Cairo University

Faculty of Computers and Information

CS251

Software Engineering I

Online University Library System

Software Analysis

Malak Ahmed Saber

Touka Mohamed Korany

Menna Tullah Hisham Mohamed

CS251: Phase 2 – <Team Name>
Project: Online Library System

Software Analysis Specification

March 2024

Contents

CS251 – CU – FCI – Software Engineering I – 2016 – Software Analysis Specifications
Prepared by Mostafa Saad and Mohammad El-Ramly
Edited by Mohamed Samir   **| 2**

CS251: Phase 2 – <Team Name>
Project: Online Library System

Software Analysis Specification

Team

ID	Name	Email	Mobile
20220341	Malak Ahmed Saber	duskee1234@gmail.com	01200970897
20220094	Touka Mohamed Korany	touka.agra@gmail.com	01205291454
20220344	Menna Tullah Hisham Mohamed	mennahesham533@gmail.com	01140799279

Document Purpose and Audience

- **what this document is. Analysis for the Requirments Document in SRS using dynamic diagram (sequence diagram) and analysis object diagram (class diagram), Analysis is the second step in SLDC process.**
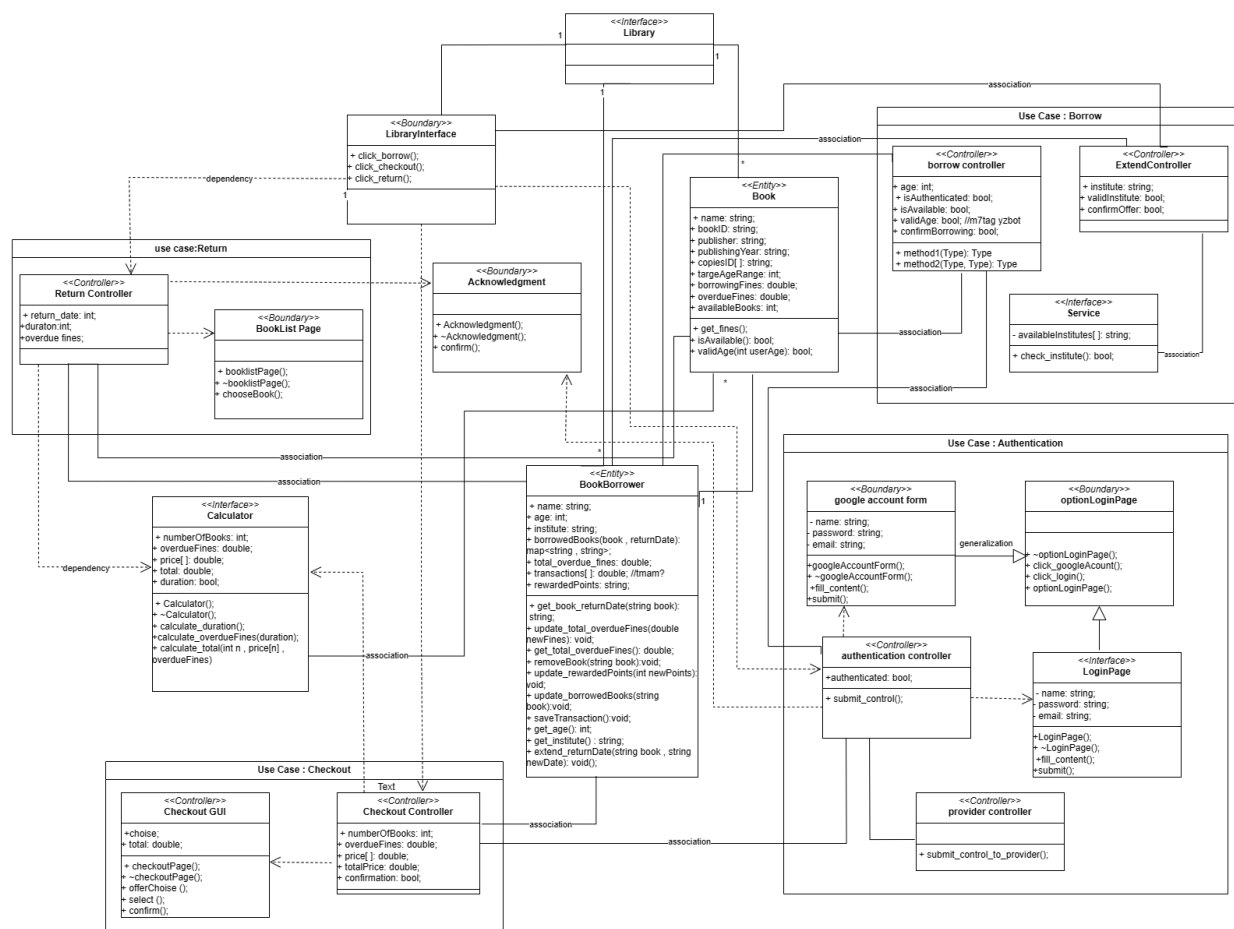- **The Target Audience : Project Manager, Project Developers**

