

CSE232: Advanced Software Engineering

Bookstore Software

Collaborators

Esraa Ramadan Aly	1808069
Basel mohamed ramadan	18p8121
Khalid Mohamed Mahmoud Salman	19p5482
Ahmed Khaled Saad Ali	1809799
Sohaila mohamed anwar	18P6717
Mohamed Khaled Talaat	1802559

Table of Contents

1. Introduction:	4
1.1 Purpose:	4
1.2 List of Definitions	4
1.3 Scope:	4
1.4 Overview:	5
2. General Description	5
2.1 Product Perspective	5
2.2 General Capabilities	5
2.3 General Constraints	6
2.5 Environment Description	6
2.6 Assumptions and Dependencies	7
3. Specific Requirements	7
3.1 Capability Requirements 1) Admins can:	7
3.2 Constraint Requirements	8
4. Use Case Diagram and Narrative Description:.....	8
➤	10
4.1 Use-Case description	11
5. Swim lane Diagram:	18
6. Noun Extraction and CRC Cards:	20
7. Class Model	21
7.1 Client – Object relation Diagram	21
7.2 Class Diagram	22
7.3 OOAD Methodologies:	23
I. Rumbaugh et. al.'s Object Modeling Technique (OMT):	23
(a) Analysis phase which consists of three different parts:	23
➤	25
(b) System design	26
(c) Object design	26
(d) Implementation	26
II. The Booch Methodology.....	27
(1) Analysis phase which consists of the following diagrams.....	27
➤ Design phase which prescribes two parts:.....	28

7.4 Comparative Analysis of the Output of the Adopted Methodologies:	29
1st Methodology: (Object Modeling Technique O.M.T)	29
2nd Methodology: (Booch methodology)	31
8. State Diagram	32
.....	33
9. Interaction Diagram	34
10. Architectural Model:	39
10.1 System Architecture	39
We will make the system architecture as that the Book store system will be divided into sub-systems, making for each of them a system architecture.	39
11. Component Diagram:	41
12. Testing:	43
12.1 Testing Strategy	43
12.1.1 Introduction	43
12.1.2 Test Procedure	44
13. Estimated project cost	47
14. User Guide:	49
➤ 14.1 Main Window	49
➤ 14.1.1 Login Window	50
➤ 14.1.2 Admin Window	51
14.2 End-User Guide	52
Admin Guide	52
1.1.1 Reader Guide	59
➤ 14.3 Reader Window	61
15. Time Plan:	62
16. Conclusion:	62

List Of Figures

Figure 1: Use Case Diagram #1	9
Figure 2: Use Case Diagram # 2	10
Figure 3: Use Case Diagram #2	17
Figure 4: Swim lane Diagram #1	18
Figure 5: Swimlane Diagram #2	19
Figure 6: CRC Card #1	21
Figure 7: Client-Object relation Diagram	21

Figure 8: Class Diagram	22
Figure 9: Object Diagram	23
Figure 10:Context Diagram	24
Figure 11: Level 0 DFD.....	25
Figure 12:Level 1 DFD	25
Figure 13:Level 2 DFD	26
Figure 14: Object Diagram	27
Figure 15: State diagram from readers view	32
Figure 16: State Diagram #2.....	33
Figure 17: Sequence Diagram (Customer)	34
Figure 18: Sequence diagram (Admin).....	36
Figure 19: Black Board	39
Figure 20: Layered Architecture.....	39
Figure 21: MVC Architecture model	40
Figure 22: Component Diagram.....	41
Figure 23:Time Plan	62

1. Introduction:

1.1 Purpose:

Project purpose is to create a book store system, managing admin , readers, books, categories and buying books.

1.2 List of Definitions

- ❖ **CRC:** Class-responsibility-collaboration cards are a brainstorming tool used in the design of object-oriented software. They were originally proposed by Ward Cunningham and Kent Beck as a teaching tool, but are also popular among expert designers and recommended by extreme programming supporters. CRC cards are part of the design phase within system/software development and gives a good overview if you go from use case descriptions to CRC cards and then to class diagrams. This allows a smoother transition with a greater overview and allows the developer to easier implement a system with low binding and high cohesion.
- ❖ **OOAD:** Object-oriented analysis and design is a software engineering approach that models a system as a group of interacting objects. Each object represents some entity of interest in the system being modeled, and is characterized by its class, its state (data elements), and its behavior.
- ❖ **DFD:** A Data Flow Diagram is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored. The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

1.3 Scope:

Our online book store software system have two users: the admin who works in the library and the reader both have to log in the application.

The admin can add categories or update ones as well as adding new books writing their name, author, price, category,...etc. He can also update a books information or delete one. The admin is

the one who creates readers accounts information and add them to the system so that the reader can log in later.

When the reader logs in he is shown the available books to buy, adds what he needs to the cart then checks out from a window that shows his purchases or removes items from the cart if he wants to, at last presses confirm order.

1.4 Overview:

In this report we want to clarify all the steps and ideas taken to made (diagrams, methodologies) to develop our bookstore software system in order to make our online bookstore system functional and provide the main functionalities provided by any bookstore .

This report is going to further explain the procedures taken and explain each diagram with detailed explanation on how and why each diagram was made in this way.

2. General Description

2.1 Product Perspective

- ❖ This product aims to develop a software that facilitates the data storage and data maintenance and its redeem for the online bookstore in a moderate way. The admin has the privileges to access, modify and delete categories, books and description that the bookstore provides to its customers. Also, the admin is responsible for adding the readers'/ customers' account. All records are stored in a single database.
- ❖ Different users have different permission to access this program.
- ❖ Each user has unique id. Users don't have right to alter other customer records.
- ❖ It is very easy to operate as it is made user friendly with the help of very effective GUI tools.

2.2 General Capabilities

This system includes a system where customers can view the categories and books that the admin has added with their price, and description. The customers can add and delete from the cart and set their shipping address.

2.3 General Constraints

- ❖ By examining the statistics of visitors, it is possible to see shortcomings of the site and look to improve those areas. Therefore, it is possible to increase or decrease the popularity of a site and the number of people that visit it.
- ❖ The program considers an easy access to all the functionality of the application whether the user organization is trained enough to use it or not.
- ❖ The developed software is truly efficient and can work on any circumstances or traditions.
- ❖ The user can't login without getting their ID, Username, password from the admin
- ❖ The user can't cancel the order after confirming

2.4 User Characteristics

In this software system we have 2 levels of users: Manager module and User module.

- ❖ In manager module, this user is an admin type who has full rights on the system and can control most of the program services like:
 - registering new customers' accounts and removing existing ones
 - adding and removing books, books' details, description, and prices.
 - That type of user is required to be trained how to use the system.
- ❖ In user ("Reader") module, users can: view the books and categories provided by the admin and can choose from them to add to the cart. He can also remove from the cart and confirm purchasing his order. They are the main users of the software services.

2.5 Environment Description

Our software management system is designed for online bookstore. It will be used in windows platform in the version of windows 7 or above.

Other details:

- Operating system: Windows platform
- Processor speed: 2.5 GHz
- RAM: 512 MB
- Hard disk drive: 50 GB

2.6 Assumptions and Dependencies

- ❖ This software package is supported by all versions of window operating systems.
- ❖ All functionalities the system could make require the application to be downloaded and a stable internet connection.
- ❖ It contains an online data base that stores the data and information entered by the users which will make data correct and up to date.
- ❖ The application serves a real life bookstore and so the users have to visit the store to register a new account and they'll get their account details (ID, username, password) once the admin registers it on the system.
- ❖ The quantity of a book is set up to 30, and it's part of the number of books that are present at the store.

3. Specific Requirements

3.1 Capability Requirements

1) Admins can:

- Sign in
- Create users' accounts
- Add, remove, and update books
- Add, remove and update categories
- Add, remove, and update books' description
- Add, remove, and update books' prices

2) Users can:

- Sign in
- Choose category
- View books details, description, and price
- Choose the quantity up to 30 books of the same book
- Add to cart
- Remove from the cart
- Confirm purchasing

3.2 Constraint Requirements

- ❖ The software is build using Java language.
- ❖ The project's budget is around 350,000 LE.
- ❖ Able to secure the accounts and data of its users.
- ❖ Users must be connected to internet to receive latest changes.
- ❖ Response time averages around 50 ms with a stable internet connection and no more than 400 ms with weak connections.

4. Use Case Diagram and Narrative Description:

A use case diagram is a visual representation of how a user might interact with a technology. A use case diagram depicts the system's numerous use cases and different sorts of users, and is frequently supplemented by other diagrams. Circles or ellipses are used to depict the use cases. The actors are frequently depicted as stick figures.



Figure 1: Use Case Diagram #1

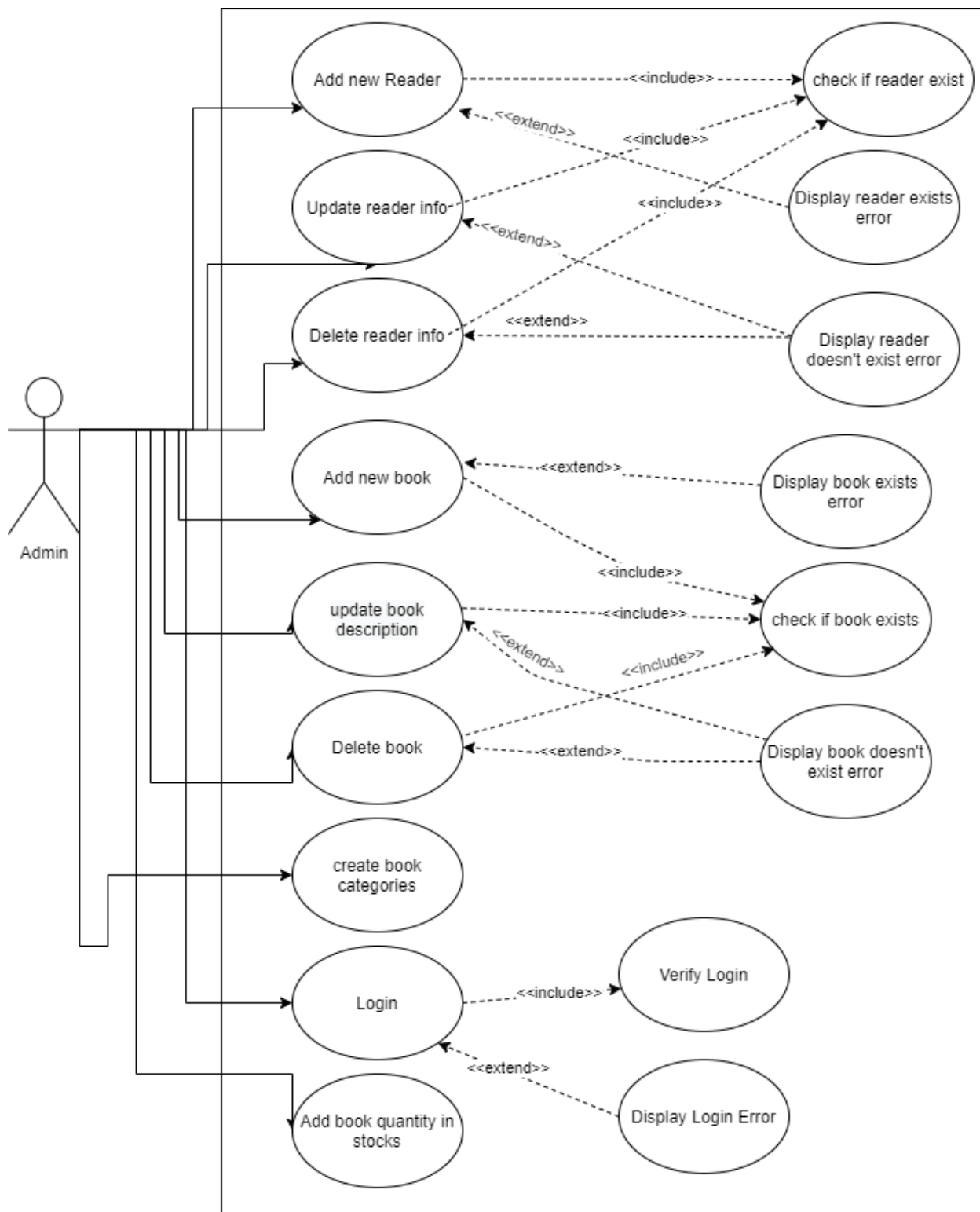


Figure 2: Use Case Diagram # 2

4.1 Use-Case description

Use case name Add new reader.

Related Requirements

Goal in context Add a new reader's info into the database.

Preconditions Reader not already existing in the database.

Successful end condition Reader info added into the database.

Failed end condition Reader not added into the database.

Primary actor Admin

Main flow	Step	Action
	1	Admin opens the Reader section in the manager window.
	2	Admin fills the reader's info form.
	3	Admin clicks on "New Reader" button to add the reader into the data base.
	4	Admin clicks on the "confirm Added".
	5	Admin clicks on the "Insert" button so the reader can benefit from the services of the application.

