
Software Requirements Specification

For



Library Management System

Prepared for:

Ardeshir Bagheri

Prepared by:

Peter Abelseth
Ken Anglas
Matthew Hinton
Daniel Huettner
Sardor Isakov
Jeremy Lerner
Carlos Lozano

Term project for CPSC 2301 - Langara College

Fall 2012



Table of Contents

Table of Contents	1
1. Introduction	1
1.1. Purpose of the System	1
1.2. Project Scope	1
2. Current System	1
3. Proposed System	1
3.1. Overview	1
3.2. Functional Requirements	2
3.3. Nonfunctional Requirements	7
3.4. System Models	7
3.4.1. Scenarios	7
3.4.2. Use case models	12
3.4.3. Object model	46
3.4.4. Class Diagrams	51
3.4.5. Dynamic Models	59
3.4.6. User Interface	68
4. Definitions, Acronyms and Abbreviations	76
5. References and Acknowledgements	77

Version	Primary Author(s)	Description of Version	Date Completed
First delivery	Carlos Lozano	First draft of the SRS including software requirement elicitation and system models	22/10/12
Corrections	Carlos Lozano	Corrections and reorganization of requirements, sequence diagram corrections.	08/12/12
Corrections	Daniel Huettner	Corrected class diagrams.	08/12/12
Correction	Jeremy Lerner	Corrected class diagrams.	08/12/12

1. Introduction

1.1. Purpose of the System

The system requested by the client is a Library Management System for the CICSIR at UBC that caters to Computer Science and Engineering students. The CICSIR is in need of a new Library Management System to track and manage its resources. The purpose of the system is to satisfy CICSIR's needs of a new library system to manage resources for loan, reference material, and library management operations.

This document describes the requirements of the project giving a detailed description of functional and non-functional requirements as per requirements given by the client complemented by elicited and discovered requirements obtained during the Requirement Elicitation stage of this project.

1.2. Project Scope

"Libris" Library Management System is intended for the use of the CICSIR Library located in the UBC grounds for the use of students of Computer Science and Engineering and drop in research patrons.

The software will manage loans of different resources and temporary loan of reference items, additionally the software will also handle library specific functions related to resource management including orders, magazine subscriptions and collection of user fines. It will also allow automation of diverse periodical tasks related to loans, management of resources, and management of users.

Main benefits include an organized and efficient front end to facilitate management of loans and resources in a more functional way allowing librarian and patrons a more pleasant and agile experience at the library.

It is not intended to connect this program to any external resource database or an external library system at the University of British Columbia or anywhere else.

2. Current System

The current system is non-existing; this is a new system proposal.

3. Proposed System

3.1. Overview

The proposed system for the benefit of the CICSIR Library at UBC will facilitate the day-to-day operations of the library and will allow more administrative control over resources, users, subscriptions, fines, and maintenance of their database.

This document intended mainly for the client and end-users will introduce the reader diverse

aspects of the proposed system starting with the set of user and system requirements decomposed in functional and non-functional requirements; secondly we will introduce some scenarios and the use cases for the whole system product of the requirements elicitation stage of this project. Lastly we will introduce the objects, classes, and dynamic models and the proposed user interface.

3.2. Functional Requirements

Functional requirements grouped by the functionality:

#	Functional Requirement	Priority
Catalog		
R1.	A staff member shall be able to add/remove resources	High
R1.1.	A resource cannot be removed if it is currently checked out or on reference	High
R1.2.	A resource cannot be added if it is a duplicate of a resource already present	High
R2.	A staff member must be able to view/change resource information	High
R2.1.	A resource's information can be edited by only one staff member at a time	High
R3.	A staff member must be able to add/remove resource copies	High
R3.1.	A resource cannot be removed if any of its copies are checked out	High
R3.2.	A resource cannot be removed if any of its copies are on reference	High
R4.	A staff member must be able to add and remove resource copies from Reference	High
R4.1.	A resource copy must be placed on reference under a faculty member's name or a staff member's name	High
R5.	A staff member must be able to add/remove/renew subscriptions	High
R5.1.	No magazine subscriptions may be duplicates of each other	High
R6.	A user must be able to search for resources using a search criteria	High
R6.1.	User must be able to see if the resource is available or on reference	High
R7.	A staff member must be able to view a resource subscription	High
R7.1.	The staff member must be able to see when the subscription was last renewed	High
R7.2.	The staff member must be able to see when the subscription expires	High
R7.3.	The staff member must be able to see when the subscription was first bought	Med.
R8.	A staff member shall be able to view/change resource copy information	High
Circulation		
R9.	A staff member shall be able to view/change fines from patrons	High
R9.1.	A patron's information can be edited to remove fines from account	High
R9.2.	A payment will be marked as paid	High
R9.3.	A record of fines and paid fines will be kept	High
R10.	A staff member shall be able to view currently checked out resources of a user	High
R10.1.	A history will be kept for resources checked out	High
R11.	A staff member should be able to see the history of checked out resources	Med.