CS251: Phase 2 – <Team Name>
Project: Online Library System

Software Analysis Specification

System Models

I. Class diagrams

<<Interface>>
Library

<<Boundary>>
LibraryInterface
- click_borrow();
- click_checkout();
- click_return();

<<Boundary>>
Acknowledgment
+ Acknowledgment();
~ Acknowledgment();
+ confirm();

use case:Return

<<Controller>>
Return Controller
+ return_date: int;
+ duraton:int;
+ overdue fines;

<<Boundary>>
BookList Page
+ booklistPage();
+ ~booklistPage();
+ chooseBook();

<<Entity>>
Book
+ name: string;
+ bookID: string;
+ publisher: string;
+ publishingYear: string;
+ copiesID[ ]: string;
+ targeAgeRange: int;
+ borrowingFines: double;
+ overdueFines: double;
+ availableBooks: int;
+ get_fines();
+ isAvailable(): bool;
+ validAge(int userAge): bool;

Use Case : Borrow

<<Controller>>
borrow controller
+ age: int;
+ isAuthenticated: bool;
+ isAvailable: bool;
+ validAge: bool; //m7tag yzbot
+ confirmBorrowing: bool;
+ method1(Type): Type
+ method2(Type, Type): Type

<<Controller>>
ExtendController
+ institute: string;
+ validInstitute: bool;
+ confirmOffer: bool;

<<Interface>>
Service
- availableInstitutes[ ]: string;
+ check_institute(): bool;

<<Interface>>
Calculator
+ numberOfBooks: int;
+ overdueFines: double;
+ price[ ]: double;
+ total: double;
+ duration: bool;
+ Calculator();
+ ~Calculator();
+ calculate_duration();
+ calculate_overdueFines(duration);
+ calculate_total(int n , price[n] , overdueFines)

<<Entity>>
BookBorrower
+ name: string;
+ age: int;
+ institute: string;
+ borrowedBooks(book , returnDate): map<string , string>;
+ total_overdue_fines: double;
+ transactions[ ]: double; //tmam?
+ rewardedPoints: string;
+ get_book_returnDate(string book): string;
+ update_total_overdueFines(double newFines): void;
+ get_total_overdueFines(): double;
+ removeBook(string book):void;
+ update_rewardedPoints(int newPoints):void;
+ update_borrowedBooks(string book):void;
+ saveTransaction():void;
+ get_age(): int;
+ get_institute() : string;
+ extend_returnDate(string book , string newDate): void();

Use Case : Authentication

<<Boundary>>
google account form
- name: string;
- password: string;
- email: string;
+ googleAccountForm();
+ ~googleAccountForm();
+ fill_content();
+ submit();

<<Boundary>>
optionLoginPage
+ ~optionLoginPage();
+ click_googleAcount();
+ click_login();
+ optionLoginPage();

<<Controller>>
authentication controller
+ authenticated: bool;
+ submit_control();