Use case name	Update reader info.	
Related Requirements		
Goal in context	Update reader's info in the database.	
Preconditions	Reader already exists in the database.	
Successful end condition	Reader info updated in the database.	
Failed end condition	Reader info not updated in the database.	
Primary actor	Admin	
Main flow	Step	Action
	1	Admin opens the Reader section in the manager window.
	2	Admin enters the reader's id.
	3	Admin fills the new info in the form.
	4	Admin clicks on the "update reader" button

Use case name	Delete reader info.	
Related Requirements		
Goal in context	Delete reader's info from the database.	
Preconditions	Reader already exists in the database.	
Successful end condition	Reader info Removed from the database.	
Failed end condition	Reader info remaining in the database.	
Primary actor	Admin	
Main flow	Step	Action
	1	Admin opens the Reader section in the manager window.
	2	Admin enters the reader's id.
	3	Admin clicks on "Remove" button to remove the reader from the data base.
	4	Admin clicks on the "confirm delete".

Use case name	Add new book.	
Related Requirements		
Goal in context	Add a new book into the database.	
Preconditions	Book not already existing in the database.	
Successful end condition	Book added into the database.	
Failed end condition	Book not added into the database.	
Primary actor	Admin	
Main flow	Step	Action
	1	Admin opens the Book section in the manager window.
	2	Admin clicks on “New Book”.
	3	Admin fills the book’s info in the form.
	4	Admin clicks on the “Add new Product”.
	5	Admin clicks on the “confirm Added”.

Use case name	Update book info.	
Related Requirements		
Goal in context	Update book’s info in the database.	
Preconditions	Book already exists in the database.	
Successful end condition	Book info updated in the database.	
Failed end condition	Book info not updated in the database.	
Primary actor	Admin	
Main flow	Step	Action
	1	Admin opens the Book section in the manager window.
	2	Admin selects the book.
	3	Admin fills the new info in the form.
	4	Admin clicks on the “update book” button.

Use case name Delete Book.

Related Requirements

Goal in context Delete book's info from the database.

Preconditions Book already exists in the database.

Successful end condition Book info Removed from the database.

Failed end condition Book info remaining in the database.

Primary actor Admin

Main flow	Step	Action
	1	Admin opens the Book section in the manager window.
	2	Admin selects the book.
	3	Admin clicks on the "Delete Book" button.

Use case name Create book category.

Related Requirements

Goal in context Create a new book category.

Preconditions Category not already existing in the database.

Successful end condition Category created in the database.

Failed end condition Category not created in the database.

Primary actor Admin

Main flow	Step	Action
	1	Admin opens the Category section in the manager window.
	2	Admin clicks on "Add New Category".
	3	Admin fills the category's info in the form.
	4	Admin clicks on the "Add New Category".

Use case name	Check if reader exists.	
Related Requirements		
Goal in context	Reader's info need to be checked on in the database.	
Preconditions	Reader's Id being available.	
Successful end condition	Reader exists in the database.	
Failed end condition	Reader does not exist in the database.	
Primary actor	Admin	
Main flow	Step	Action
	1	The reader's id is provided to the system.
	2	System checks for a reader with a matching id.
	3	The result is returned.
Use case name	Display reader exists error.	
Related Requirements		
Goal in context	Display an error message in case the new added user already exists.	
Preconditions	New user being added already exists.	
Successful end condition	Error message displayed.	
Failed end condition	No error message is displayed.	
Primary actor	Admin	
Main flow	Step	Action
	1	Admin adds an already existing user in the database.
	2	System checks for a reader with a matching id.
	3	Error message is displayed.

Use case name	Login.
Related Requirements	
Goal in context	Login the admin into his account.
Preconditions	Admin inserts his login credentials.
Successful end condition	Admins gets logged into his account.
Failed end condition	Admin does not login into his account.
Primary actor	Admin

Main flow	Step	Action
	1	Admin selects “Admin Login”.
	2	Admin enters his login credentials.
	3	System validates the credentials.
	4	Admin is logged into his account.

Use case name	Add book to cart.	
Related Requirements		
Goal in context	Reader adding books to his cart.	
Preconditions	Reader choosing a book to add.	
Successful end condition	Book gets added to the reader’s cart.	
Failed end condition	Book does not get added to the reader’s cart.	
Primary actor	Reader.	
Main flow	Step	Action
	1	Reader clicks the “Buy Book” button.
	2	Reader chooses the book he wants.
	3	Reader selects the quantity he wants.
	4	Reader clicks the “Add Cart” button.

Use case name Confirm order.

Related Requirements

Goal in context Confirm the reader's order.

Preconditions Cart has items.

Successful end condition order confirmed.

Failed end condition order not confirmed.

Primary actor Reader.

Main flow

Step	Action
1	Reader clicks the "View Cart" button.
2	Reader views his items.
3	Reader clicks the "Product to checkout" button.
4	Reader provides the shipping address.
5	Reader clicks the "Confirm Order" button.

5. Swim lane Diagram:

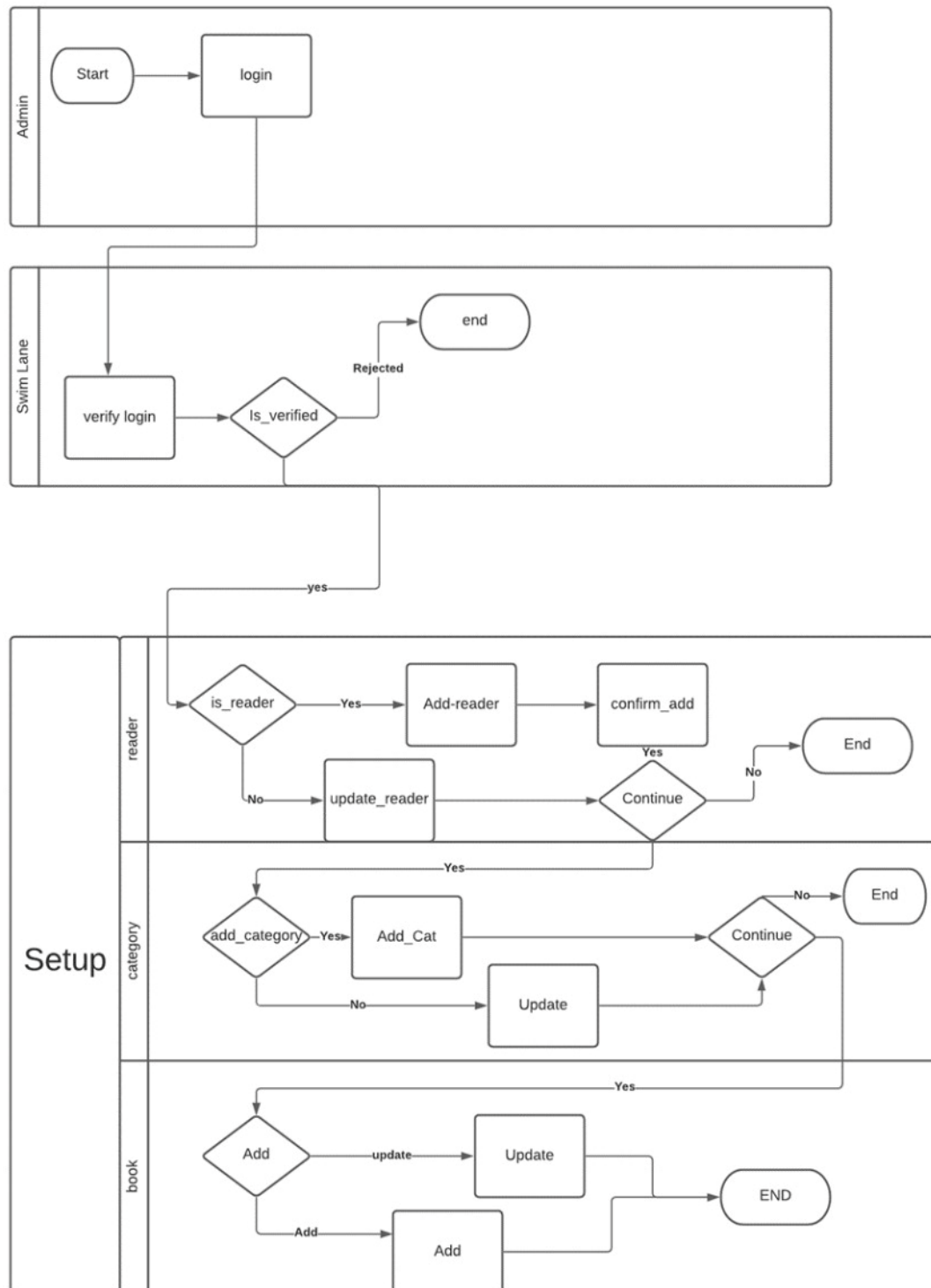


Figure 4: Swim lane Diagram #1

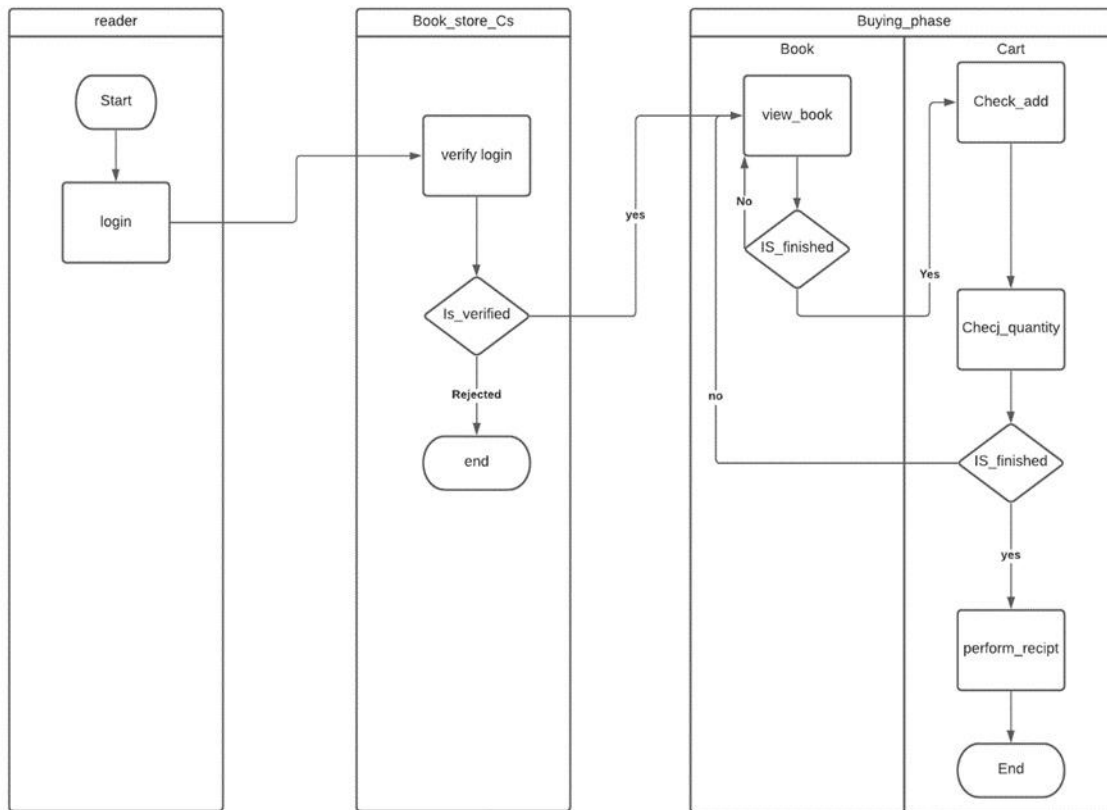


Figure 5: Swimlane Diagram #2

Swim lane diagrams aim to specify who does what in each process therefore it adds clarity and accountability as each user has all of the processes laid in front of it in its own lane as in the following figure.