	of any user.	
R11.1.	History will be given for a minimum of last 5 actions	Med.
R11.2.	Actions include checked out material, fines, and notifications	Med.
R12.	A staff member shall be able to renew/extend due date of checked out material not requested by another patron.	High
R12.1.	Resource will be marked if requested by another patron	High
R12.2.	It is not possible to perform function of renew if its requested	High
R13.	A staff member shall be able to issue/check-out material for patrons	High
R13.1.	Material shall not be on reference or reserved for another patron	High
R14.	A patron shall not be able to exceed a limit of checked-out resources	High
R14.1.	Limit will be checked upon request for checkout	High
R14.2.	No more than 30 resources may be checked	High
R15.	Only available resources can be checked-out	High
R15.1.	Reference material will not have option to check out	High
R16.	Reference material cannot be checked-out (or removed from library).	High
R16.1.	Referenced material will be marked on the resource information	High
R17.	Patrons cannot check-out more than one copy of same resource	High
R17.1.	Checkout will reject loan of two copies of same resource	High
R18.	A staff member shall be able to check-in returned material	High
R18.1.	Staff member can check-in material with or without a patron	High
R19.	Staff member shall be able to put a resource on reserve for another user	High
R19.1.	User could be patron, faculty, or another staff member	High
R20.	A patron shall be able to reserve material already checked-out	High
R20.1.	Patron will be notified once material has been checked-in	High
R20.2.	Patrons reserve on a resource expires after 2 weeks	High
R21.	Patron shall be able to check his reservation records and his position in the queue for materials that the patron currently has on reserve	High
R21.1.	Any patron shall be able to recall list of items put on hold.	High
R21.2.	System should be able to display reservation details for that patron.	High
R22.	System shall able to update the reserve queues once resources are returned	High
R22.1.	Once a book is checked in, system should be able to check for reservations	High
R22.2.	If a hold is found, system should be able update the queue for that book	High
R23.	Librarian shall be able to cancel reservations	High
R23.1.	Librarian shall be able to see patron's reservation records	High
R23.2.	Librarian should be able to select items to cancel	High
R23.3.	System shall ask librarian to proceed or to cancel removal	High
R24.	A Patron shall be able to view a list of their own checked-out resources	High
R25.	A Patron shall be able to view a summary of their current outstanding fines	High
R25.1.	Patron shall be able to view what money is owed in fines	Med.

R26.	A staff member shall be able to view patrons' reserved resources	High
R27.	Patrons shall be able to cancel their own reservations	High
R27.1.	Patrons shall be able to list reservations	High
R27.2.	Patron shall be able to select reservations to cancel	High
R27.3.	Patron shall be able to cancel selected reservations	High
R28.	A staff member shall be able to put/remove a resource on reference for a faculty member	High
R28.1.	Librarians shall be able to open the reference master to add or remove reference items.	High
R28.2.	Librarian shall be able to identify the faculty member adding or removing item to/from the reference master.	High
R29.	A staff member shall be able view the list of reserved resources(from Resource Record)	High
R29.1.	A staff shall be able to a list of reserved items	High
R30.	A staff member shall be able to view resource history and resource copy history	High
R30.1.	A staff member shall be able to open history of any resource to see who checkout resource in the past.	High
R31.	System shall be able to remove an expired reservation	High
R31.1.	System shall be able to drop reservations not filled by 2 weeks after due date of return.	High
R32.	Patrons are fined \$0.25 per day that resources are overdue, to a maximum of \$5.00 per overdue item	High
R32.1.	Staff members do not pay fines	High
R32.2.	Fines are no longer increased when a resource is returned	High
R33.	Student can check out a book for 4 weeks	High
R34.	Faculty member may check out a book for 3 months	High
R35.	Staff member may check out a book for 1 year	High
R36.	Resources other than books may be checked out for 1 week at a time, regardless of the user class	High
User Management		
R37.	Administrator shall be able to change a user class	High
R37.1.	An administrator cannot change their own user class	High
R37.2.	An Administrator can change the user class of another Administrator	High
R38.	A staff member shall able to add and disable students and faculty members from the system	High
R38.1.	A warning message is given if the account has unpaid fines or resources check out	High
R38.2.	A staff member shall be able to view student/faculty information	High
R39.	A staff member shall be able to view/change patrons' information (name, ID number, etc.)	High
R40.	A staff member shall be able to disable an account that currently has no outstanding fines	High
R40.1.	A staff member shall be able to view what resources and outstanding fines the user has	High