<<Interface>>
LoginPage
- name: string;
- password: string;
- email: string;
+ LoginPage();
+ ~LoginPage();
+ fill_content();
+ submit();

Use Case : Checkout

<<Controller>>
Checkout GUI
+ choise;
+ total: double;
+ checkoutPage();
+ ~checkoutPage();
+ offerChoise ();
+ select ();
+ confirm();

<<Controller>>
Checkout Controller
+ numberOfBooks: int;
+ overdueFines: double;
+ price[ ]: double;
+ totalPrice: double;
+ confirmation: bool;

<<Controller>>
provider controller
+ submit_control_to_provider();

CS251: Phase 2 – <Team Name>
Project: Online Library System

Software Analysis Specification

Class ID	Class Name	Description & Responsibility
1	CheckOut GUI	Lets the user interact with the controller, viewed only when pressing checkout button.
2	CheckOut Controller	Manages the Interactions between other objects in the checkout.
3	Calculator	Calculates the total charge in the "checkout" and the overdue fines in "return".
4	BookBorrower	Saves user data.
5	LibraryInterface	Shows the main page.
6	Book	Saves book data and details.
7	optionLoginPage	Viewed only when click borrowBook, and shows two options to login as certified member university.
8	Google account form	Allows the member university to enter his data.
9	Authentication controller	Manages interaction between classes by creating optionLoginPage , google account form and acknowledgement.
10	Provider control (Authentication Use case)	Allows communication between book borrower controller and identity provider controller by authentication.
11	Provider identity	Checks authentication certified.
12	Book list page	Viewed only when click on returnButton (the list of borrowed book), Allows to choose book to return it .
13	Return controller	Manages interaction between classes by create calculator and acknowledgement.
14	Acknowledgement	Announce the completion of an action and informs the user of it.
15	Service	Confirms that user belongs to one/some of the syndicates organizations.
16	ExternalController	Manages the Interactions between other objects in the Borrow.
17	BorrowController (Borrow use case)	Manages the interactions between syndicates organizations service provider and the external controller.

Software Analysis Specification

Important Algorithm

- **Checkout : calculateSpecifiedAmount(Totaloverduefines) -> adds a specific amount of fines on the borrowing fines (amount of overdue fines [from the Total overduefines on the user] less than a specific amount) AND decrements the Total overdue fines on the user by this specific amount, if Total overdue fines on the user are less than that specific amount, let the specific amount equal the remaining overdue fines and let Total overdue fines on BookBorrower equals ZERO.**

- **Return : after the User chooses a book to return, the controller gets the original return date and calculates the duration (difference between the original return date and the actual return date) if duration exceeded he marks duration as 'true' then gets the borrowing fines of the book and calculates the overdue fines, if duration was 'true' overdue fines are added to TotalOverdue Fines of the user, otherwise overdue fines are set to ZERO. Then controller updates the new TotalOverdue fines.**

