסנת ספרים לפי מספר ISBN.	<code>unordered_map<string, Book> m_booksISBN</code>
מפה לא מסודרת שבה כל מפתח הוא מספר ISBN, והערך הוא תור עדיפות של לקוחות שמחכים לספר. (לפי תאריך ובחירת עדיפות של המשתמש)	<code>unordered_map<string, priority_queue<Customer*, vector<Customer*>, ResCompare>> m_ResQ</code>
מבנה נתונים המתאים שמות ספרים למצביעי הספרים המתאימים להם (מאפשר מספר ספרים עם אותו שם).	<code>unordered_multimap<string, const Book*> m_booksName</code>
מבנה נתונים המתאים שמות מחברי הספרים למצביעי הספרים המתאימים להם (מאפשר מספר ספרים עם אותם מחברים).	<code>unordered_multimap<string, const Book*> m_booksAuthors</code>
מבנה נתונים המתאים מוציא לאור למצביעי הספרים המתאימים להם (מאפשר מספר ספרים עם אותו מוציא לאור).	<code>unordered_multimap<string, const Book*> m_booksPublisher</code>
מבנה נתונים המתאים קטגוריה למצביעי הספרים המתאימים להם (מאפשר מספר ספרים עם אותה קטגוריה).	<code>unordered_multimap<string, const Book*> m_booksCategory</code>
מבנה נתונים המתאים לכמות העותקים מהספר למצביעי הספרים המתאימים להם (מאפשר מספר ספרים עם אותה כמות עותקים).	<code>unordered_multimap<string, const Book*> m_booksSimLoan</code>
מבנה נתונים המתאים זמן השאלה מקסימלי של הספר למצביעי הספרים המתאימים להם (מאפשר מספר ספרים עם זמן השאלה מקסימלי).	<code>unordered_multimap<string, const Book*> m_booksLoanTime</code>
מבנה נתונים האחראי לשלוח הודעה למשתמש במידה והתפנה ספר מרשימת ההזמנות שלו (ההעברה מתבצעת באופן אוטומטי)	<code>unordered_multimap<string, string> m_PrintLoansnISBN</code>
מבנה נתונים האחראי לשלוח הודעה למשתמש במידה והוחזר ספר מרשימת ההשאלות שלו (ההחזרה מתבצעת באופן אוטומטי)	<code>unordered_multimap<string, string> m_PrintAutoReturn</code>

¹ מחלקה זו הינה מחלקה חברה (friend) של Library_App

:Function Members

להלן מספר פונקציות נבחרות הנמצאות במחלקה הנ"ל, בחלוקה לנושאים ופעולות:

1. ניהול הספרים:

AddBook: אחראית להוספה של ספר חדש לספרייה, מתבצע עדכון בכלל מבני הנתונים הרלוונטיים. המתודה מתבצעת על ידי הספרן בלבד.

RemoveBook: אחראית על הסרה של ספר מספרייה, מתבצע עדכון בכלל מבני הנתונים הרלוונטיים. המתודה מתבצעת על ידי הספרן בלבד.

2. ניהול לקוחות:

AddClient: הוספה של לקוח חדש למערכת, עבורו נפתחות האפשרויות להשאלות והזמנות.

RemoveClient: הסרה של לקוח קיים ומחיקת כלל הנתונים שהיו ברשותו (השאלות והזמנות עתידיות)

3. ניהול הזמנות והשאלות:

LoanBook: אחראית על השאלה של ספר על ידי הלקוח ועדכון מבני הנתונים הרלוונטיים, במידה והספר אינו פנוי להשאלה, ייכנס לרשימת ההזמנות לפי התאריך הנוכחי והעדיפות של המשתמש. הלקוח ייכנס לתור העדיפויות המוגדר של הספר (הוספת הספר להשאלה או להזמנה מתבצע במחלקה Customer).

PrintResReturnMSG: הדפסת הודעת מערכת במידה והספר הוחזר באופן אוטומטי לספרייה (פקע תאריך ההשאלה) או במידה וספר התפנה והלקוח ראשון בתור ההזמנות לספר זה.