R40.2.	The account shall not be the staff member himself who is deleting the account	High
R41.	An administrator shall be able to import a list of new patrons to the database	High
R41.1.	A patron shall not already be in the system	High
Reporting		
R42.	System shall be able to send email reminders of materials that is nearly due and overdue	High
R42.1.	The system shall keep up to date records of which resources are overdue	High
R42.2.	The system shall be able to send emails to users	High
R43.	System shall be able to send an email message to a patron when a resource he/she has on reserve becomes available.	Med.
R43.1.	The system shall be able to detect when a resource on reserve is check in	Med.
R43.2.	The system shall be able to clearly notify the staff member when they have checked in a resource that has been reserved.	Med.
R44.	A staff member shall be able to view resources that are currently checked out and overdue.	High
R44.1.	The system shall keep an up to date record of currently checked out and overdue resources.	High
R44.2.	The system shall allow staff members to view checked out resources and overdue resources.	High
R45.	A staff member shall be able to view all outstanding fines for all users	Med.
R45.1.	The system shall keep up to date records of fines for a user and for which loan the fines were incurred	Med.
R46.	System shall be able to create TO-DO to renew a subscription	Low
R46.1.	The system shall be able to determine if a subscription needs renewing	Low
Browsing & Guest Access		
R47.	User shall sign in to a terminal using a username and password (a terminal is a computer running the client software)	High
R47.1.	The system shall be able to verify a user's username and password	High
R47.2.	The system shall be able to determine the access level of a user	High
R48.	Patrons shall be able to search catalog for resources	High
R48.1.	The system shall maintain up to date records of all resources.	High
R48.2.	The system shall be able to determine if a resource is available.	High
R49.	By default, if no one is logged in, the client is in a "guest" mode	High
R49.1.	The system shall support a public interface where anyone can search for resources	High
R50.	In guest mode, the only commands available are: "search for resource", "login"	High
Application Management		
R51.	A staff member shall be able to start/shutdown the server program	High
R51.1.	Any authorized staff member shall be able start/shutdown the server from their computer	High
R51.2.	A staff member issues "shutdown server" command from the client computer	High

R51.3.	Staff member clicks start the server button, system issues “start server” action	High
R51.4.	Server system receives the “shutdown server” command and stops all running programs by killing all processes	High
R51.5.	Server system receives command and executes server program	High
R52.	A staff member shall be able to start/shutdown the client	High
R52.1.	Client program can be started by clicking on start	High
R52.2.	A staff member can stop the client by closing the client program	High
R52.3.	Client program start and connects to server program which hosted on remote machine	High
R52.4.	Starting the client shall be restricted to authorized users listed on a management access control list	High
R52.5.	Client disconnects all connection with the server	High
R52.6.	Close client shall be restricted to authorized users listed on a management access control list	High
R53.	Any user shall be able to sign into the client and sign out	High
R53.1.	User will be prompted to enter her/his username, password and click “enter”	High
R53.2.	The system shall authenticated user by checking correctness of the login and looking up user’s table from database	High
R53.3.	The system shall automatically determine user role and credentials in the system	High
R53.4.	The system shall open correct user mode according user’s role and credentials	High
R53.5.	The system can be sign out by clicking sign out button	High
R54.	System shall be able to create TO-DO for expired reference resources	Low
R55.	At the end of the day a staff member shall be able to create/restore a backup of database	Med.
R55.1.	To create a backup of database, user authorized and listed on a management access control list	Med.
R55.2.	A staff member clicks “backup database” button, system prompts for user’s password	Med.
R55.3.	The system checks user credentials, if access granted, the system starts backing up the database	Med.
R55.4.	A staff member, whose user credentials has been approved, can perform restore backup of database.	Med.
R55.5.	The system shall be able to restore backup of database, when user issues restore command	Med.

3.3. Nonfunctional Requirements

#	Non-Functional Requirement
R101.	The client should take no longer than 5 seconds to respond to a simple requests and no longer than 10 second seconds to start displaying results for more complex requests that fetch multiple results, not exceeding 48 seconds to complete request.
R102.	System shall be able to perform several transactions at a given time, but only one user shall be able to change a database record at a time.
R103.	System shall have clear, friendly and intuitive user interfaces adaptable to the type of user.
R104.	System shall be able to support up to 24 users at the same time.
R105.	Guest and unidentified patrons shall not need training to browse the catalog.
R106.	System shall be able to support more than 24 tasks per minute.
R107.	System shall be able to survive invalid user input.
R108.	System shall be reliable having mean time to failure of one a week.
R109.	System should be available to library staff and patrons during regular opening hours to a rate of not less than 95% of the time.
R110.	System shall be able to allow the creation of new type of resources not available at the time of implementation. i.e. Microchip Books
R111.	System should be able to be modified to correct unexpected functionality and to include new requirements that do not force a change of system architecture.
R112.	System shall be able to run on any OS and hardware that supports JAVA and Java Swing functionality.
R113.	System requirements shall be able to be traced to make sure functional requirements are satisfied.

3.4. System Models

3.4.1. Scenarios

The following are some of the most representative scenarios for what the system should do:

Name:	LookForBook
Actors:	Billy:Student
Flow of Events:	<ol style="list-style-type: none"> 1. Billy, A user without an account accesses the guest system. 2. Billy wants to search for a book, whose name they can't remember, but they know the authors name. 3. Billy accesses the searching panel, and search for the author 4. The server returns the list of resources. 5. Billy opens the desired resource record to view more information. 6. The resource record shows that the only copy of the book is on reference.

Name:	ImportNewUsers
Actors:	<u>John:Administrator</u>
Flow of Events:	<ol style="list-style-type: none">1. John, An administrator wants to add new users to the database for the start of term.2. John receives a list of new students from human resources in his mail.3. John accesses the login panel, and logs in with his information.4. John requests to import new users and selects the list file.5. A prompt is given by the server to selectively pick users to add from that file.6. The server accepts the list of new students and adds their accounts to the database.7. The server displays a message that the users have been added.