CS251: Phase 2 – <Team Name>
Project: Online Library System

Software Analysis Specification

II. Sequence diagrams

Checkout Sequence Diagram (Use Case ID : 1)
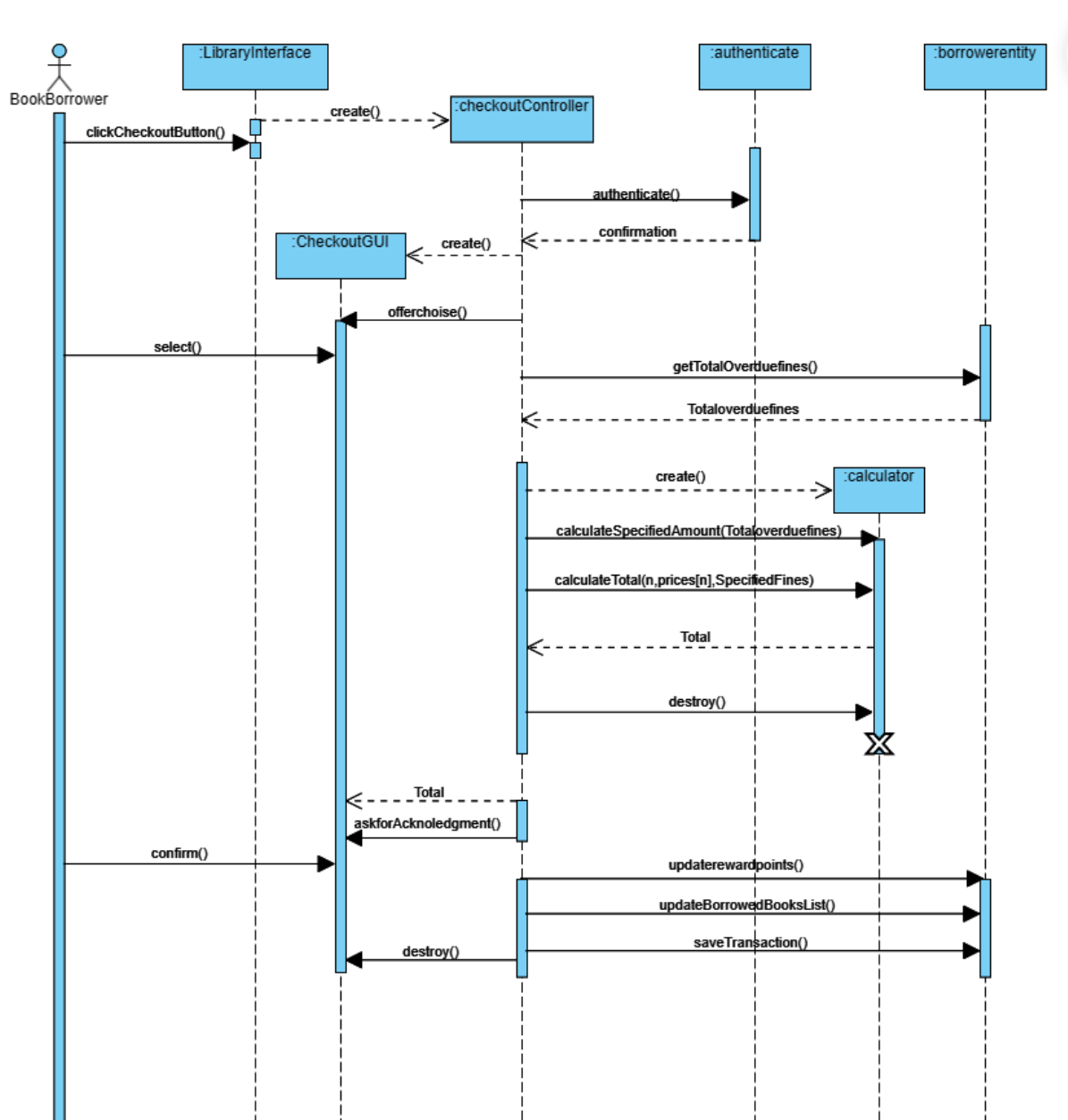
Software Analysis Specification