6. Noun Extraction and CRC Cards:

- In our book store software system two users: the admin who works in the library and the reader both have to log in the application.
- The admin can add categories or update ones as well as adding new books writing their name, author, price, category,...etc. He can also update a books information or delete one. The admin is the one who creates readers accounts information and add them to the system so that the reader can log in later.
- When the reader logs in he is shown the available books to buy, adds what he needs to the cart then checks out from a window that shows his purchases or removes items from the cart if he wants to, at last presses confirm order.
- Nouns: bookstore, users, admin, library, reader, application, categories, books, name, author, price, information, accounts, cart, window, purchases, order.
- The noun extraction and CRC cards are more like tools than diagrams in that they help with brainstorming and object-oriented design. They are frequently used to lay out concepts and then test each design using multiple scenarios

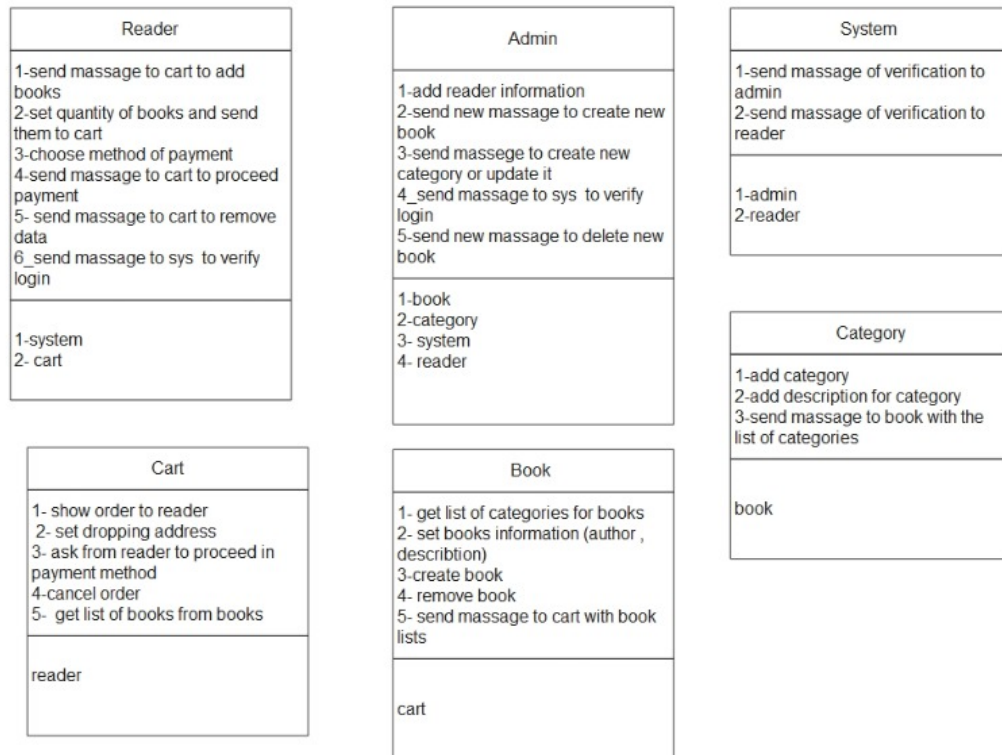


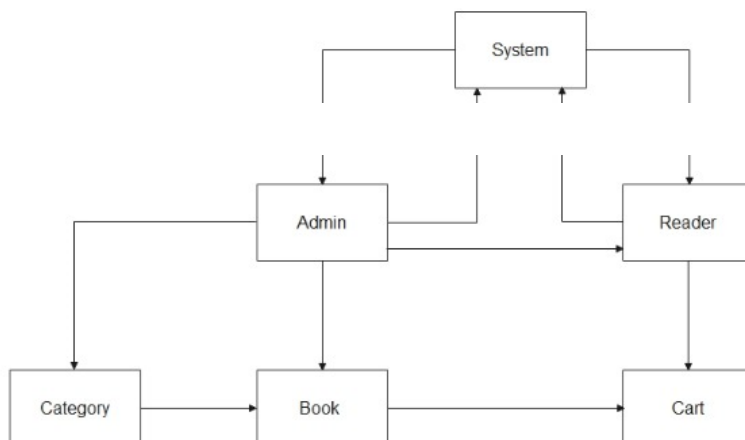
Figure 6: CRC Card #1

The CRC cards are a useful tool that enables us to understand what needed to be done in the project so that the entire team could work together!

7. Class Model

7.1 Client – Object relation Diagram

Figure 7: Client-Object relation Diagram



7.2 Class Diagram

The class diagram is main building unit of any software based on object-oriented concepts as it is used for general modelling of the whole software structure and even translating the models into understandable programming code (pseudo code).

We started working on our system's class diagram as soon as we finished the primary design phases; In order to help us during the programming process as it worked as some kind of a reference for all of the team members to understand what was going to be.

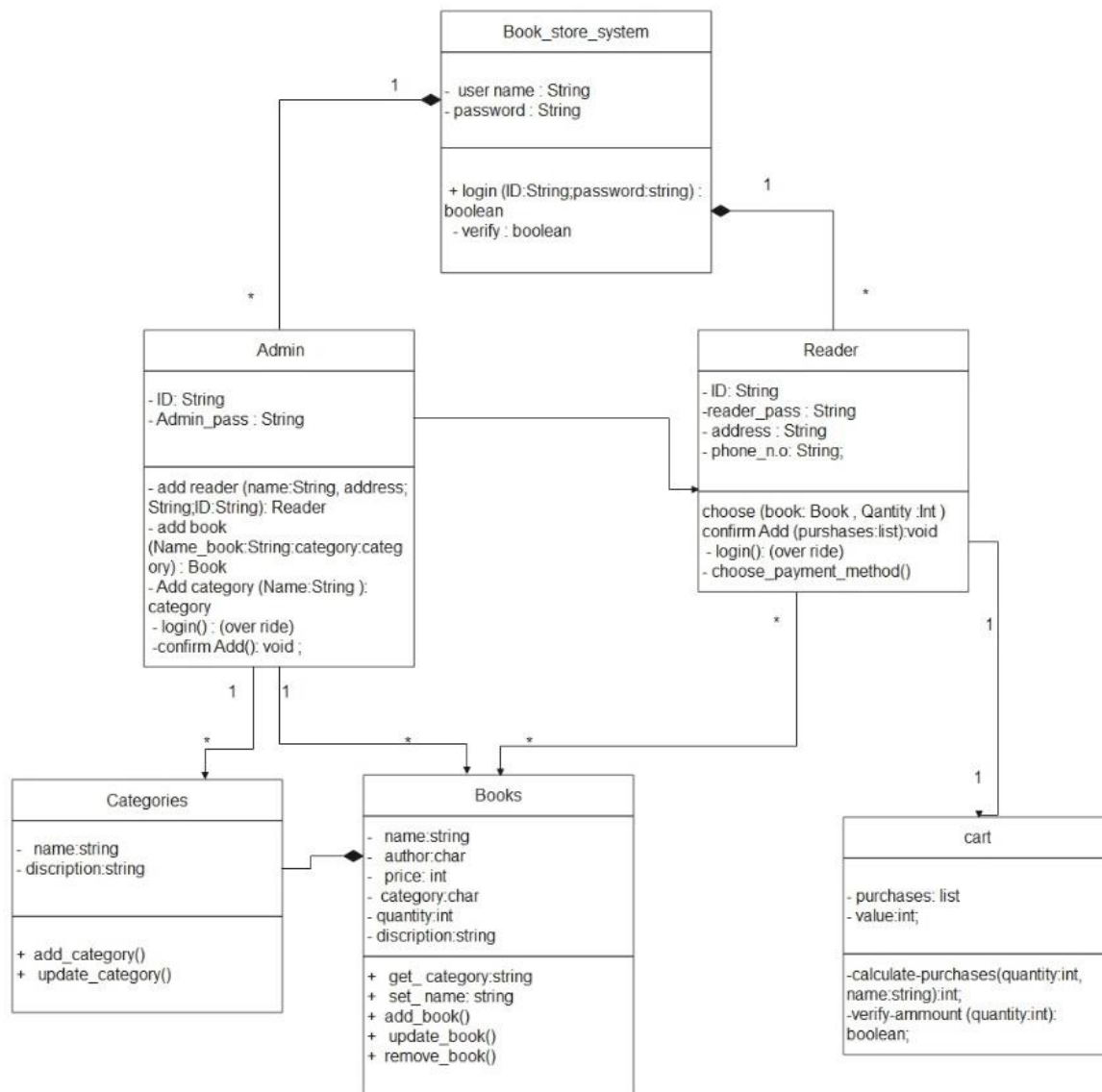


Figure 8: Class Diagram

7.3 OOAD Methodologies:

I. Rumbaugh et. al.'s Object Modeling Technique (OMT):

OMT consists of four phases which can be performed iteratively which makes this methodology flexible to apply:

(a) Analysis phase which consists of three different parts:

1. Object model and the data dictionary:

Object model which represents the objects and the relationships between them

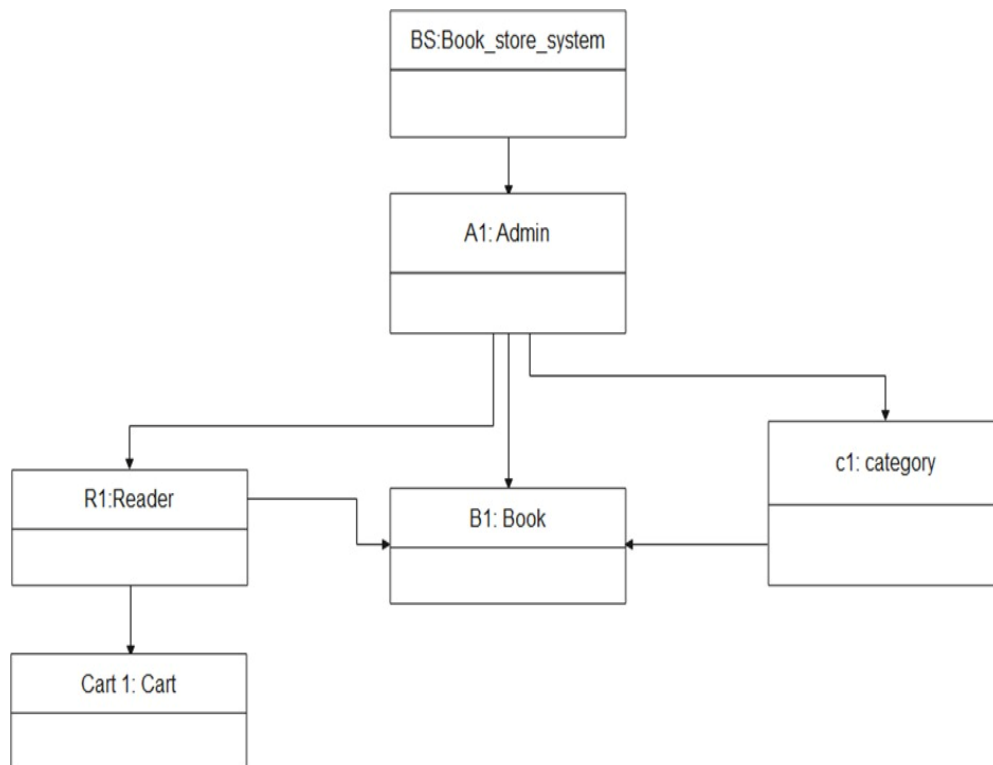


Figure 9: Object Diagram

-Data dictionary which has the description of these relationships:

1. Book store class has reader and admin (composition relationship)
2. Book has a category (composition relationship)
3. Reader choose books (association relationship)
4. Reader add books to cart (association relationship)
5. Admin add books (association relationship)
6. Admin add category (association relationship)
7. Admin delete book (association relationship)
8. Admin update book (association relationship)

2. Dynamic model, presented by the state diagrams .

And is represented by the state diagram which is mentioned before in section (8) of this project where we discussed in detail our state diagram.

3. Functional model which shows how the data is flowing, where its stored and how it is processed by the different processes that constructs our system. And is represented by the Data Flow Diagram.