Name:	PutBookOnReference
Actors:	<u>Holly:Faculty, Bob:Librarian</u>
Flow of Events	<ol style="list-style-type: none">1. Holly, a faculty member, wants to put a book on reference for her class.2. Holly approaches the front desk and asks Bob, a librarian if he can put the book on reference for her.3. Bob, who is already logged into the system, accesses the tab to add resources.4. Bob enters the information of the book into the prompt that is given, marks it as on reference, and takes the book from Holly.5. Bob sends his request with the new resource's information to the server.6. The server replies that the resource has been entered to the system.

Name:	PlaceReservationAndNotifyOfAvailability
Actors:	<u>Harry:Student, McGonagall:Staff</u>
Flow of Events:	<ol style="list-style-type: none"> 1. Harry wants to check out the book "Harry Potter and the Sorcerer's Stone", but it is currently being loaned by another user. 2. Harry places a reserve on the book by logging into his account using a client. 3. The system notifies Harry that he is 3rd in line to borrow the book. 4. The other users have checked the book out, returned it and now it is Harry's turn. 5. The system sends an email to Harry notifying him that the book is ready to be checked out. 6. The next day Harry walks up to the staff, McGonagall, and notifies her that he is picking up a reserved book. 7. McGonagall opens Harry's account and sees that he did indeed reserve the book and it is ready. 8. McGonagall gets the book and checks it out for Harry.

Name:	NotifyToPayFines
Actors:	<u>Libris:System, Gavin:Student, Geoff:Librarian</u>
Flow of Events:	<ol style="list-style-type: none"> 1. Gavin, a student with many books checked out under his name and has neglected to return his books on time. 2. Libris has records of Gavin's books under his name 3. After a week in which Gavin has not returned his books yet the system sends Gavin an email explaining how much he owes in fines, how much per day he is being charged, which books are overdue, and the dates in which they needed to be sent by. 4. The email is received by Gavin and he takes his books to the library to return them. 5. Gavin approaches the front desk and asks Geoff, a staff member to pay his fines.

Name:	RenewSubscriptionOfMagazine
Actors:	<u>Edward:Librarian</u>
Flow of Events:	<ol style="list-style-type: none"> 1. Edward notices that a subscription for “Time” magazine is about to expire by looking at the renew subscription To-Do. 2. Edward opens the record of the “Time” magazine subscription. 3. Edward sees the sales office information of “Time” magazine and phones them. 4. Edward then renews the subscription for “Time” magazine. 5. Edward updates the magazine record in the system to the new subscription expiry date.

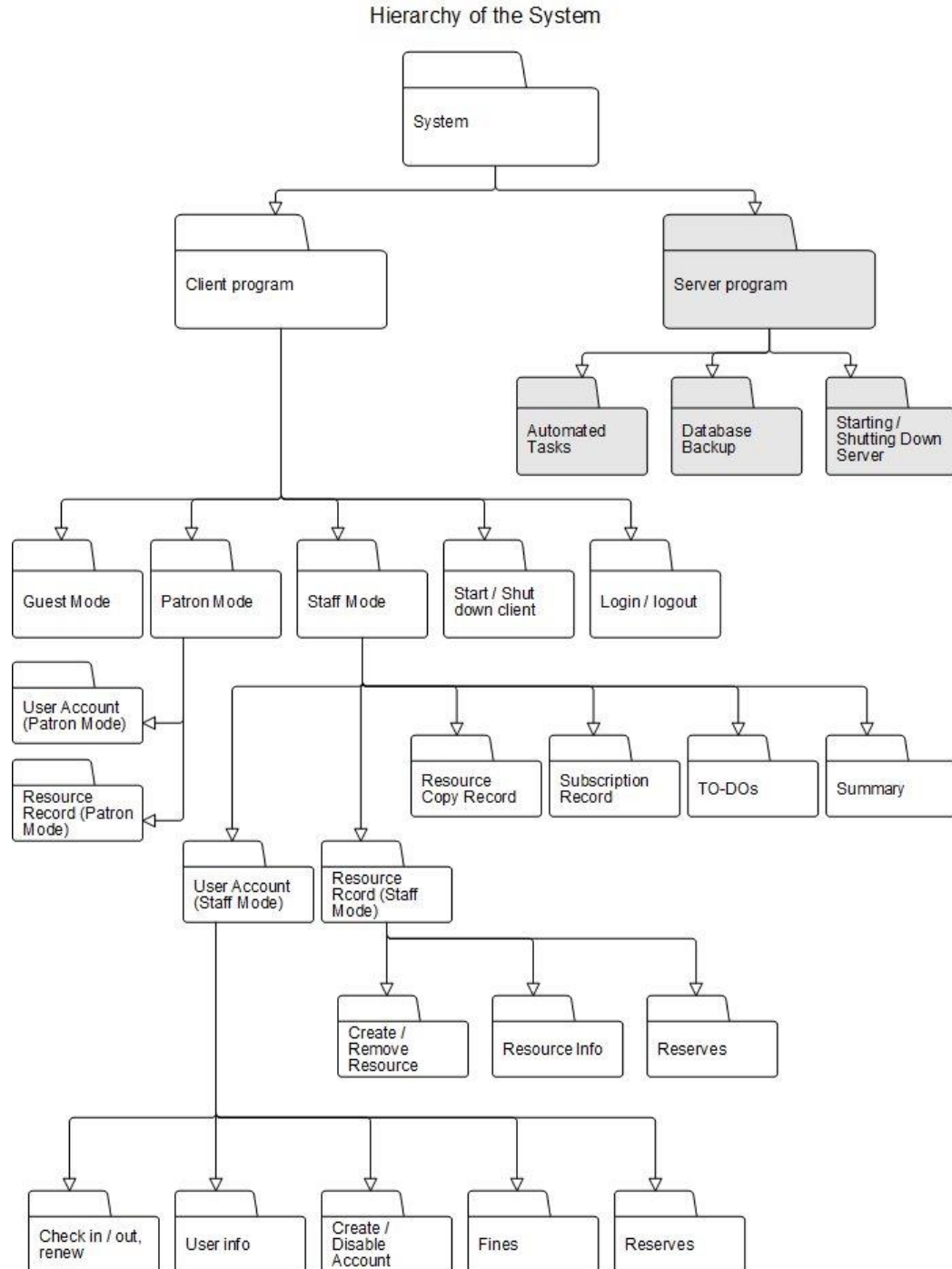
Name	CheckOutBook
Actors	<u>Barack:Patron, Michelle:Librarian</u>
Flow of Events	<ol style="list-style-type: none"> 1. Barack looks for a book named “Republican Romnesia” 2. Barack finds the book and takes it to the Michelle. 3. Michelle asks Barack for his Library card and enters it in the system. 4. Michelle opens book record and clicks on “Checkout” button. <ol style="list-style-type: none"> 5. The system checks whether Barack meets any of the conditions to refuse a loan and does not find one, loan is approved 6. Michelle prints receipt with due date, puts it in the book and hands book and receipt to Barack.