Authentication Sequence Diagram (Use Case ID : 2)
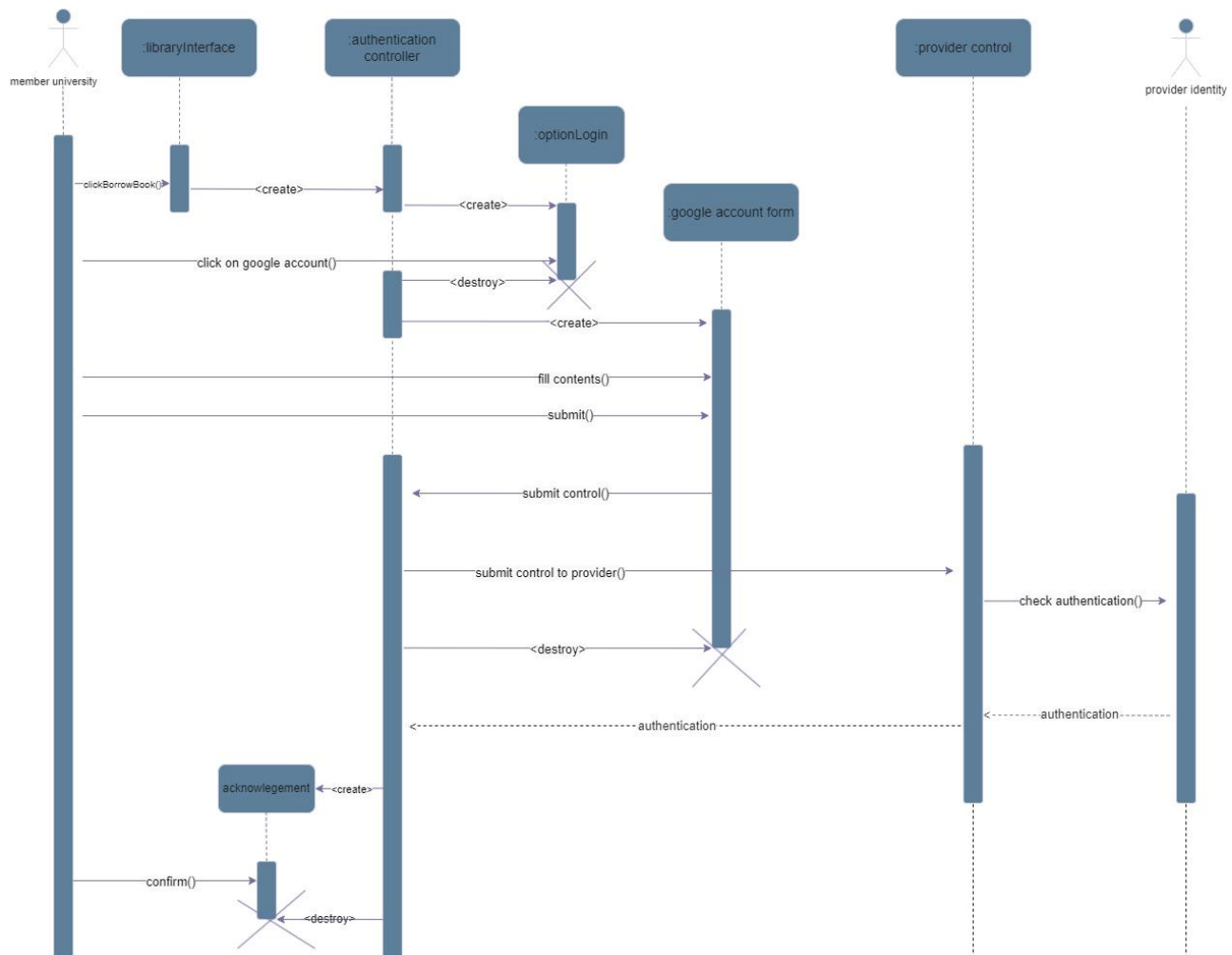
Software Analysis Specification