4. ניהול פרטי משתמש:

Update***: אחראי על עדכון פרטי ה-*** (לקוח או ספרן) בהתאם לרצונו ולבחירתו (שם, סיסמא או שניהם).

AddNewLibrarian: הוספה של ספרן חדש על ידי הזנת הסיסמא OOPLib.

5. ניהול קבצים:

SaveFiles: אחראי על עדכון וכתובה לכלל הקבצים בעת יציאה מהמערכת וסיום התוכנית.

WriteToFile: פונקציית Template אשר אחראית לכתוב לקובץ מסוים בהתאם למבנה נתונים ולשם הקובץ שהתקבלו כארגומנטים.

מחלקת User :

מחלקת ה-User משמשת כמחלקת בסיס אשר ממנה יורשות המחלקות Customer (לקוח/שואל בספרייה) ו-Librarian (ספרן). המחלקה מגדירה תכונות ומאפיינים המשותפים בין כל המשתמשים. המחלקה מאחסנת בתוכה מידע בסיסי על המשתמש כגון מס' תעודת זהות, שם המשתמש וסיסמא אישית. כמו כן, למחלקה נתונים סטטיים משותפים לכלל המשתמשים כדוגמת מספר הספרים המרבי שלקוח יכול לשאול - מאפיין שיכול להשתנות על ידי הספרן בלבד. המחלקה אחראית על חיפוש בעלי מאפיינים זהים לכלל המשתמשים על פי טקסט שהוזן. בנוסף מחלקה זו הינה מופשטת שכן ישנם מתודות אשר ממומשות באופן שונה במחלקות הנגזרות ממחלקה זו (לדוגמה מתודה HandleOtherCommands).

Data Members :

תיאור	Data Member
מספר זהות קבוע המזהה כל משתמש באופן ייחודי. ערך זה מוגדר ביצירת מופע במחלקה ולא ניתן לשנות אותו לאחר היצירה.	<code>const string m_id</code>
שם המשתמש, אותו ניתן לשנות באמצעות המתודה הייעודית <code>SetName()</code> .	<code>string m_name</code>
סיסמת המשתמש, ניתנת לשינוי באמצעות <code>SetPassword()</code> .	<code>string m_password</code>
משתנה סטטי המשותף לכל המשתמשים, המציין את המספר המרבי של ספרים שלקוח רשאי לשאול בכל עת. ביכולתו של הספרן לשנות ערך זה.	<code>static int m_MaxCustomerLoans</code>

Function Members :

להלן מספר פונקציות נבחרות הנמצאות במחלקה הנ"ל, בחלוקה לנושאים ופעולות :

1. חיפוש בספרייה :

*****Search** : פונקציה המחפשת התאמה במבנה הנתונים *** (ספרים, השאלות וכו') או בהתאמה לנתוני התאמות של המשתמש.

2. ביצוע פעולות משותפות :

HandleSharedCommands : אחראית על ניתוב הבחירה של המשתמש לפעולות המשותפות ללקוח ולספרן.

3. קריאת נתונים מקובץ :

ContFromCSV : אחראית לקריאת הנתונים ממאגר נתוני הספרנים והלקוחות.

מחלקת Customer :

מייצגת לקוח בודד במערכת. המחלקה יורשת ממחלקת User ומרחיבה אותה עם מאפיינים ופונקציונליות ספציפיים ללקוח הקשורים להשאלות ספרים, הזמנות והיסטוריית ההשאלות של הלקוח. מחלקה זו מנהלת את ההשאלות הנוכחיות של הלקוח, בקשות להשאלות עתידיות ורישומי השאלות בעבר, ומספק מתודות לניהול רשומות אלו, חיפוש ספרים ואינטראקציה עם מערכת הספרייה ומבנה הנתונים שלה.

