ManagerActor income:float fireLibrarian(ma_id:int, L_id:int) registerLibrarian(ma_id:int, L_id:int, fname:string, Iname:string,password:string) getIncome() enterMAID() enterLibrarianID() enterFirstName() enterLastName() enterPassword()

BookRecord

b_id:int

title:string

author:string

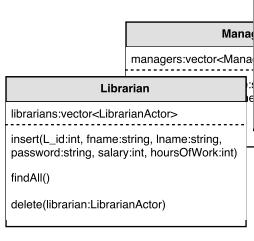
getBookID()

getAuthor()

getPublisher()

getTitle()

publisher:string



		getLastActivity()	
		getFname()	
		getLname()	
	Mana	getPassword()	
	managers:vector <mana< td=""><td>setLastActivity(Date:Date8</td><td>kTime)</td></mana<>	setLastActivity(Date:Date8	kTime)
Librarian		getIsLogin()	
ns:vector <librarianactor></librarianactor>		setIsLogin(islogin:bool)	
L_id:int, fname:string, Iname:string, ord:string, salary:int, hoursOfWork:int)			
()			
(librarian:LibrarianActor)			

UserActor

u_id:int

isLogin:bool

fname:string

Iname:string

password:string

logout(u_id:int)

getUID()

getIsLogin()

lastActivity:Date&Time

login(u_id:int, password:string

u_id:int

b_id:int

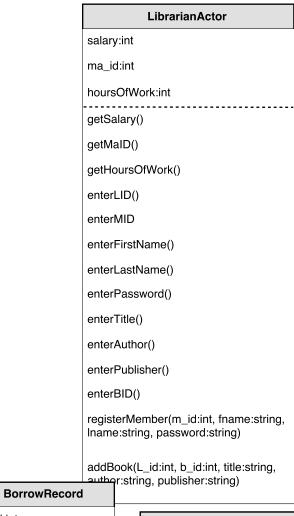
from:Date&Time

to:Date&Time

getUID()

getBID() getFrom()

getTo()



Borrow

borrows:vector<BorrowRecord> delete(borrowBook:BorrowRecord) insert(b_id:int, m_id:int, from:Date&Time, to:Date&Time) findAll()

User users:vector<UserActor> insert(u_id:int, fname:string, lname:string, password:string) findAll() find(u_id:int) update(user:UserActor) getUserType(u_id:int)

> MemberActor age:int phoneNumber:string enterMID() getAge() getPhoneNumber() enterTitle() search(title:string, m_ID:int)

books:vector<BookRecord> find(title:string) insert(b_id:int, title:string, author:string,

Book

publisher:string) findAll()

Admin

Member

insert(m_id:int, fname:string, lname:string,

password:string, age:int, phoneNumber:string)

members:vector<MemberActor>

findAll()

name:string

registerManager(ma_id:int, fname:string, Iname:string)