Borrow Sequence Diagram (Use Case ID : 3)
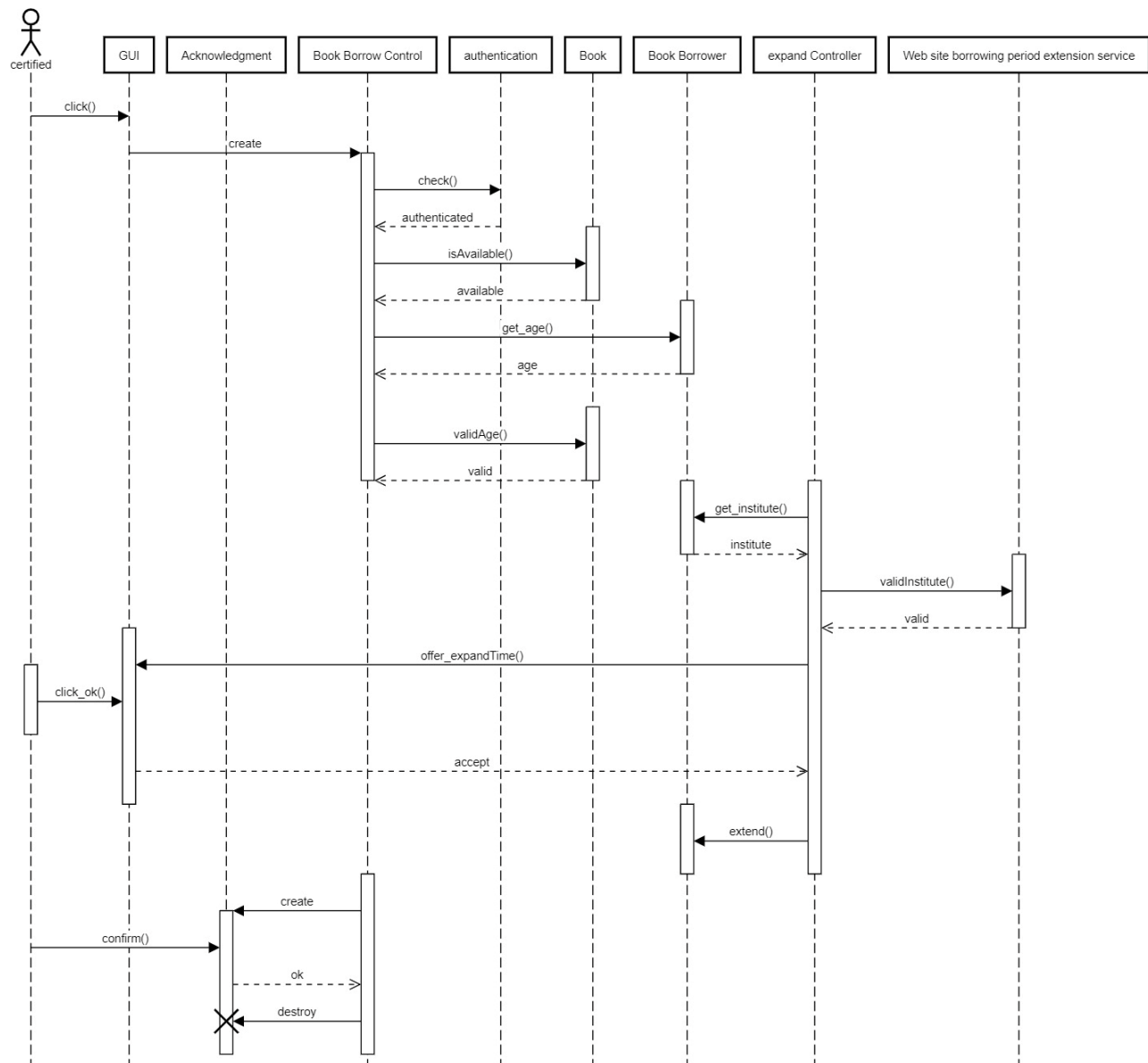
Return Sequence Diagram (Use Case ID : 4)
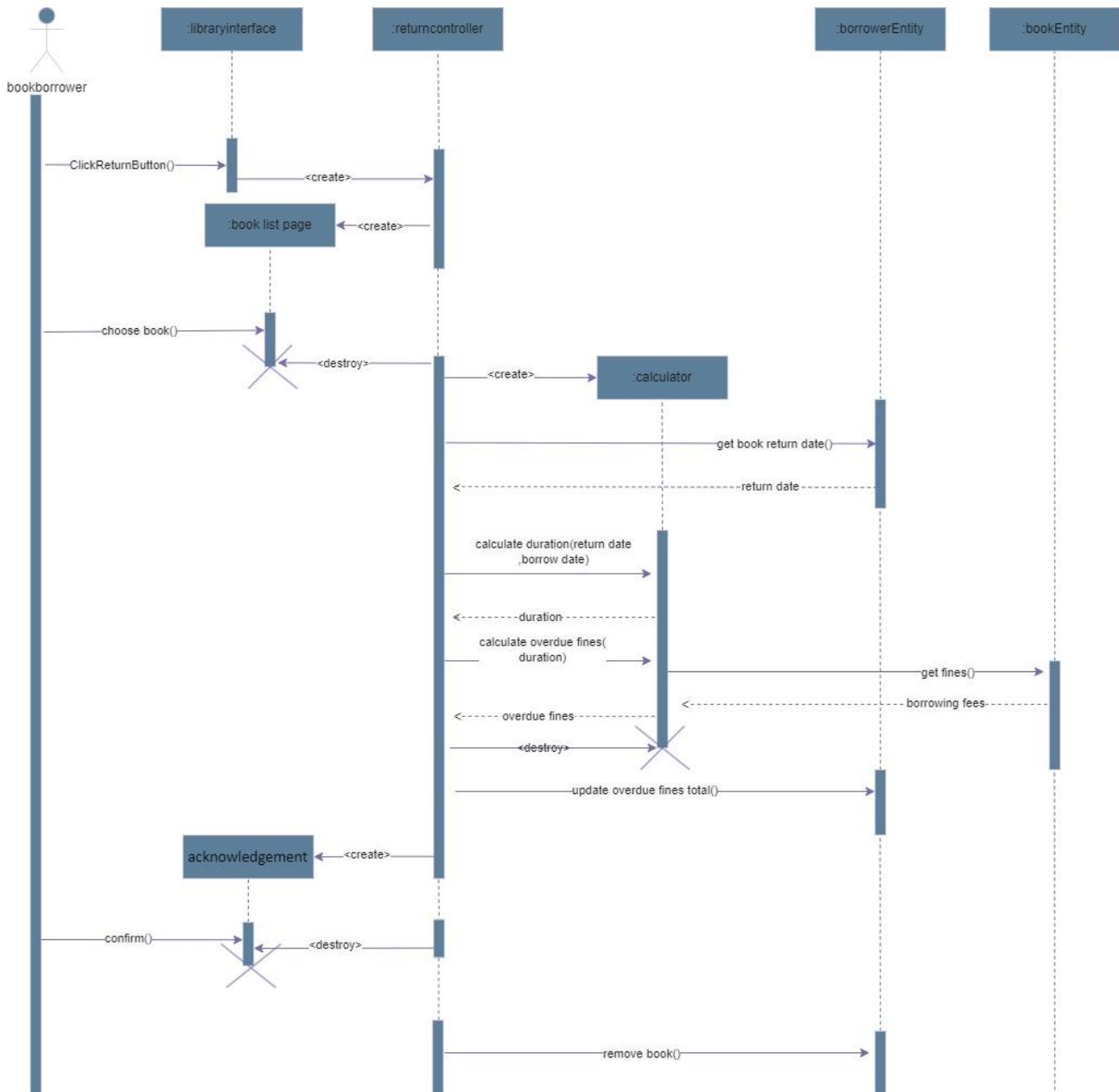
Software Analysis Specification

Software Analysis Specification

Ownership Report

Item	Owners
Checkout Sequence Diagram	*Malak*
Authorization Sequence Diagram	*Menna*
Borrow Sequence Diagram	*Touka*
Return Sequence Diagram	*Menna (All)* *Malak (shared in draft)* *Touka (shared in draft)*
Checkout classes	*Malak*
Authentication and Return classes	*Menna*
Borrow classes	*Touka*
Class Diagram	**ALL team members**
Classes Description	*Malak* *Menna*
Important Algorithm	*Malak*

Policy Regarding Plagiarism:

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
5. إذا لم تكن متأكدا أن فعلاً ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر.

CS251: Phase 2 – <Team Name>
Project: Online Library System

Software Analysis Specification

References

- http://www.mhhe.com/engcs/compsci/pressman/graphics/Pressman5sepa/common/cs1/design.pdf

Authors

- Mostafa Saad and Mohammad El-Ramly (Edited by Mohamed Samir)