Data Members

תיאור	Data Member
משתנה העוקב אחר כל הספרים המושאלים כעת על ידי הלקוח. המפתח הוא ISBN, והערך הוא אובייקט Loan.	<code>unordered_map<string, Loan> m_Loans</code>
מתעדת את היסטוריית ההשאלות של הלקוח. כל ספר שהושאל בעבר עשוי להופיע יותר מפעם אחת, כאשר כל רשומה היא אובייקט LoanHistory והמפתח הוא ISBN.	<code>unordered_multimap<string, LoanHistory> m_LoanHistory</code>
משתנה העוקב אחר ההזמנות העתידיות של הלקוח, בעת שהספר הרצוי אינו זמין להשאלה. המפתח הוא ה-ISBN והערך הוא אובייקט Reservation.	<code>unordered_map<string, Reservation> m_reservations</code>
מונה המתעד את כמות הספרים המושאלים כעת על ידי הלקוח.	<code>int m_currloans</code>

Function Members

להלן מספר פונקציות נבחרות הנמצאות במחלקה הנ"ל, בחלוקה לנושאים ופעולות:

1. ניהול השאלות, הזמנות והיסטוריה:

LoanBookNameISBN: השאלת ספר לפי ISBN או לפי השם שלו. במידה וקיימים מספר ספרים באותו השם, תינתן למשתמש אפשרות לבחור איזה ספר להשאל מבין האופציות שהוצגו בפניו.

CheckHistoryDateRange: באפשרות הלקוח לסנן את היסטוריית ההזמנות שלו לפי טווח תאריכים ולהציג את הרשומות הנמצאות בטווח זה.

DeleteLoan: הסרה של השאלה לאחר החזרתה לספרייה.

PrintCustomerFile: הדפסת כלל הרשומות הנמצאות בקובץ הנבחר (הזמנות, השאלות או היסטוריה).

PrintPriority: הדפסת העדיפויות שנבחרה להזמנה, מבין 1-5.

PrintBookCondition: הדפסת מצב הספר ברשומות ההשאלות הנוכחיות של הלקוח.

:Librarian מחלקת

מחלקה זו יורשת ממחלקת User. המחלקה מספקת את הפונקציונליות שהספרן צריך כגון חיפוש מורחב, עדכון מערכת הנתונים ועוד.

:Data Members

למחלקה זו אין משתנים נוספים מעבר לאלו הנגזרים ממחלקת User.

:Function Members

להלן מספר פונקציות נבחרות הנמצאות במחלקה הנ"ל, בחלוקה לנושאים ופעולות:

1. ניהול מגבלת השאלות של לקוחות:

SetCustomerLoans: מתודה סטטית לשינוי ועדכון כמות ההשאלות המרבית של הלקוחות. השינוי ישפיע על כלל הלקוחות במערכת.

2. חיפוש מורחב:

SearchAsLibrarian: באפשרות הספרן לחפש בכלל הרשומות הנמצאות במערכת, בהתאם למחרוזת החיפוש שהתקבלה. החיפוש נעשה גם בלקוחות אחרים ובנתונייהם ואף בספרנים אחרים.

:Book מחלקת

המחלקה מייצגת ספר בספרייה. לכל ספר יש מזהה ייחודי (ISBN) ומספר מאפיינים נוספים המרחיבים את המידע עליו ועל הזמינות שלו להשאלות.

:Data Members

תיאור	Data Member
מזהה ייחודי וקבוע של הספר, מורכב מ-13 ספרות.	<code>const string m_ISBN</code>
שם/כותרת הספר.	<code>string m_name</code>
שמות המחברים.	<code>string m_authors</code>
שם המוציא לאור.	<code>string m_publisher</code>
קטגוריית הספר (סיפורת, עיון וכו').	<code>string m_category</code>
מספר העותקים הפנויים מהספר.	<code>int m_nloans</code>
תקופת ההשאלה מקסימלית בימים.	<code>int m_maxloan</code>

:Function Members

להלן מספר פונקציות נבחרות הנמצאות במחלקה הנ"ל, בחלוקה לנושאים ופעולות:

1. תצוגת הספר:

PrintBook: הדפסה מעוצבת של פרטי הספר המבוקש.

2. עדכון פרטי ספר (Setters):

Set***: עדכון שדה *** בספר. נעשה על ידי בקשה של הספרן או בעת מצב השאלה, החזרה או הזמנה.