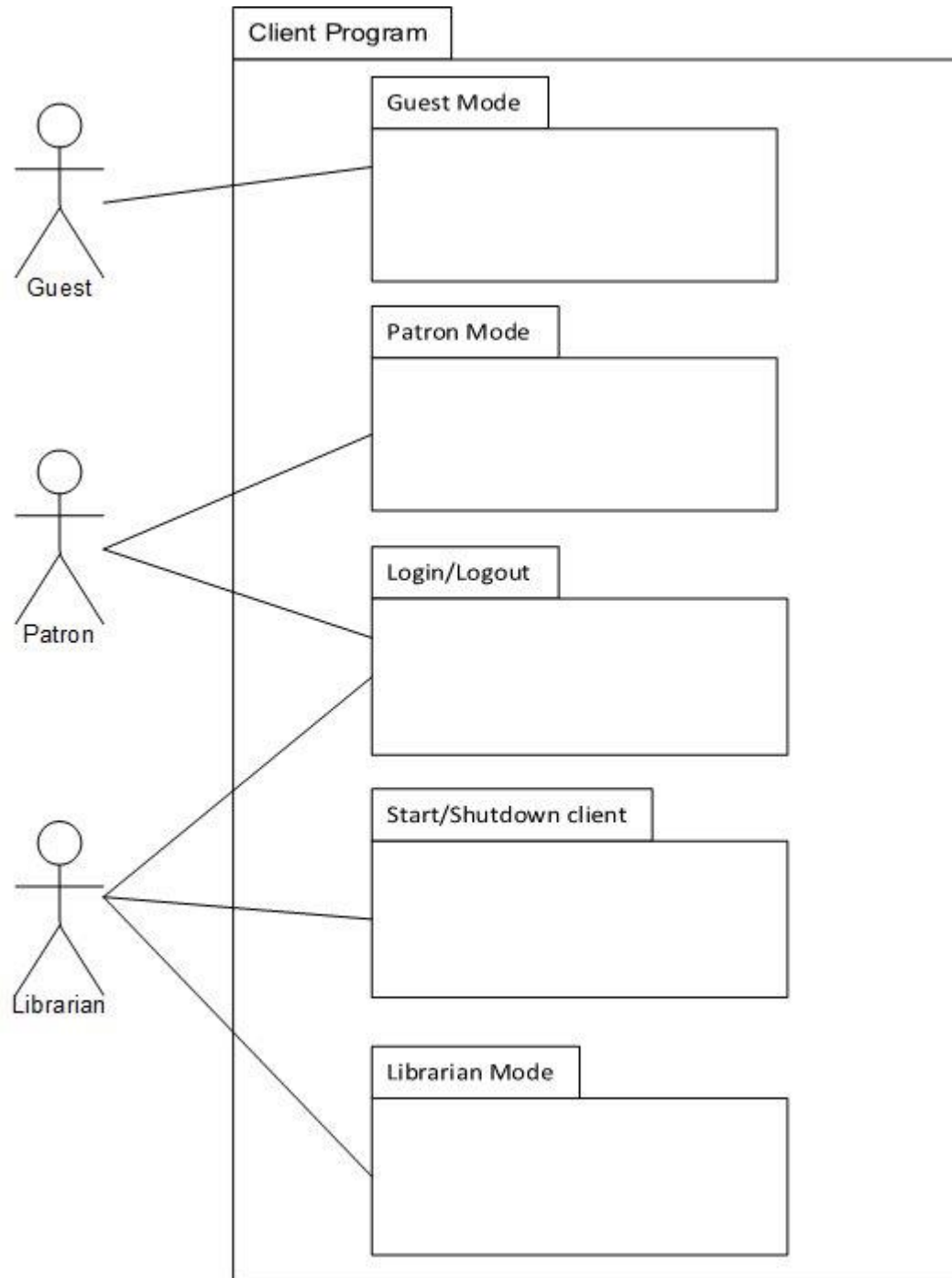
Name:	CreateUserAccount
Actors:	<u>Roxi:Librarian, Bert:Student</u>
Flow Of Events:	<p>Bert is new and has no user name and library card.</p> <ol style="list-style-type: none"> 1. Bert asks Roxana to open and account for her. 2. Roxi validates student ID and issues command "Add new user". <ol style="list-style-type: none"> 3. System prompts for person’s info: <ul style="list-style-type: none"> • first and last name • username • email • password (and password confirmation) • User class (Administrator, staff member, faculty, student) 4. Roxi confirms information and sends it to the database. <ol style="list-style-type: none"> 5. System confirms creation of new user 6. Roxi prints account details and hands receipt and library card to Bert.

