**Name** Login

**Participant** bob:Librarian

**Instances** alice:Subscriber

**Flux of events** 1.Alice and Bob want to use the terminal to see the book list

2.Alica and Bob walk to the terminal and enter their data

3.The terminal checks the data

4.Alice and Bob succesfuly login

**Name** Login

**Participant** bob:Librarian

**Instances** alice:Subscriber

**Flux of events** 1.Alice and Bob want to use the terminal to see the book list

2.Alica and Bob walk to the terminal and enter their data

3.The terminal checks the data

4.Alice and Bob fail to login due to wrong credentials

5.The terminal shows and error message

**Name** Manage all subscribers

**Participant** bob:Librarian

**Instances**

**Flux of events** 1.Bob wants to see a list of all the subscribers in the system

2.Bob logins at the system and the terminal will show him a list of all subsribers

**Name** Manage all books

**Participant** bob:Librarian

**Instances**

**Flux of events** 1.Bob wants to see a list of all the books in the system

2.Bob logins at the system and the terminal will show him a list of all books,how many are available and how many are borrowed

**Name** Consult available books

**Participant** alice:Subscriber

**Instances**

**Flux of events** 1.Alice wants to borrow a book from the library

2.Alice logins at the terminal

3.Alice will see a list of all the available books to choose from

**Name** Add new subscriber

**Participant** bob:Librarian

**Instances** alice:Subscriber

**Flux of events** 1.Alice wants to become a subscriber

2.Alice walks to a Librarian(Bob) and asks him to create her

an account

3.Bob logins with his account at the terminal,chooses the Manage all Subscribers option,then the Add new Subscriber

option

4.Bob enters Alices data in the terminal

5.The terminal checks if the new Subscriber doesn’t exist in the system

6.Alice becomes a new subscriber

**Name** Add new subscriber

**Participant** bob:Librarian

**Instances** alice:Subscriber

**Flux of events** 1.Alice wants to become a subscriber

2.Alice walks to a Librarian(Bob) and asks him to create her

an account

3.Bob logins with his account at the terminal,chooses the Manage all Subscribers option,then the Add new Subscriber

option

4.Bob enters Alices data in the terminal

5.The terminal checks if the new Subscriber doesn’t exist in the system

6.An error message pops up,saying Alice is a subscriber

7.The terminal asks again for the information

**Name** Update existing subscriber

**Participant** bob:Librarian

**Instances** alice:Subscriber

**Flux of events** 1.Alice wants to change her subscriber information

2.Alice walks to a Librarian(Bob) and asks him to uptade her

account

3.Bob logins with his account at the terminal,chooses the Manage all Subscribers option,then the Update existing Subscriber option

4.Bob enters Alices new data in the terminal

5.The terminal checks if the Subscriber does exist in the system

6.Alices data is updated

**Name** Update existing subscriber

**Participant** bob:Librarian

**Instances** alice:Subscriber

**Flux of events** 1.Alice wants to change her subscriber information

2.Alice walks to a Librarian(Bob) and asks him to uptade her

account

3.Bob logins with his account at the terminal,chooses the Manage all Subscribers option,then the Update existing Subscriber option

4.Bob enters Alices new data in the terminal

5.The terminal checks if the Subscriber does exist in the system

6.An error message pops up,saying that Alices information is invalid

7.The terminal goes back to the Manage all Subscribers menu

**Name** Delete existing subscriber

**Participant** bob:Librarian

**Instances** alice:Subscriber

**Flux of events** 1.Alice wants to delete her subscriber information

2.Alice walks to a Librarian(Bob) and asks him to delete her

account

3.Bob logins with his account at the terminal,chooses the Manage all Subscribers option,then the Delete existing Subscriber option

4.Bob enters Alices data in the terminal

5.The terminal checks if the Subscriber does exist in the system

6.Alices data is removed

**Name** Delete existing subscriber

**Participant** bob:Librarian

**Instances** alice:Subscriber

**Flux of events** 1.Alice wants to delete her subscriber information

2.Alice walks to a Librarian(Bob) and asks him to delete her

account

3.Bob logins with his account at the terminal,chooses the Manage all Subscribers option,then the Delete existing Subscriber option