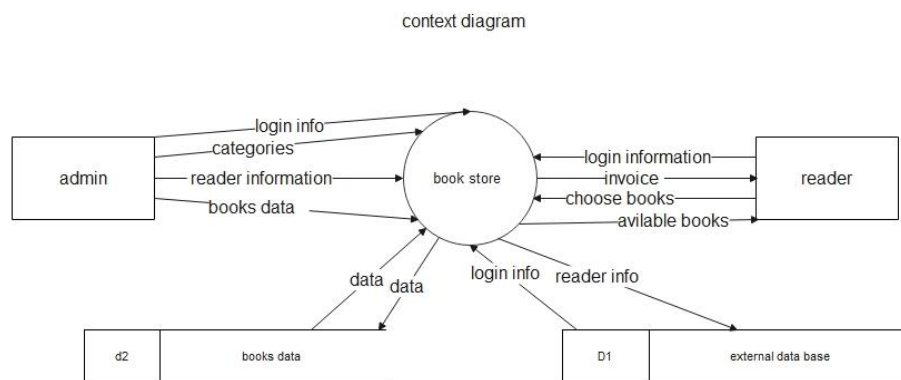


Figure 10:Context Diagram

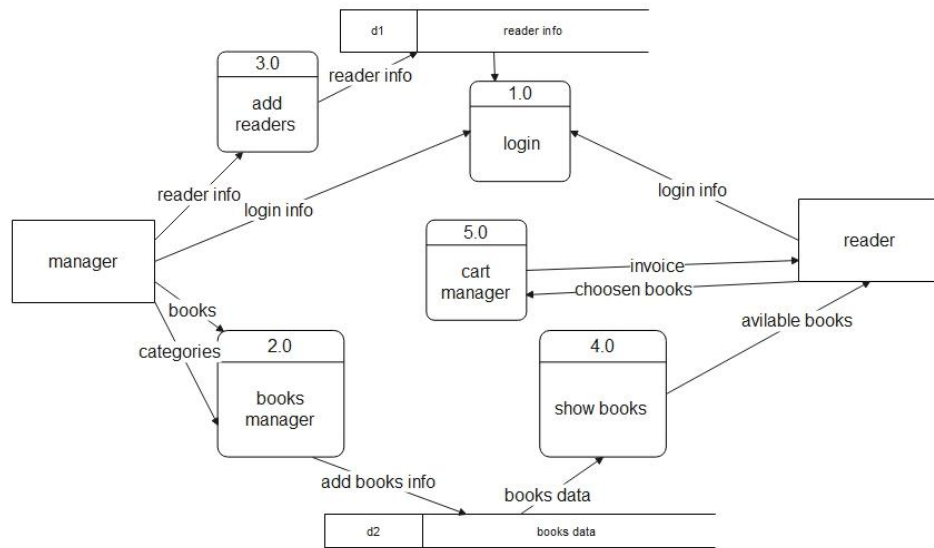


Figure 11: Level 0 DFD

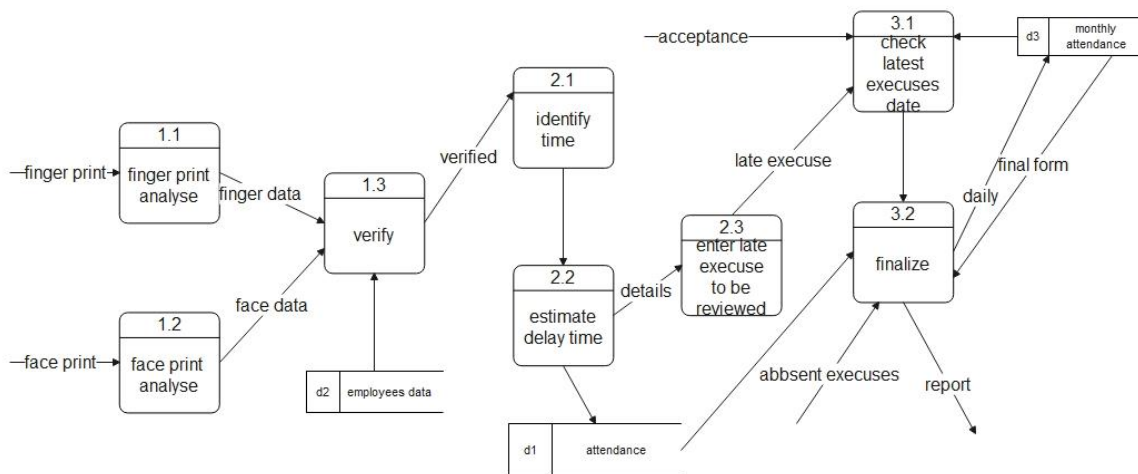


Figure 12:Level 1 DFD

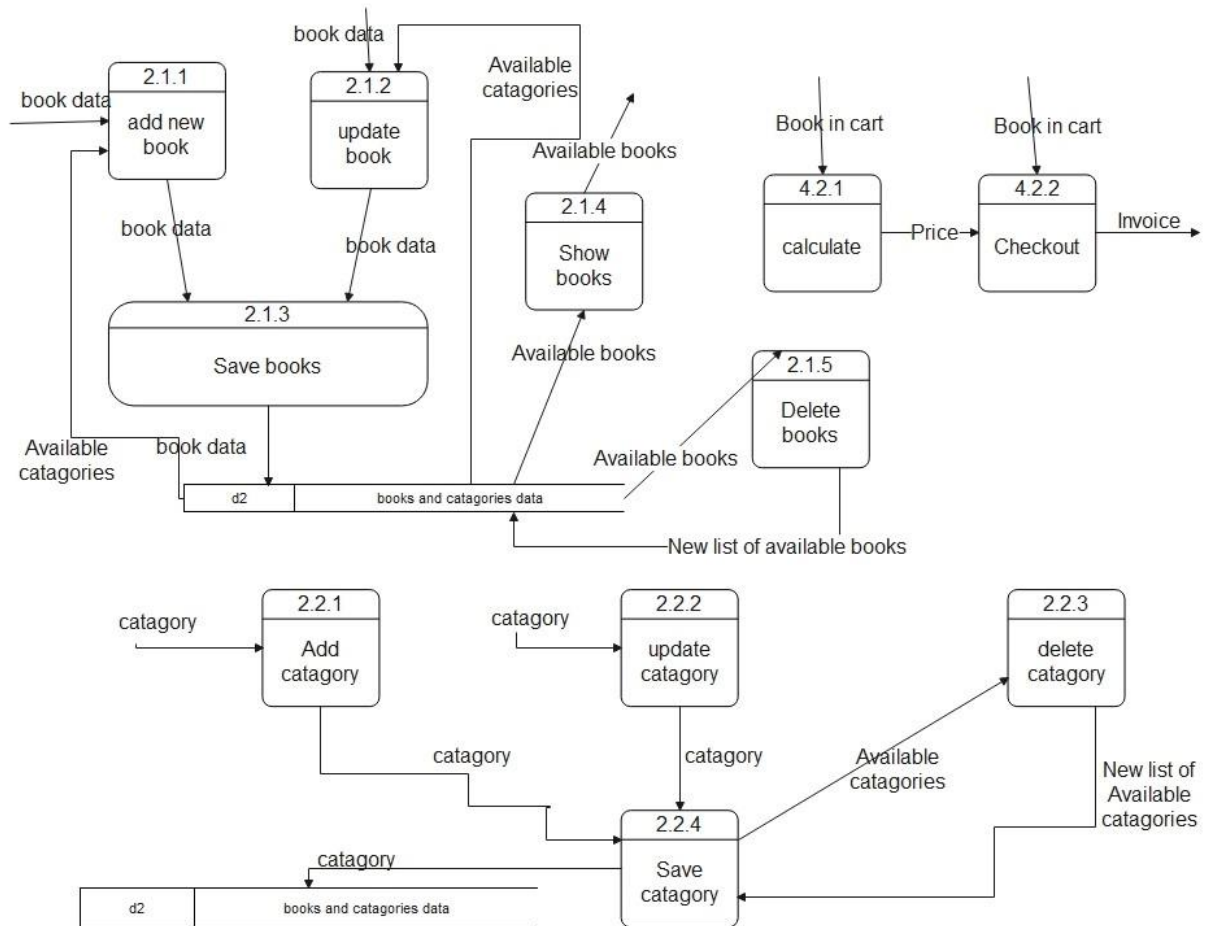


Figure 13:Level 2 DFD

(b) System design

In this phase, we describe the high level architecture of the system by decomposing the system into a set of subsystems or packages as it will be mentioned in section(10) . Of this report.

(c) Object design

This phase produces a design document, consisting of detailed objects and dynamic and functional models. It is concerned with classification of objects into different classes and about attributes and necessary operations needed, which mentioned in the below detailed class diagram (check section (7). Of this report).

(d) Implementation

In this phase, the design is translated into the software. And is presented in a separate file attached with the report containing the source code.

II. The Booch Methodology

This methodology covers both the analysis and the design phases of system development and the methodology is criticized for its large set of symbols.

Booch Methodology tried to avoid the OMT downsides, and that is achieved by setting defined to identify the objects and the methods.

This Methodology consists of two main phases:

(1) Analysis phase which consists of the following diagrams

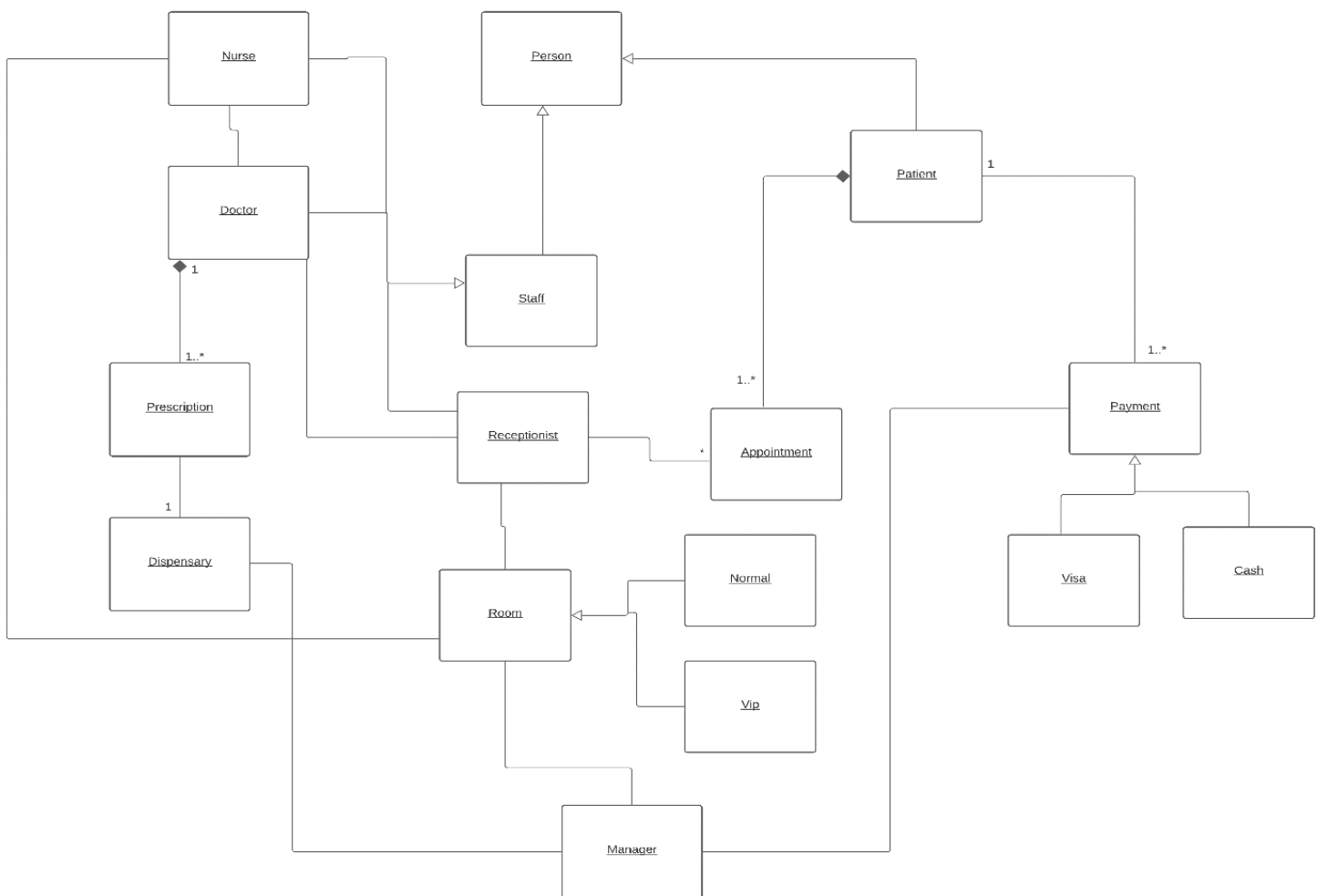


Figure 14: Object Diagram

- Class Diagram: which is identical to the UML Class Diagram which is mentioned in section 7. in this report

- Object Diagram: which shows mainly the life cycle of the object inside our system which the class diagram doesn't show.
- State transition diagrams: which shows the transitions between the system's processes, and these transitions resembles the methods inside every class. This can be shown by a UML state diagram which is mentioned in section 8. Of this report.
- Module diagram: In this diagram: we group the classes which have similar functionalities and then we put them in a module (package/subsystem), and it shows the interactions between the modules. And this can be shown by a State diagram which will be mentioned in section 8. Of this report.
- Process diagram: which shows the interactions of the objects with the system's environment. This can be shown by a UML Use Case Diagram which is mentioned in section 4. In this report.
- Interaction diagram: which shows us the interactions between the system's participants. This actually contributes a great contribution towards identifying the classes in detail as after collecting the methods, we actually can know which method belongs to which class. And this can be shown by either the collaboration or the sequence diagram which is mentioned in section 9. Of this report.

➤ **Design phase which prescribes two parts:**

- A macro development process (top down decomposition)
- A micro development process (bottom up decomposition)

As we look to our system from the whole point of view first then we dive deeply into the details, then we will start with the macro development process.

❖ *The macro development process which consists of the following steps:*

1. Conceptualization: the main goal(s) of the system: reader can choose different books and add them to the cart ,then checkout his order.
2. Analysis and development of the model.
3. Design or create the system architecture which will be discussed in section 10. Of this report.

4. Evolution or implementation

5. Maintenance: we keep supporting the system via maintenance and regular updates.

❖ *The micro development process which consists of the following steps:*

The micro development mainly aims for the details of the design and the details of the classes in our system, so it consists of the following: -

1. Identify classes and objects: which is mentioned in section 7. Of this report.
2. Identify class and object semantics (translation of the object in the real world).
3. Identify class and object relationships Both points 2,3 are mentioned in OMT analysis phase and specifically shown via the Object Model
4. Identify class and object interfaces and implementation. And the classes' interfaces are shown in the component diagram which is mentioned in sections 7. and 11. In this report.

7.4 Comparative Analysis of the Output of the Adopted Methodologies:

1st Methodology: (Object Modeling Technique O.M.T)

- It offers ways for analyzing, designing, and implementing a system using object-oriented programming through a number of approaches, beginning with the concept and ending with the completed application. We will define these methodologies and demonstrate their strengths and weaknesses.
- It comprises mostly of four phases, each of which is applied iteratively to provide the model strength. Analysis is concerned with identifying the objects and their relationships (not with their qualities or techniques). Concerned with the items and their relationships first is defining a goal and problem statement, which will be followed by dynamic and functional modelling. Identifying the object is done subjectively depending on the analyst's point of view and the aim and domain of the problem (by gathering information from the user).
- The cloud model was used in this manner (cloud and he puts the objects in this cloud and word class and object were interchangeable).

