**EXPERIMENT 4:**

**CASE STUDY: CUSTOMER SUPPORTING SYSTEM (OnlineBook store)**

**Aim:** Identify and analyze events

**Procedure:**

Generally event is specification of a significances occurrence that has a location in time & space .The Flow of events of a usecase contains the most important information derived from usecase modeling work. It should describe the usecases flow of events clearly enough for an outsider to easily understand it. Remember flow of events should present what the system does, not how the system is design to perform the required behavior.

Guidelines for the contents of the flow of events are,

1) Describe how the usecase starts and ends.

2) Describe what data is exchanged between the actor and the usecase

3) Do not describe the details of the user interface, unless it is necessary to understand the behavior of the system.

**b) Aim: Identify use cases:**

Use case description for online book store: in these we have

**Login:** here we will our credentials and login to the online book store.

**Browser catalog**: here user can browse his required book according to his need such as amount, author, version etc.….

**Order book**: after selecting the book can order the book on checking the option “buy now’.

**Manage account**: here we will have the option of online payment through our account were it will manage our account details like sending OTPS checking balance ect.

**Ship book:** after the payment of money to the book, the book is shipped to our address in various steps.This is managed by the system.

**Manage catalog**: in this the admin will check and analyze the details of shipping books.

**Login**

**Search book**

**Browse catalog**

**Order book**

**Manage account**

**Ship book**

**Manage catalog**

**c) Aim:Draw Event table**

**Event table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Event | Actor | Description | Object |
| Login | Customer | Here we will open the web page and by using our credentials we can login to the website of the online book store. | login |
| Search book | Customer | In search books the users will search books according to their requirements | book |
| Browser catalog | customer | The user can browser his required book according to their needs such as author, amount ,version | book |
| Order book | Customer | After selecting the book the customer can order the book on clicking the commands such as “buy new”(or)”add to cart” | book |
| Manage account | customer | Here we will have the option of cash on delivery (or) online payment .if we choose online payment through our account ,it will manage our account details like sending OTP,checking balance etc. | account |
| Ship book | system | After the payment of money to the book which the customer had selected ,the book is shipped to the customer address in various steps. This is managed by system | books |
| Manage catalog | Administrator | In managing catalog the administrator will check and analyse the details of shipping books like weather the correct book is being delivered (or)not | books |

**d) Aim:** Identify and analyze domain classes:

**Customer address orderform**

|  |
| --- |
| customer |
| name:string  id:string  password:int  address:string |
| Orderbooks()  Searchtitle()  Fill the order form() |

|  |
| --- |
| address |
| Street:string  City:string  State:string  Country:string  Pincode:int |
|  |

|  |
| --- |
| order form |
| Customer: string  Tax:int  Paymentmode:int  Deliveryfee:int  Totalbill:int  Dateoforder:int |
|  |

**Login logout Seller**

|  |
| --- |
| Login |
| Customerid:string  Id:address |
|  |

|  |
| --- |
| Logout |
| confirm |
|  |

|  |
| --- |
| Seller |
| Name: string  Address: string  Url:int |
| Check availability()  Confirm()  Give catalog()  Send details()  Give order form() |

**Book Modeofpayment**

|  |
| --- |
| Book |
| Title: string  Author: string  Publisher: string |
|  |

|  |
| --- |
| Mode of payment |
| Creditcardno:int  Checkno:int |
| Pay by cash()  Pay through money order() |

**e) Aim: Represent usecases and domain class diagram using rational rose**

**🡪Usecase diagram**

****

**->Class diagram:**

**f)Aim:** Develop CRUD matrix to represent relationship between use cases and problem domain classes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entity/function | customer | login | address | logout | seller | order | book | Mode of payment |
| Inventory |  |  |  |  | C,R |  |  |  |
| Create order | C,R,D |  |  |  | C | C |  | C,U |
| Bookcatalog | R |  |  |  | C |  | R |  |
| Login |  | C,R |  |  | C,R |  |  |  |
| Management payment | R |  |  |  | C |  | C | C,R |
| Order status |  |  | R |  | C | C |  |  |

**Result:** The Design was successfully completed.

**EXPERIMENT4:**

**CASE STUDY: LIBRARY MANAGEMENT SYSTEM**

1. **Aim:** Identify and analyze events

The flow of events of a usecase contains the most important information derived from usecase modeling work. It should describe the usecases flow of events clearly enough for an outsider to easily understand it. Remember flow of events should present what the system does, not how the system is design to perform the required behavior.

Guidelines for the contents of the flow of events are,

1) Describe how the usecase starts and ends.

2) Describe what data is exchanged between the actor and the usecase

3) Do not describe the details of the user interface, unless it is necessary to understand the behavior of the system.

**b) Aim:** Identify usecases

**Procedure:**

a)**Register member:**

Here the members who can access the library are registered based on the id nos.

b) I**ssue book:**

**b.1) verify member:**

Before issuing book the specific member is verified whether he/she is registered or not

**b.2) check availability or book:**

After verifying the person then the availability of book which he/she selected is checked whether the book is available or not after verifying & checking the book is issued

1. **Return book:**

After the specific time is over the person should return book during returning

**c.1) Calculate fine:**

The fine is calculated

**d) Enquiry:**

In this the member is checked if fine is existed or not

**e) Maintaining books:**

This is done by the librarian that how many books are present, person or members login id and their registrations etc.

**Graphical representations:**

****

1. **Develop event table:**

|  |  |
| --- | --- |
| **Usecase** | **Description** |
| Register member | The members who can access the library are registered based on the ids |
| Issue book-verify member | Before issuing the book the specific member is verified whether he/she is registered or not |
| Check availability of book | Check the book is available or not |
| Return book | After the specific time is over the person should return |
| Calculate fine | The fine is calculated if their fine |
| Enquiry | In this the member is checked if fine is existed or not |
| Maintaining books | This is done by the librarian to cross checking the books |

**d) Aim:** Identify and analyze domain classes



**e) Aim:** Represent class diagram for library management system

🡪Class diagram



**🡪Usecase diagram**



**f) Aim:** Develop CRUD matrix to represent relationships between usecase and problem domain classes

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entity/Function | Librarian | Transaction | Book | Journals | Study Books | Magzines | Bill | Memberrecord | Student | faculty |
| Register member | UC,D |  |  |  |  |  |  |  |  |  |
| Enquiry | C |  |  |  |  |  |  | R | R | R |
| IssueBook | C |  | R | R | R | R |  |  |  |  |
| VerifyBook | R |  |  |  |  |  |  |  |  |  |
| VerifyMember |  |  |  |  |  |  |  |  |  |  |
| CheckAvalilability of Book |  |  | R | R | R | R |  | R | R | R |
| ReturnBook | R | R |  |  |  |  |  |  | C | C |
| CalculateFine | C | C |  |  |  |  |  |  | R | R |
| MaintainBooks | C |  |  |  |  |  |  |  |  |  |

**Result:** The Design was successfully completed.