4.Bob enters Alices data in the terminal

5.The terminal checks if the Subscriber does exist in the system

6.An error message pops up,saying that Alice is not a subscriber

7.The terminal goes back to the Manage all Subscribers menu

**Name** Add new Book

**Participant** bob:Librarian

**Instances**

**Flux of events** 1.Bob recieved a new book at the library

2.In order for the book to be borrowed,the book must appear in the Book List

3.Bob logs in the system,activated the Manage all Books option,then activates the Add new Book option

4.Bob inserts the information of the book in the terminal

5.The system checks the information in order to see if the book exists in the data base already

6.The new book is added

**Name** Add new Book

**Participant** bob:Librarian

**Instances**

**Flux of events** 1.Bob recieved a new book at the library

2.In order for the book to be borrowed,the book must appear in the Book List

3.Bob logs in the system,activated the Manage all Books option,then activates the Add new Book option

4.Bob inserts the information of the book in the terminal

5.The system checks the information in order to see if the book exists in the data base already

6.An message appears that the book already exists in the system

7.In this case the book is not added as a new one,but the quantity of the book is modified

**Name** Update existing Book

**Participant** bob:Librarian

**Instances**

**Flux of events** 1.Bob recieved information that a publishing house wants to wants to modify some information for a specific book

2.Bob logs in the system,activated the Manage all Books option,then activates the Update existing Book option

3.Bob choses a book from the Book list

4.Bob inserts the new data for the selected book

5.The system checks the information

6.A message pops up, saying that the information has been updated

**Name** Update existing Book

**Participant** bob:Librarian

**Instances**

**Flux of events** 1.Bob recieved information that a publishing house wants to wants to modify some information for a specific book

2.Bob logs in the system,activated the Manage all Books option,then activates the Update existing Book option

3.Bob choses a book from the Book list

4.Bob inserts the new data for the selected book

5.The system checks the information

6.A message pops up, saying that the information inserted is invalid

7.The system will return Bob to the Manage all Books menu

**Name** Delete existing Book

**Participant** bob:Librarian

**Instances**

**Flux of events** 1.Bob wants to remove a book from the data base

2.Bob logs in the system,activated the Manage all Books option,then activates the Delete existing Book option

3.Bob choses a book from the Book list

4.Bob presses the delete button

5.A message pops up, saying that the information has been deleted and the book dissapears from the Book list

**Name** Return existing Book

**Participant** bob:Librarian

**Instances** alice:Subscriber

**Flux of events** 1.Alice wants to return a book she borrowed

2.Alice goes the a terminal where Bob works

3.She hands the book to Bob

4.Bob logs in the system,activated the Manage all Books option,then activates the Return existing Book option

5.Bob selects the book from the Book List

6.Bob presses the Return borrowed book button on the terminal

7.The system updates the available quiantity of the returned book

8.The system pops up a message that says that the book as been returned succesfuly

**Name** Return existing Book

**Participant** bob:Librarian

**Instances** alice:Subscriber

**Flux of events** 1.Alice wants to return a book she borrowed

2.Alice goes the a terminal where Bob works

3.She hands the book to Bob

4.Bob logs in the system,activated the Manage all Books option,then activates the Return existing Book option

5.Bob selects the book from the Book List

6.Bob presses the Return borrowed book button on the terminal

7.The system updates the available quiantity of the returned book

8.The system pops up a message that says that the book cannot be return,since it was not borrowed

**Name** Borrow existing book

**Participant** alice:Subscriber

**Instances**

**Flux of events** 1.Alice wants to borrow a book

2.She goes to a terminal in the library,logs in and activates the Borrow existing book option

3.She choses a book from the book lists

4.Alice presses the Borrow book button

5.The system updates the available quantity of the book

6.A message pops up, saying that the book has been borrowed succesfuly

7.Alice goes then to recieve the book from a librarian