מחלקת Loan:

המחלקת בסיס המייצגת רישום השאלה במערכת. עוקבת אחר המידע על השאלה של ספר מסוים וכוללת בתוכו תאריך השאלה והחזרה עתידית, חלק מפרטי הספר, מצבו והזמן שנותר להחזרת הספר.

Data Members

תיאור	Data Member
מזהה ייחודי וקבוע של הספר, מורכב מ-13 ספרות.	<code>const string m_ISBN</code>
שם/כותרת הספר.	<code>string m_name</code>
קטגוריית הספר (סיפורת, עיון וכו').	<code>string m_category</code>
תאריך התחלת ההשאלה.	<code>const Date m_StartLoan</code>
תאריך סיום עתידי ומקסימלי של ההשאלה.	<code>const Date m_EndLoan</code>
מצב נוכחי של הספר.	<code>string m_BookState</code>
כמות הימים שנשארו עד להחזרת הספר.	<code>int m_DaysLeft</code>

Function Members

להלן מספר פונקציות נבחרות הנמצאות במחלקה הנ"ל, בחלוקה לנושאים ופעולות:

1. גישה לנתונים (Getters):

*****Get**: מחזיר את נתוני ההשאלה המבוקשים בשדה *** (שם הספר, תאריך תחילת ההשאלה וכו')

2. קריאה מקובץ:

fromCSVTemp: פונקציית Template אשר אחראית ליצירת אובייקטים מסוג Loan והיורשים ממנה.

מחלקת LoanHistory:

המחלקה יורשת ממחלקת Loan שתוארה קודם לכן, ומייצגת את היסטוריית השאלות הספרים של הלקוח, לרבות פרטים כגון הלקוח ששאל את הספר, פרטי הספר והתאריך שבו הספר הוחזר. מחלקה זו מספקת פונקציונליות נוספת לטיפול ברשומות הלוואות, תוך התמקדות בתאריך ההחזרה.

Data Members

נוסף למשתנים הנמצאים במחלקת Loan, המחלקה בעלת המשתנה הנוסף הנ"ל:

תיאור	Data Member
תאריך החזרת הספר מהשאלה.	<code>const Date m_ReturnDate</code>

Function Members

להלן מספר פונקציות נבחרות הנמצאות במחלקה הנ"ל, בחלוקה לנושאים ופעולות:

1. גישה לנתונים (Getters):

GetReturnDate: מחזיר את תאריך סיום ההשאלה והחזרת הספר לספרייה.

מחלקת Reservation:

המחלקה יורשת ממחלקת Loan שתוארה קודם לכן, ומייצגת הזמנת ספר שאינו פנוי, כלומר השאלה רצויה ועתידיה. המחלקה כוללת מידע על עדיפות ההזמנה (במקרה שהספר אינו זמין כעת) ותאריך ההזמנה.

Data Members

נוסף למשתנים הנמצאים במחלקת Loan, המחלקה בעלת המשתנה הנוסף הנ"ל:

תיאור	Data Member
עדיפות ההזמנה של הלקוח, ממוספר מ-1 עד 5.	<code>int m Priority</code>
מייצג את תאריך יצירת ההזמנה.	<code>const Date m ResDate</code>

Function Members

להלן מספר פונקציות נבחרות הנמצאות במחלקה הנ"ל, בחלוקה לנושאים ופעולות:

1. גישה לנתונים (Getters):

GetPriority: מחזירה את עדיפות ההזמנה שנקבעה על ידי הלקוח.

מחלקת Date:

המחלקה Date מייצגת תאריך בעל יום, חודש, שנה ונתון נוסף המאחסן את מספר הימים שעברו מאז תאריך היחסי (00/00/0001). מחלקה זו כוללת פעולות שונות לעיבוד תאריכים, עיצובם למחרוזות וביצוע השוואות תאריכים או פעולות אריתמטיות.

Data Members

תיאור	Data Member
מייצג את היום בחודש	<code>int m day</code>
מייצג את החודש בשנה	<code>int m month</code>
מייצג את השנה	<code>int m year</code>
מאחסן את מספר הימים שעברו מהתאריך היחסי	<code>unsigned m day number</code>

Function Members

להלן מספר פונקציות נבחרות הנמצאות במחלקה הנ"ל, בחלוקה לנושאים ופעולות:

1. מניפולציות על תאריכים:



add: הוספה או החסרה של כמות ימים מסוימת שהתקבלה כארגומנט.