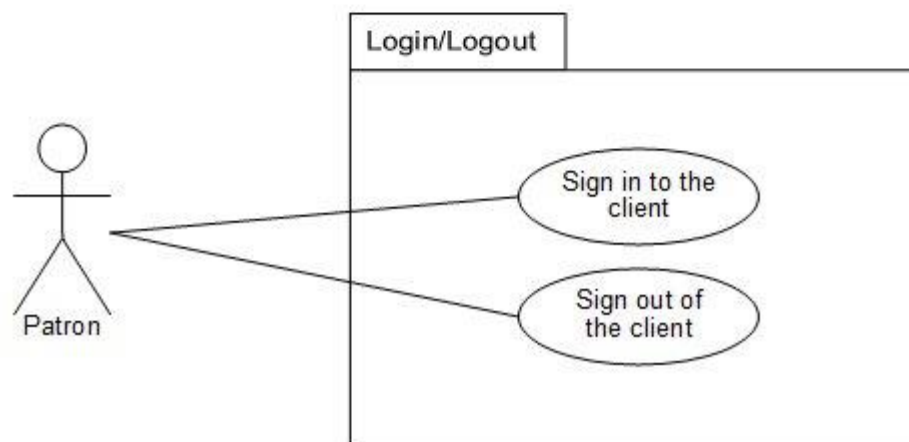
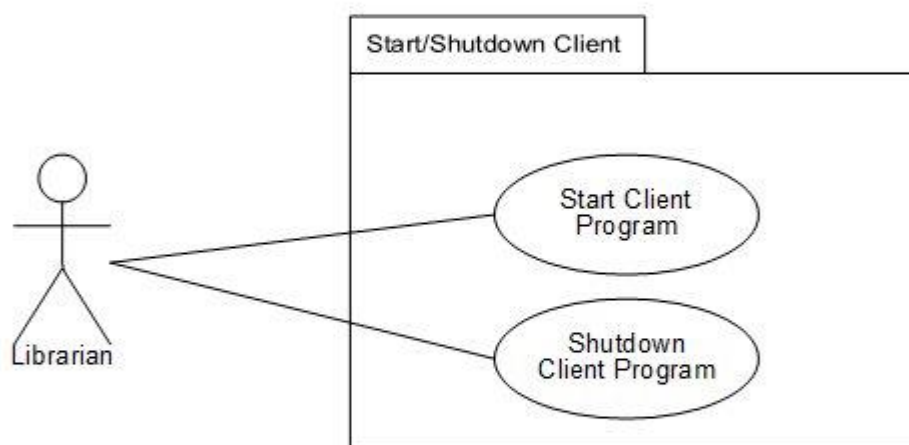
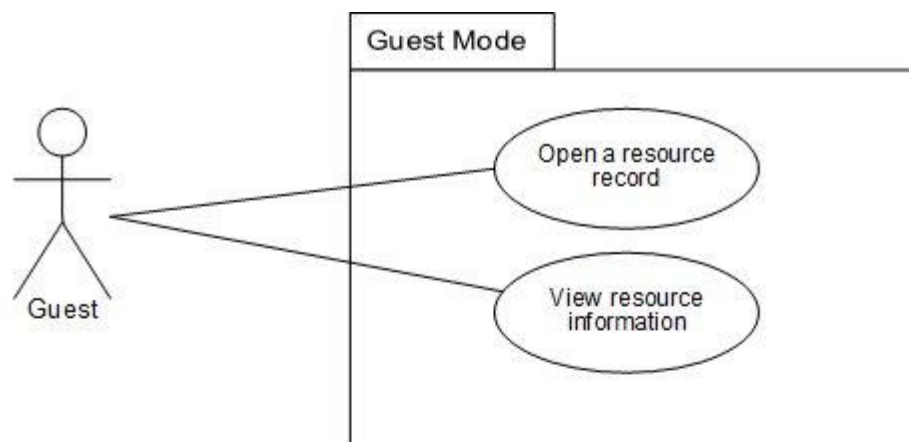
Name:	PayFine
Actors:	<u>Linda:Librarian, Ferdinand:Patron</u>
Flow Of Events:	<ol style="list-style-type: none">1. Ferdinand has some outstanding fines, he approaches Linda the librarian2. Linda opens Ferdinand record3. Linda informs Ferdinand that he owes \$3.504. Ferdinand hands \$5.00 to Linda5. Linda zeroes Ferdinand fines6. Linda hands Ferdinand his change and a receipt for \$3.50