- It also employs a dynamic and functional model to determine the attributes and methods for these objects (using a variety of strategies to determine the methods and attributes of objects or classes), bearing in mind that this is an iterative process.
- System design: determine system architecture and data storage for the system, which means breaking the system down into packages and subsystems and determining the relationships between them, as well as which subsystem the object will be in.
- Object design: if you have complete details of each object, and in OMT, he interchangeably uses the terms class and object. This phase creates a thorough object, dynamic, and functional model (detailed class diagram if we're talking about UML), and it's responsible for locating specific characteristics and functions for each class (object). Classification of objects into various classes, with each class having its own set of attributes and methods Implementation: convert the design into software code, which some argue is not a part of the design or analysis, but those who use this method respond that they are talking about modelling that can be used from the project concept to the final product.
- The analysis is divided into three parts, which are as follows:
- Object model (class diagram without attributes or methods, just classes and their relationships) and data dictionary are presented (which shows the details of the relationships of the classes and the details of each class and what is this class in the real world)
- dynamic model: which is similar to a UML state diagram (drawing the main object of the system).
- functional model: also known as the DFD model, focuses on how data flows, where it is stored, and how it is used by various processes. Based on this, you can figure out how objects interact with the file system or data base.

➤ **OMT Advantages:**

It is a simple methodology which is easily to understand.

➤ **Disadvantages:**

- It is not an accurate methodology
- It has a very high number of iterations
- It has high mistake percentage

2nd Methodology: (Booch methodology)

- Booch methodology: includes analysis and design while attempting to avoid the problems of OMT method.

Booch sometimes is criticized for his large set of symbols.

- **The Booch method consists of the following diagrams:**

- *Class diagrams*
- *Object diagrams*
- *State transition diagrams*
- *Module diagrams*
- *Process diagrams*
- *Interaction diagrams*

- **Booch has divided the design phase into two phases which are:**

1. Macro development
2. Micro development

- **The macro development process consists of the following steps:**

1. Conceptualization
2. Analysis and development of the model
3. Design or create the system architecture
4. Evolution or implementation
5. Maintenance

- **The micro development process consists of the following steps:**

1. Identify classes and objects
2. Identify class and object semantics
3. Identify class and object relationships
4. Identify class and object interfaces and implementation

8. State Diagram

the state diagram describes the behavior of the classes when an external stimulus interacts with them, usually the state diagram defines what the system does when a single object responds to a number of events that happen in the system

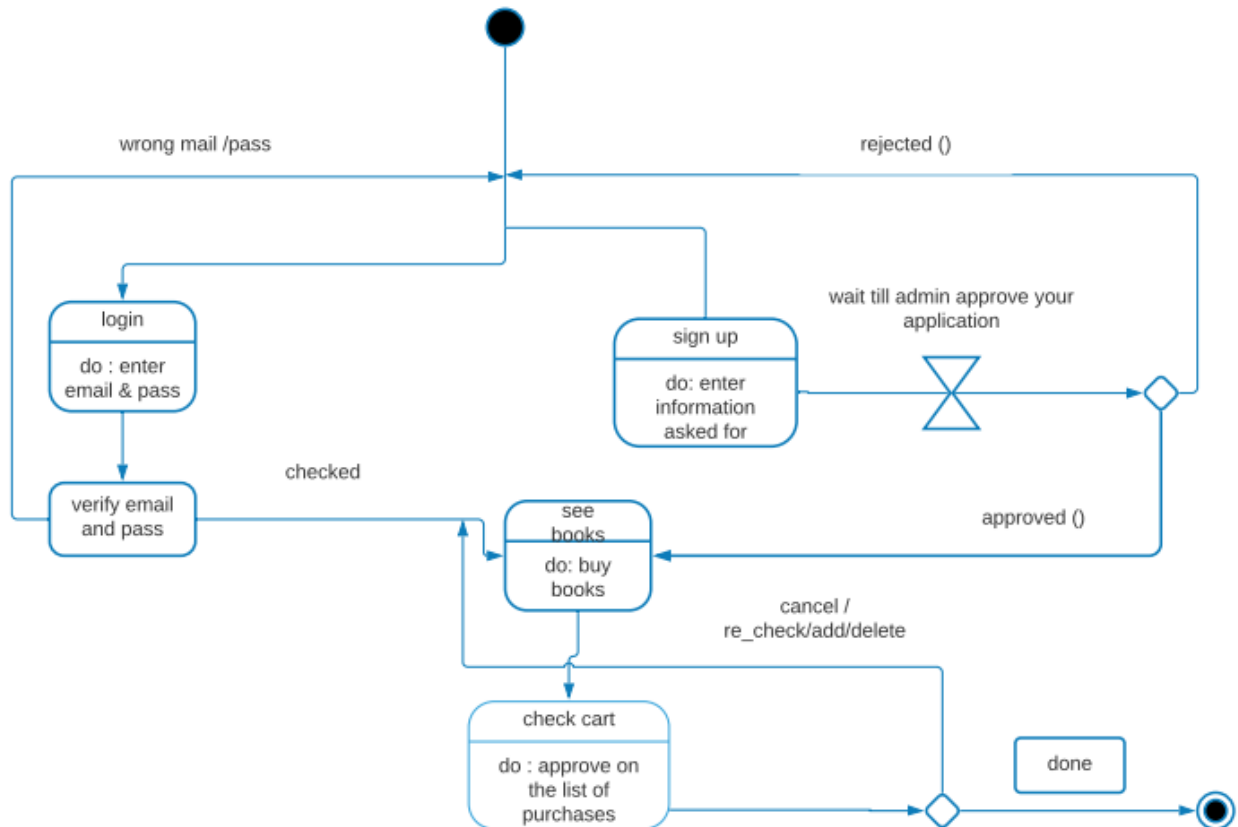


Figure 15: State diagram from readers view

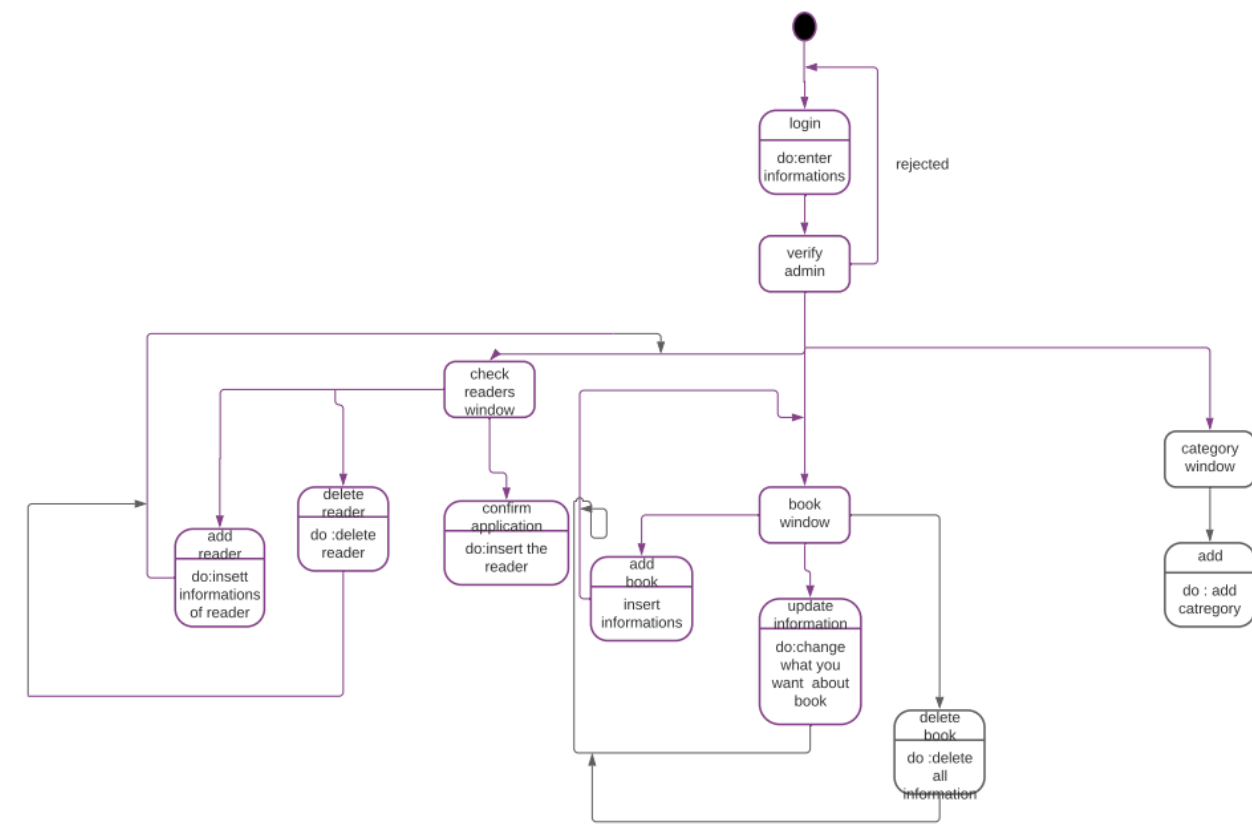
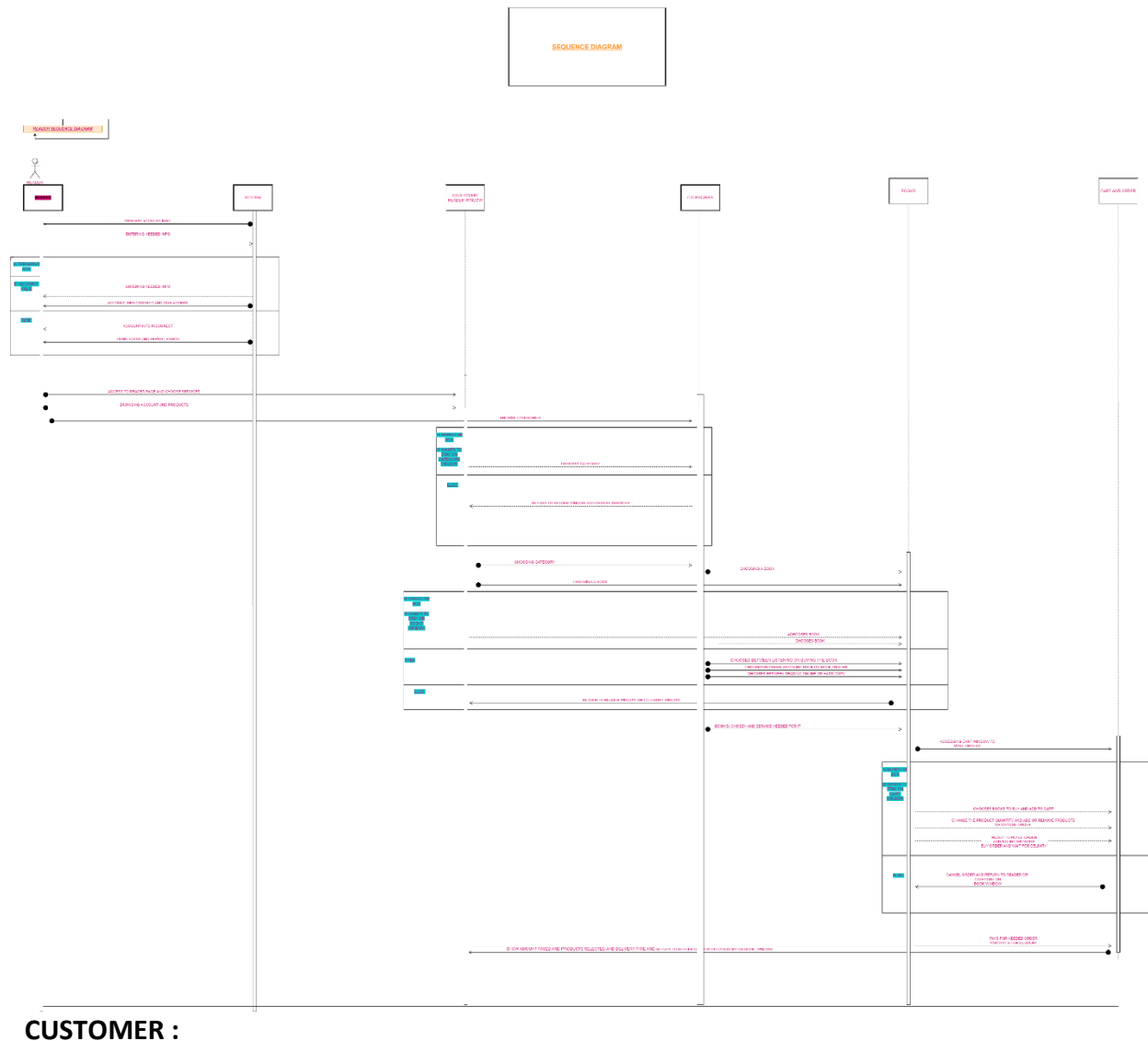


Figure 16: State Diagram #2

9. Interaction Diagram



SEQUENCE DIAGRAM STEPS AND DESCRIPTION :

- the reader (customer) should enter needed info to validate his account
- the system validates account and gives him the services
- the reader can browse into his page and choose category , book type , buy book and check his chart and the amount he will pay
- he can edit his chart before making order

- after the order confirmation he waits for delivery

ALTERNATIVE BOX 1 :

- the account info is asked from reader
- if they are correct than he can go through and use his services
- else his access is denied and an error is reported

ALTERNATIVE BOX 2:

- reader can get into categories page to choose one
- else leave it

ALTERNATIVE BOX 3:

- reader can choose a book
- choose its type (audio,readable,both)
- watch a trailer for the book
- choose between hard and soft copy for the book
- or he can return to category or reader page

ALTERNATIVE BOX 4:

- choosing the book to buy and adding it to cart
- change product (add more of the same product or just change the product)
- when ready choose to pay and wait for delivery (within choosing a payment method)
- else reader can cancel the order and return to category or reader page

SEQUENCE DIAGRAM



Figure 18: Sequence diagram (Admin)

ADMIN :

SEQUENCE DIAGRAM STEPS AND DESCRIPTION :

- The admin inserts his info to the system *GUI*
- He gets the validation from the system
- The admin can go through his services
- Then he enters to change categories or edit them or to check books left or check prices
- He confirms any changes done
- He can check if a customer password is valid and change it if needed
- The system returns the updates after his confirmation
- He can help change or update customer chart
- the admin can stop or denie customer services
- The system updates the confirmations
- The system reports any errors and updates for needed maintenance to the admin
- The admin can delete or update account
- The admin confirms last changes on customer account to the system so it can know if its ready for services or not

ALTERNATIVE BOX 1 :

- System assures that account is valid
- Requests the pin
- If the pin is right then the account is valid
- The admin can access his window and start using the services
- If the account pin is not valid the system reports an error

ALTERNATIVE BOX 2 :

- If The admin makes some changes he can submit it to the manager
- Then the system is updated
- And the admin can go on with services
- He can check again or just save changes

- Else of all this he can decline the changes and return to admin window

ALTERNATIVE BOX 3 :

- if the account is valid then the edits needed to be done by admin are done (or no need for admin to edit a customer information now)
- If the account of the customer is not valid it is reported to the system and admin
- Checking on password to see if it is the problem or not
- If the password is the problem then admin can help you with the password the right one or the resetting of it
- Now the account should be fixed and valid
- System is updated
- Any other help or services needed can be done
- If not working after all then the system reports an error and checks if account even exists

10.Architectural Model:

10.1 System Architecture

We will make the system architecture as that the Book store system will be divided into sub-systems, making for each of them a system architecture.

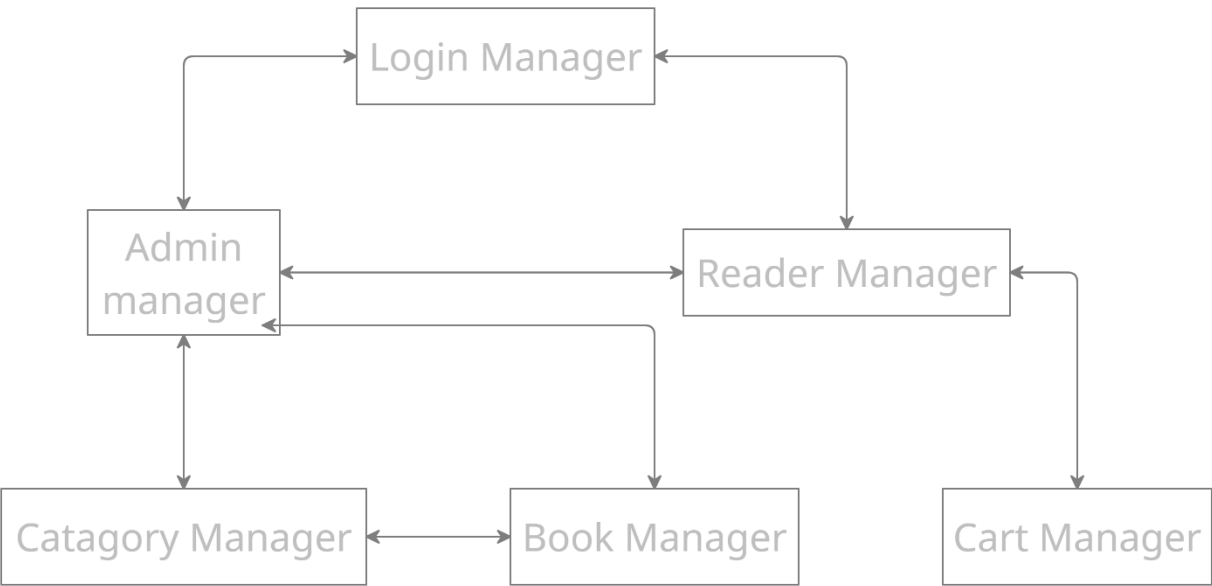


Figure 20: Layered Architecture

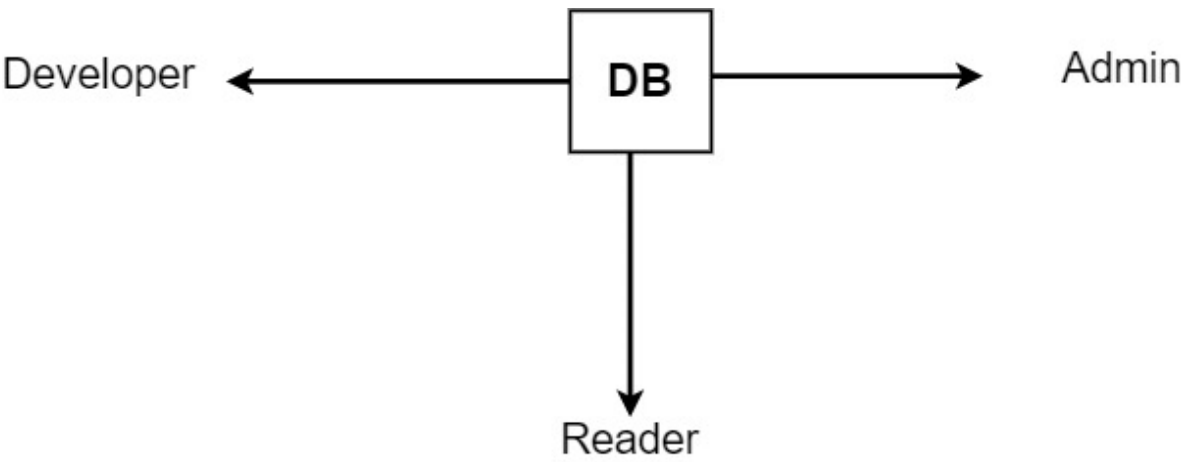


Figure 19: Black Board

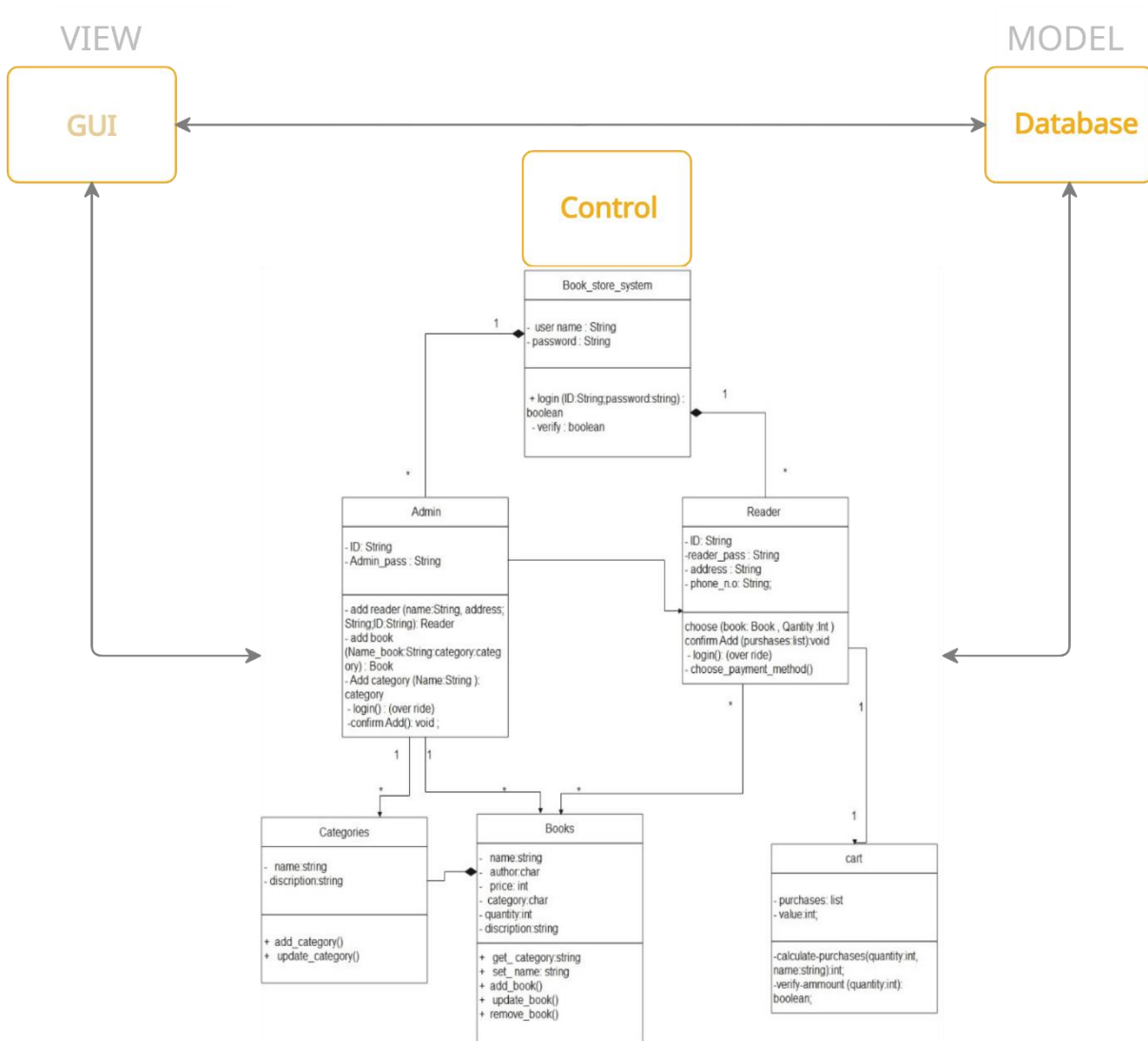


Figure 21: MVC Architecture model

11.Component Diagram:

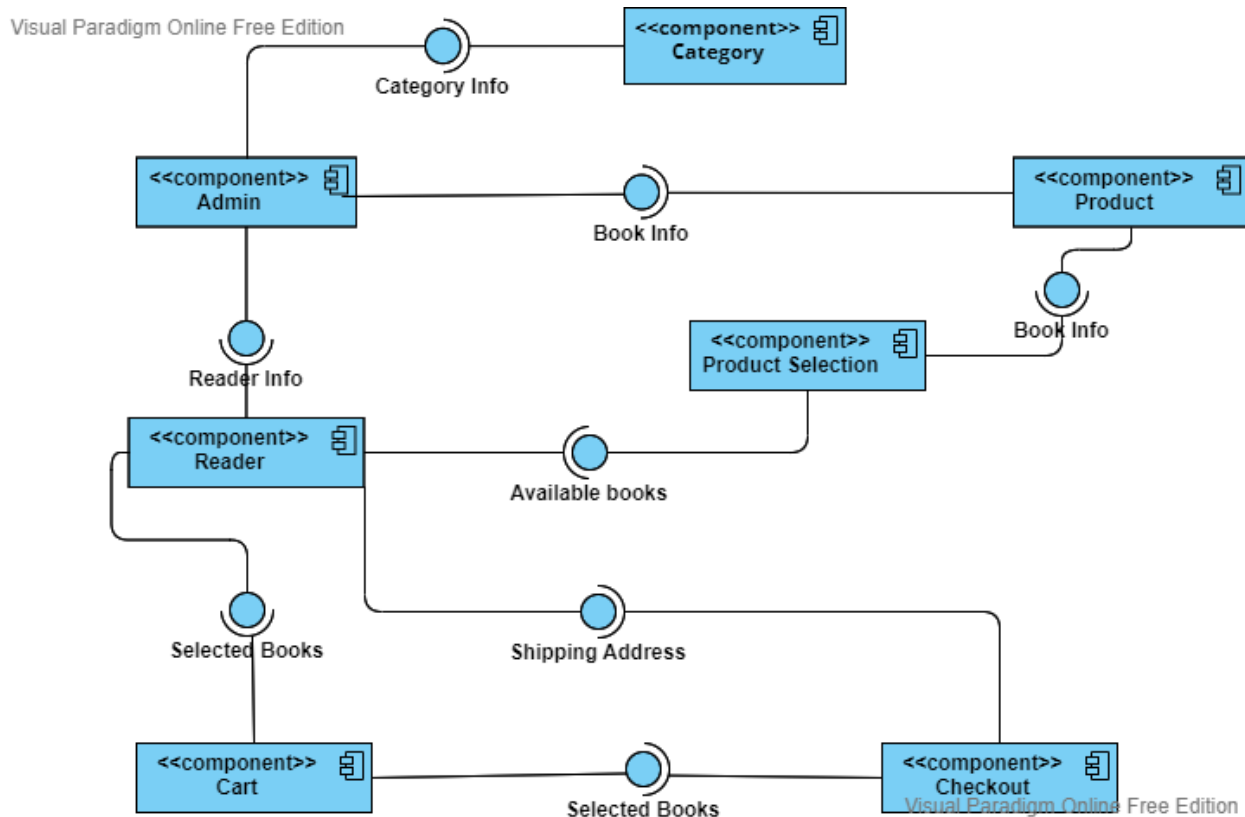


Figure 22: Component Diagram

Component Diagram Description:

Admin:

The admin component is responsible for giving the admin access to control many things in the system. Such as, adding and updating categories, books and readers to the system through the manager window.