Operator**: השוואה או החסרה (**) בין שני תאריכים, מחזירה ערך בוליאני או תוצאת החיסור.

מבנה ResCompare

משמש להשוואה בין שני אובייקטים מסוג Customers בהתבסס על ההזמנות שלהם עבור ספר ספציפי המזוהה על ידי ISBN (ונשמר במשתנה MY_ISBN). הוא כולל בנאי לאתחול ה-ISBN ואופרטור השוואה (over load operator()) כדי להשוות בין ההזמנות של שני אובייקטים של לקוחות בהתבסס על תאריכי ההזמנה והעדיפות שניתנה להם. במבנה זה נעשה שימוש לצורך ניהול תור עדיפויות של ההזמנות.

Data Members

תיאור	Data Member
מכיל את ה-ISBN של הספר לתור העדיפויות	<code>string MY_ISBN</code>

Function Members

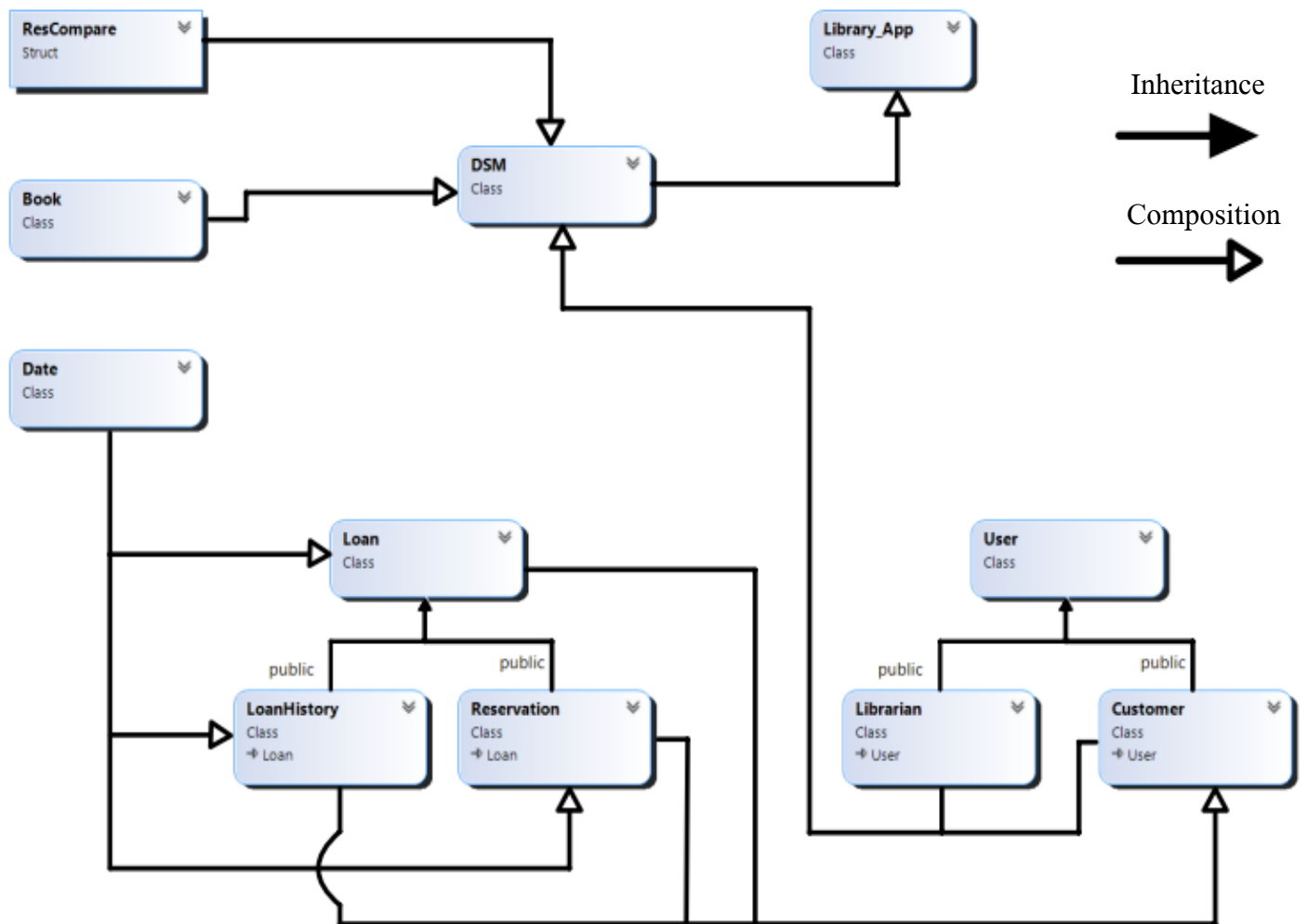
אופרטור השוואה: מבצעת השוואה בין שני אובייקטים של לקוחות על סמך ההזמנות שלהם עבור ספר ספציפי. האופרטור בודק את ההזמנות של שני הלקוחות, משווה אותם תחילה לפי תאריך ההזמנה ולאחר מכן לפי עדיפות, במידה והתאריכים זהים.