It provides Category information to category component, provides book information to product component and provides reader information to reader component. All these information are entered by the admin.

1. Category:

The category component manages category names and descriptions. It requires its information from admin component in order to map categories to books.

2. Product:

The product component manages product's name, authors' name, price and category of the book. This information is provided from the admin.

Product component also provides the books and their information to product selection component so that the reader afterwards chooses from them to buy in cart.

3. Product Selection:

This component provides which are the available books to buy to the reader component.

4. Reader:

After this component has reader info from admin and available books from product selection, in the reader window by pressing "buy book" it gives access to reader to select from list of available books to buy from. These selected books then is provided to cart component buy clicking "add cart" button and reader's address is provided to checkout when pressing "product to checkout" button.

5. Cart:

By pressing "view cart" from reader window, cart window is opened showing selected books their quantity, descriptions and total price, also you can remove any from selected books.

By pressing "product to checkout" button selected books are provided to checkout component.

6. Checkout:

By pressing "product to checkout" button cart checkout windows is opened showing reader's shipping address, selected books their quantity, descriptions and total price, also you can cancel checkout.

At last press "Confirm order" button to checkout.

12. Testing:

12.1 Testing Strategy

12.1.1 Introduction

- ❖ Test process and test cases play a vital role in ensuring the accuracy of the program. It serves for quality control, verification, and validation purposes. Studies show that testing is given 50 per cent of the cost of software development. The testing method is essentially an examination performed to provide product or service quality information. It can also provide an unbiased view of the program so that the user can understand the system's danger. We use this approach to identify software glitches or flaws in the hospital administration system. It means doctors can access their I'd anytime they need it and patients can search or update their care information whenever they want. In hospital management system, unit monitoring, integration testing, white and black box testing, etc. are used.
- ❖ A test case is a series of conditions under which a tester must assess whether or not device or software features are working well. Test cases are a time-consuming process. Automation is, therefore, a significant issue. Two main approaches to test cases are available-

Generating test cases from requirements and design specification.

Using source code.

In order to fully test that all the requirement of a system is met, there must be at least two test cases that is one positive test and one negative test. We can define test cases as-

Formal test cases-- In the formal test case, the test case is characterized by a known input and the predicted output that is processed before the test is carried out. Formal testing is officially conducted by research team.

Informal test cases- In informal test cases, testing is performed by a coder before a test team is given script. In informal research no protocol is applied.

12.1.2 Test Procedure

Check technique assists in maintaining the accuracy of the program. A quality assurance, verification, and validation are the purpose of the test procedure. The testing method is essentially an examination performed to provide product or service quality information. Testing is the method where a program is executed with the intention of discovering errors. It is used to reflect the ultimate specification, design, and code review. Specific test conditions should be carefully tested, and the found bugs should be corrected. Patient management program monitoring is followed as:

1) Unit testing

Unit testing plays a key role in the early phase of life cycle software development that assists in identifying bugs; and once properly designed and implemented, it decreases quality costs. However, if unit tests are not well written, or if they are poorly performed, bugs move through the development process. This negligence can cost a company thousand, if not hundreds of thousands of dollars, depending on the size of the project. Unit testing involves extensive coding experience with multiple viewpoints such as System coverage, Loop coverage, Branch coverage, Condition coverage, and Fuzzing system to predict the expected failure, pre-defined negative testing when creating code.

Testing Goal - Dots squares testing service unit aim is to find as many bugs as possible early in production. This is accomplished by writing accurate and reliable unit tests, while providing comprehensive documentation for the phase of creation. This ensures that the development team can re-facture bugs and inconsistencies which we find early in the development stage. We work closely with your team to ensure that project targets are achieved within budget and on schedule

Benefits of Unit Testing

- ❖ **Finding problems early** -Unit tests find problems early in the development cycle
- ❖ **Facilitate change** - Allows the programmer to re-factor code during the development process and at a later date, while ensuring the software is still running properly
- ❖ **Simplify integration** -By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.
- ❖ **Documentation** - Developers looking to get a clearer understanding of the outcome of a unit test should look at the unit test report

- ❖ **Design-** During the project life-cycle, each unit test can be seen as a design feature that defines classes, methods and measurable behaviors.
- ❖ **Integration testing** - After unit testing we usually do Integration. Once we have developed and checked all the individual components, we start to combine those "Unit Checked" modules and begin the integrated testing. The definition of integration testing is therefore relatively straight forward-Integrate / combine the element evaluated by the user one by one and test the actions as a unit.
- ❖ **Testing goal** - Checking the interfaces between units / modules is the key feature or aim of Integration testing. Second, the individual modules are checked in isolation. If the modules are checked unit, they are integrated one by one until all the modules are incorporated to verify the combination actions and confirm whether the specifications are correctly implemented or not.
- ❖ Here we should understand that integration testing does not take place at the end of the process but is carried out at the same time as the development. And most of the time, all the modules are not currently available for testing and here is what the challenge comes up to testing something that doesn't exist.
- ❖ **There are two main approaches to generate test cases: -**
 - Bottom-up approach
 - Top-down approach.

2) Black box testing

Black Box testing is a software development methodology in which the software under test (SUT) functionality is checked without looking at the internal code structure, specifics of implementation and knowledge of the software 's internal paths. This method of testing is solely based on the needs and requirements of the program.

Black Box Testing-Steps

The standardized steps followed to conduct some kind of Black Box Testing are here.

- Parameters and device specifications are initially investigated.
- Tester selects appropriate inputs (positive test scenario) to verify that SUT correctly processes them. Also, some incorrect inputs (negative test scenario) are selected to check that they can be identified by the SUT.
- The check defines estimated outputs for all of those inputs.
- Software tester builds test cases using the inputs picked.
- Conduct test cases.
- Tester software compares the real performance to the predicted outputs.
- Defects are patched and re-tested if any.

Types of Black Box Testing

- There are many types of Black Box Testing but following are the prominent ones.
- Functional testing - This black box testing type is related to functional requirements of a system; it is done by software testers.
- Non-functional testing - This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
- Regression testing - Regression testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

13. Estimated project cost

non functional	Number
Back _up and recovery	0
Data communication	3
Distributed processing functions	4
performance critical	0
Existing operating environment	2
Online data entry	3
Input transaction Built over multiple screens	5
Master files updated online	3
Complexity of inputs , outputs , files ,inquiries	2
Complexity processing	0
Code design for reuse	2
Are conversions/ insulation included in design	1
Multiple insulations	0
application designed to facilitate change by the user	3
Total f ($\sum_{i=1}^{14} f$)	28

functional	input	output	inquiry	files	interface
Examples	Add reader info	Show cart	Show cart	Reader data base	Admin adding reader
	Add books info	Check out	book description in reader window	Books	
	Add category				
number	3	2	2	2	1
Weighting factor	Avg	low	Avg	low	Avg
Weighting cost	4	4	4	7	7
total	12	8	8	14	7
Function points	49				

- $FP = UFC * [0.65 + 0.01 * \sum_{i=1}^{14} f_i]$

- $FP = 49 * [0.65 + 0.01 * 28] = 45.57$

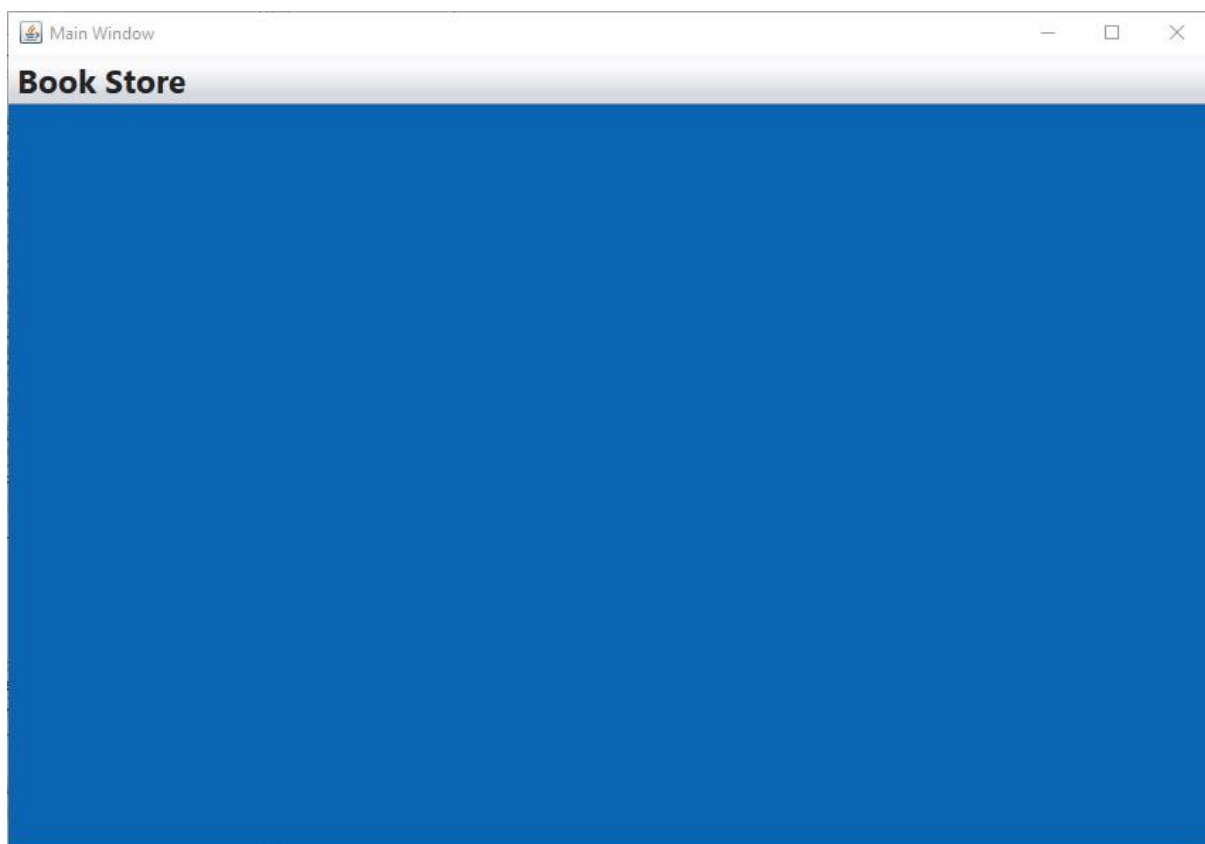
1. Productivity of one man is ->5
2. Men needed is 9 to 10 men, men existing are 3 men
3. Salary for each is 3200 so total salaries is 32000 + 10000(bonuses)
4. Duration 2 to 3 months within hiring one or more and working for 4 days per week
5. Servers would cost 50.000 L.E.
6. Frame works, tools ...etc would cost 20.000 L.E.
7. Maintenance per year would cost 25.000 L.E.
8. Rents, services, food ... etc would cost 15.000 L.E.

9. Total estimated cost =377000 i.e., for a good and high scale framework for an online and offline bookstore with
10. Closely no risks and long living up to date and keeping up with the ages.

14. User Guide:

➤ 14.1 Main Window

Has a combo box so that the user can choose to log in as admin or reader or to exist from the app.



➤ 14.1.1 Login Window

There are two buttons to log in as reader or to login in as admin, the admin should add the reader information then give him a username and password so he can be able to log in.



The image shows a screenshot of a web application window titled "Login". The window has a blue header bar with the text "Login Window :". Below the header, there are two tabs: "Admin" and "Reader". The "Admin" tab is currently selected. The main content area is light gray and contains two input fields. The first field is labeled "Username :" and has a placeholder text "Enter Username". The second field is labeled "Password :" and has a placeholder text "*****". Below the input fields, there are two buttons: "Login" and "Cancel".

➤ 14.1.2 Admin Window

After admin log in successfully, he will view admin screen, then he can be able to add new reader, add new book, add new category or update their old information.

The screenshot displays a web application window titled "Manager Window". It features a blue header bar with the text "Manager Window :". Below the header, there are three tabs: "Reader", "Book", and "Category", with "Reader" being the active tab. The main content area is divided into two sections. On the left, under the heading "Customar List :", there is a large, empty white rectangular box. On the right, there is a form with five input fields, each preceded by a label: "Reader ID :", "Reader Name :", "Address :", "User Name :", and "Password :". Below these fields, there are four buttons: "New Reader", "Insert", "Update Reader", and "Confirm Added".

14.2 End-User Guide

Admin Guide

How admin can login.



The image shows a screenshot of a web-based login window. The window has a title bar with a small icon and the text "Login". Below the title bar is a blue header area with the text "Login Window :". Underneath the header is a light gray bar containing two tabs: "Admin" and "Reader". The "Admin" tab is currently selected. The main area of the window is light gray and contains the following elements:

- A label "Username :" in blue text, followed by a text input field containing the placeholder text "Enter Username".
- A label "Password :" in blue text, followed by a password input field containing eight asterisks "*****".
- Two buttons at the bottom: "Login" and "Cancel", both in blue text.

How admin can add new reader information his (id,name,address,username,password)

Then press confirmed added button then insert, also updating old reader information.

The screenshot shows a window titled "Manager Window" with a blue header bar containing the text "Manager Window :". Below the header is a tabbed interface with three tabs: "Reader", "Book", and "Category". The "Reader" tab is selected. The main content area is titled "Customer List :" and features a large empty rectangular box on the left. To the right of this box are five input fields, each with a label: "Reader ID :", "Reader Name :", "Address :", "User Name :", and "Password :". Below these input fields are four buttons: "New Reader", "Insert", "Update Reader", and "Confirm Added".

Manager Window :

Reader Book Category

Customer List :

Reader ID :

Reader Name :

Address :

User Name :

Password :

New Reader Insert Update Reader

Confirm Added

Manager Window

Manager Window :

ReaderBookCategory

Customar List :

basel (Is Insert ? true)

Reader ID :1

Reader Name :basel

Address :nasr

User Name :basel

Password :20

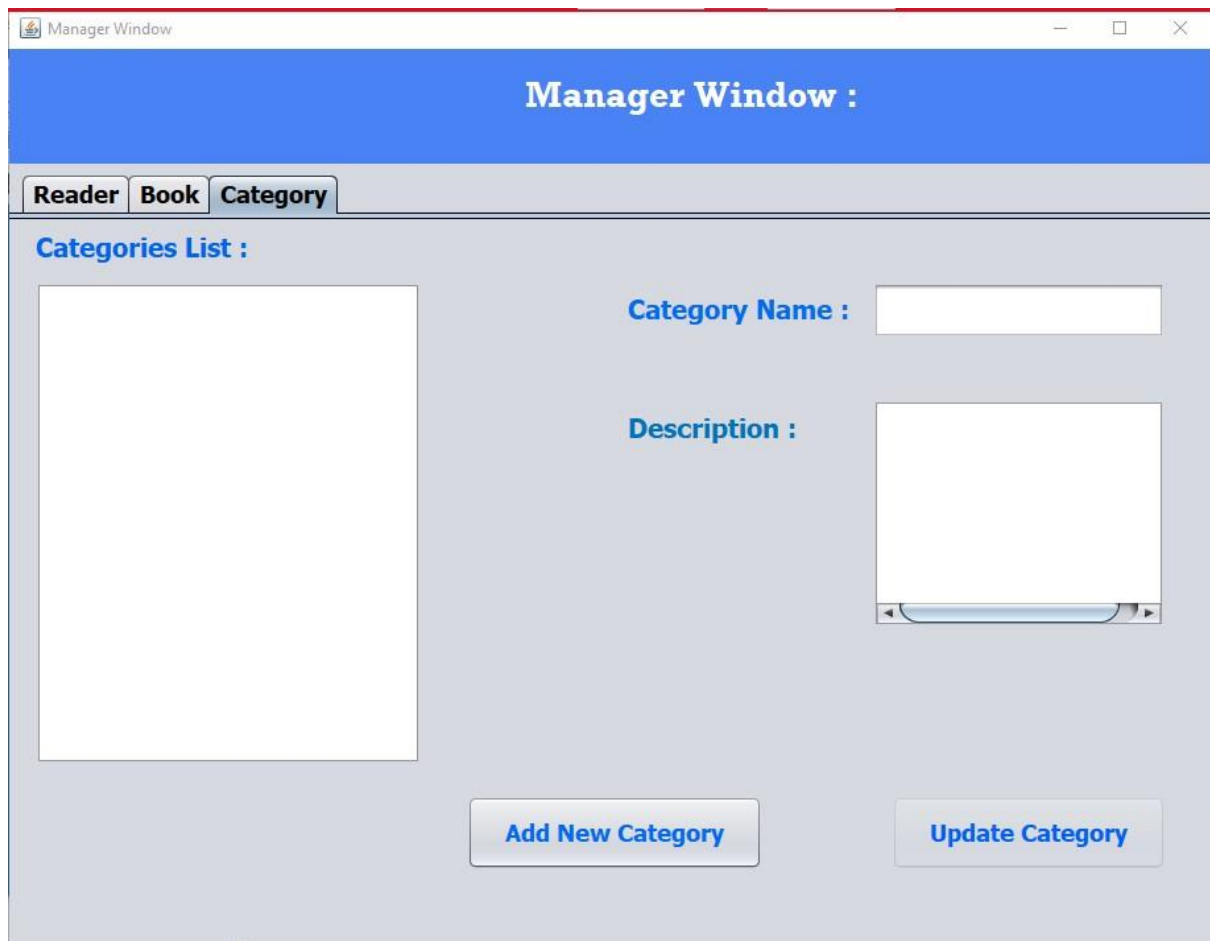
New Reader

Insert

Update Reader

Confirm Added

How admin can add category and its description and also updating old categories.



The image shows a software window titled "Manager Window" with a blue header bar. Below the header, there are three tabs: "Reader", "Book", and "Category", with "Category" being the active tab. The main content area is divided into two sections. On the left, under the heading "Categories List :", there is a large, empty rectangular box intended for displaying a list of categories. On the right, there are two input fields. The first is labeled "Category Name :" and is a single-line text box. The second is labeled "Description :" and is a multi-line text area with a visible scrollbar. At the bottom of the window, there are two buttons: "Add New Category" and "Update Category".

Manager Window

Manager Window :

Reader Book **Category**

Categories List :

Category Name :

Description :

Add New Category Update Category

How admin add books information its (name, author, price, category, available quantity, description) , and also updating and deleting old books information.

Manager Window

Manager Window :

Reader

Book

Category

Books List :

Book Name :

Author :

Price :

category :

Available Quantity :

Description :

New Book

Delete

Update Book



Add New Product



Add New Book :

Book Name :

Little Womwn

Author :

Loiuse May

Price :

50

Quantity:

10

Decscription :

Department :

Add New Product

Confirm Add

1.1.1 Reader Guide

Reader chooses the book and press the button buy book.

Customar Window

Reader Window :

[View Cart](#) [Buy Book](#)

- ▼ **Book Store**
 - ▼ **stories**
 - Little Women**

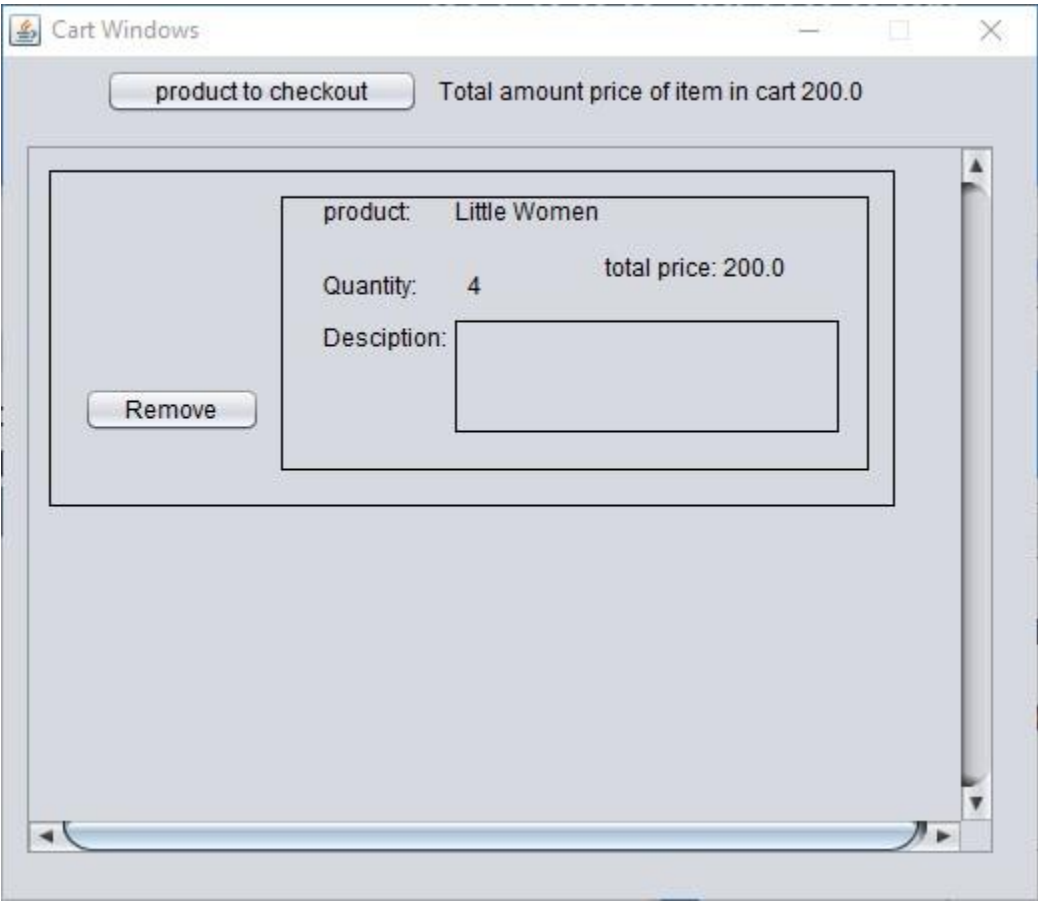
Quantity : 4 ▼ [Add Cart](#)

Book Name : Little Women

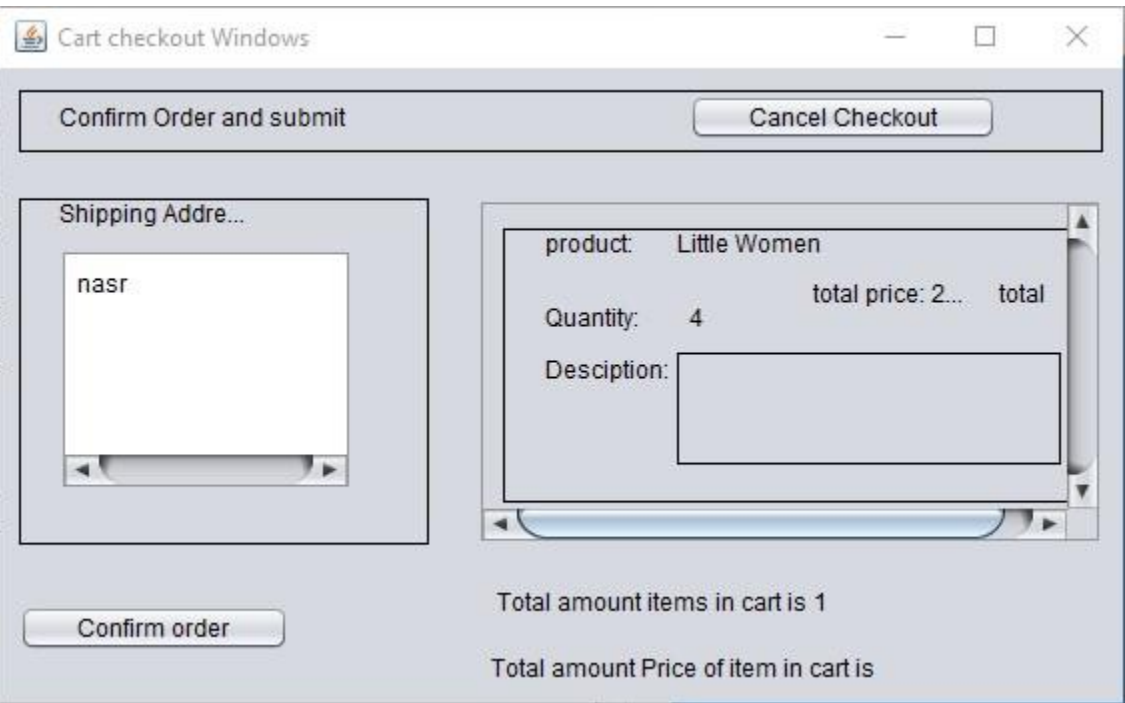
Price : 50.0

Description :

Reader then presses view cart button to show what he buys.



Reader then presses product checkout button to add his shipping address and confirmed the order or cancel it.



➤ 14.3 Reader Window

After reader log in successfully, he will view reader window , the he can be able to buy different books and choose the quantity he wants then, add them to cart by pressing the add cart button , after buying he can view the cart by pressing view cart button

The screenshot shows a web application window titled "Reader Window :". The interface is divided into several sections. At the top, there is a blue header bar with the title. Below the header, there is a light gray bar containing two buttons: "View Cart" and "Buy Book". The main content area is split into two columns. The left column is a large, empty white space. The right column has a blue header and contains a form with the following elements: a "Quantity :" label followed by a dropdown menu showing the value "1"; an "Add Cart" button; a "Book Name :" label; a "Price :" label; and a "Description :" label followed by a text input field with a vertical scrollbar on the right side.

Reader Window :

View Cart Buy Book

Quantity : 1 Add Cart

Book Name :

Price :

Description :

15. Time Plan:



Figure 23: Time Plan

16. Conclusion:

We implemented our software in way such that all the user requirements are fulfilled to satisfy the client of this software and we didn't stop at that, we added our own functionalities in order to make the software logical and similar to an actual bookstore's software.

We have also implemented all the UML diagrams that are required to help in the development of such a software and act as reference for the client if he ever asked for proof of work update.