ממשק המשתמש

הממשק למשתמש הינו מערכת ה-CLI (Command Line Interface) המבוסס על DOS ואינו כולל ממשק גרפי.

בעת התחברות ללקוח או לספרן, יופיע תפריט האפשרויות לכל משתמש. בנוסף אם המשתמש הינו לקוח, ייתכן שתופיע הודעת מערכת (System Message) אשר מודיעה ללקוח זה על מעבר ספר מרשימת ההזמנות לרשימת ההשאלות (המעבר נעשה בצורה אוטומטית ורק כאשר קיימים עותקים זמינים מספר זה) או לחילופין הודעה בה הספר המושאל הוחזר בחזרה לספרייה עקב סיום זמן ההשאלה המקסימלי של ספר זה (נעשה בצורה אוטומטית גם כן).

דיאגרמת UML :





מבני נתונים:

במהלך מימוש הספרייה הדיגיטלית השתמשנו במבני נתונים רבים מאוד, מתוך הצורך לחסוך בזמן ולשפר את ביצועי התוכנית ככל הניתן. להלן מבנה הנתונים בהם השתמשנו:

Unordered map: מבנה נתונים המאחסן צמדים לפי מפתח-ערך, בדומה למילון בפייתון. מבנה זה משתמש בטבלת hash פנימית כך שחיפוש במבנה, הכנסה ומחיקה ממנו נעשים בסיבוכיות $O(1)$ בממוצע. יש לציין שבמבנה זה לכל מפתח משויך ערך אחד בלבד וסדר האלמנטים במבנה אינו מובטח. עקב תכונותיו המהירות בהכנסה מחיקה וחיפוש, מבנה זה הינו אידיאלי למימוש הספרייה (הוספה והסרה של לקוחות או ספרים, חיפוש לפי טקסט מסוים ועוד). במבנה זה נעשה שימוש בשמירת לקוחות, ספרנים, ספרים (לפי ISBN הייחודי) ותור הזמנות לפי ISBN.

Unordered multimap: דומה בתכונותיו ל-`Unordered map` אך מאפשר מפתחות כפולים, כלומר למספר אלמנטים יכולים להיות אותו המפתח. נעשה שימוש במבנה זה לצורך גישה מהירה לספרים לפי פרמטרים אשר אינם מספר ה-ISBN (לדוגמה קטגוריות ומחברי הספר). בנוסף, נעשה שימוש במבנה לצורך כתיבת הודעת מערכת למשתמש בעת החזרה אוטומטית של ספר או השאלה של ספר לאחר המתנה בתור ההזמנות.

priority queue: מבנה נתונים המייצג תור עדיפויות, כלומר, אלמנט בעל העדיפות הגבוהה ביותר תמיד יהיה ראשון בתור ויישלף ראשון. ניתן להשתמש בפונקציית השוואה מותאמת אישית (בוצע בפרויקט זה). פעולת ההכנסה של אלמנט חדש ותור והוצאתו ממנו הינו בסיבוכיות של $O(\log n)$ – הסיבוכיות הטובה ביותר לאחר $O(1)$. שליפת האלמנט מראש התור נעשית בסיבוכיות של $O(1)$. שימוש במבנה נתונים זה הקל על ניהול תור ההזמנות לפי עדיפות (פרמטר נוסף שלנו) ולפי תאריך ההזמנה.

Stack: מבנה נתונים מסוג LIFO כאשר הגישה מתבצעת לאלמנט אחרון שהוכנס בלבד. הוצאה והכנסה של אלמנטים למחסנית מתבצעת בסיבוכיות $O(1)$. חשוב לציין כי למבנה נתונים זה **אין איטרטור למעבר על הנתונים**. במבנה נתונים זה השתמשנו כדי לאחסן את המצביעים שהוקצו דינאמית כאשר המערכת נתקלה בספר בעל שתי קטגוריות או יותר. בעת יציאה מהתוכנית, נעבור בלולאה ונשחרר את הזיכרון שהוקצה בסיבוכיות הוצאה מינימלית ביותר.

מקורות:

1. כלל מצגות הקורס.
2. <https://cplusplus.com/> - מידע אחר פונקציות שונות בהם השתמשנו בפרויקט.
3. חומרי הקורס "מבני נתונים ואלגוריתמים" הנלמד במכללה.
4. <https://chatgpt.com/>, <https://claude.ai/new> - הסברים שונים להבנה עמוקה יותר של החומר.

בעיות ידועות:

לאחר מעבר מקיף על העבודה תוך ביצוע בדיקות קפדניות והרצת מבחנים שונים, **לא ידוע לנו על בעיות הקיימות במימוש הפרויקט הנ"ל.**

נספחים:

נספח 1:

נרצה לפרט כעת על שתי פונקציות משמעותיות לריצת הפרויקט:

ReadFromFile: פונקציה אשר אחראית לקריאת כלל הנתונים מקבצי ה-CSV ומתבצעת לפני הצגת התפריט למשתמש. הפונקציה הנ"ל מומשה כ-Template שכן כלל מחלקות הפרויקט השתמשו בה. הפונקציה הנ"ל יודעת לעבוד עם כל קונטיינר (מבנה נתונים) המתאים לתבנית הבאה:

```
template <class Key, class Val, class... Args> class Container = unordered_map<
```

כאשר ברירת המחדל של הקונטיינר הינו המבנה unordered map.

ניתן להבחין בפרמטרים שהתבנית מקבלת שהם: מפתח, ערך ופרמטרים נוספים (... Args).

הארגומנטים שהפונקציה מקבלת: name-שם הקובץ ו-path – ניתוב הקובץ.

לאחר הקריאה מהקבצים ויצירת האובייקטים, הפונקציה מחזירה Container<string, T>, כלומר, כל סוג של מבנה נתונים יכול לחזור מפונקציה זו כך שהמפתח הינו מחרוזת (לרוב ISBN) והערך הוא האובייקט שנוצר.

בפונקציה זו השתמשנו לצורך יצירת מבני נתונים מסוג unordered map ו-unordered multimap (לטובת היסטוריית השאלות).

CLS: פונקציה אשר אחראית לניקוי המסך למען תצוגה אסתטית וידידותית יותר למשתמש בעת הרצת התוכנית.

במידה ומערכת ההפעלה היא Windows, המערכת תשתמש בפקודה system("cls") המנקה את המסוף (תומכת בכל גרסאות של מערכת ההפעלה הנ"ל).

עבור מערכות הפעלה אחרות כדוגמת MAC או Linux, נשתמש בפקודה: `cout << "\033[2J\033[H"` הערך `"\033[2J"` מנקה את מסך המסוף (טרמינל) כאשר `"\033"` הינו תו בריחה (Escape Character) בקוד ASCII. הערך `"\033[H"` מעביר את הסמן (Cursor) לראש המסך.