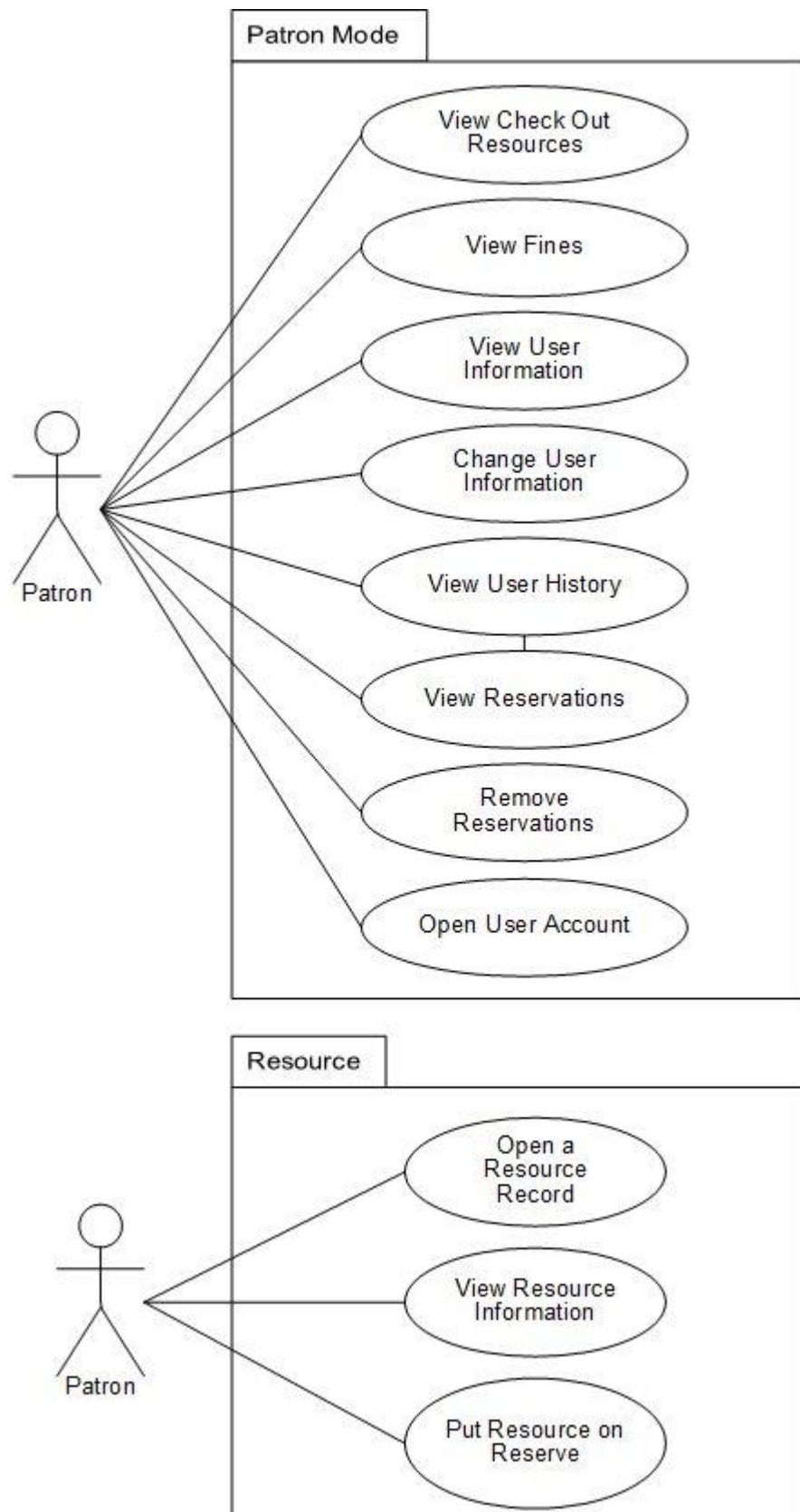
3.4.2. Use case models

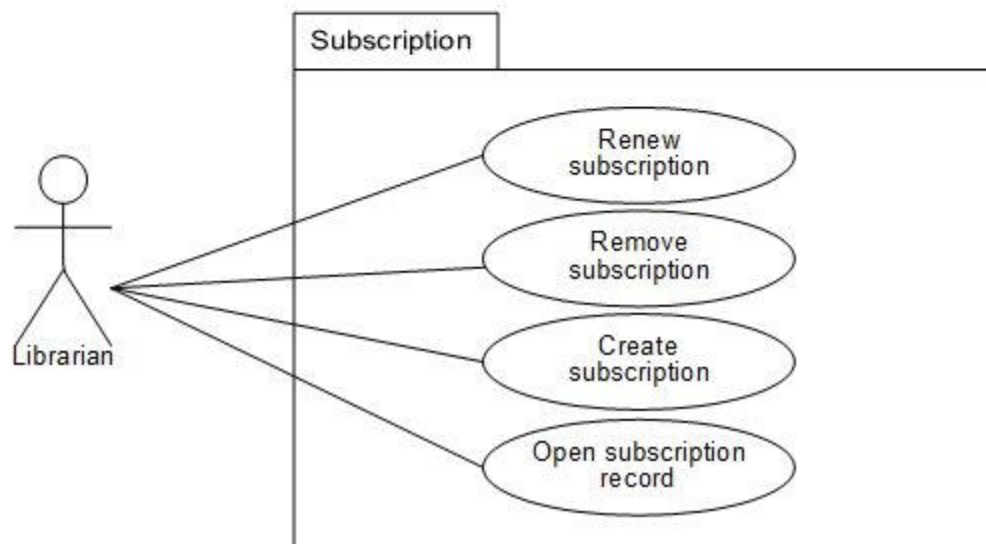
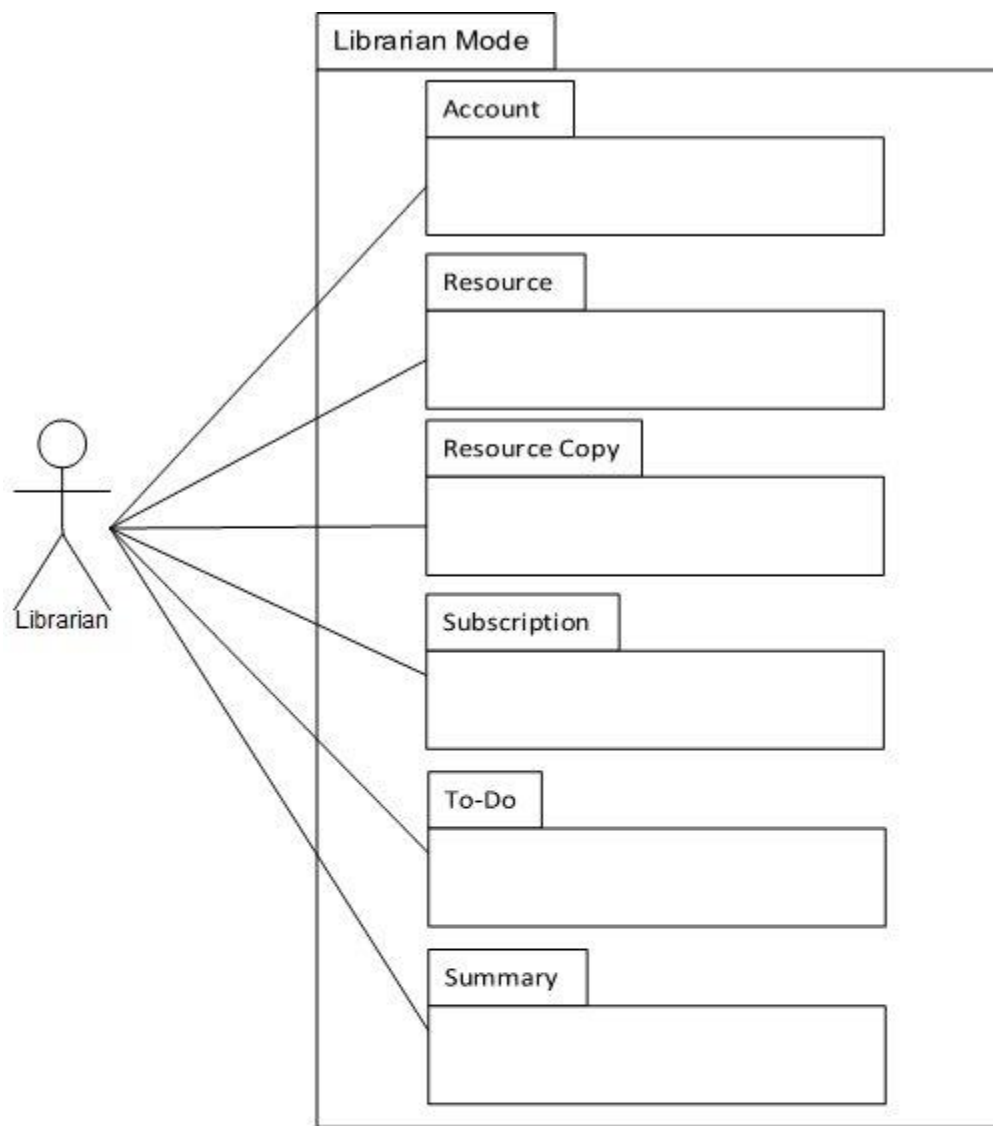
To understand better how the proposed system will work, the following graphic describes the hierarchy of the whole system:

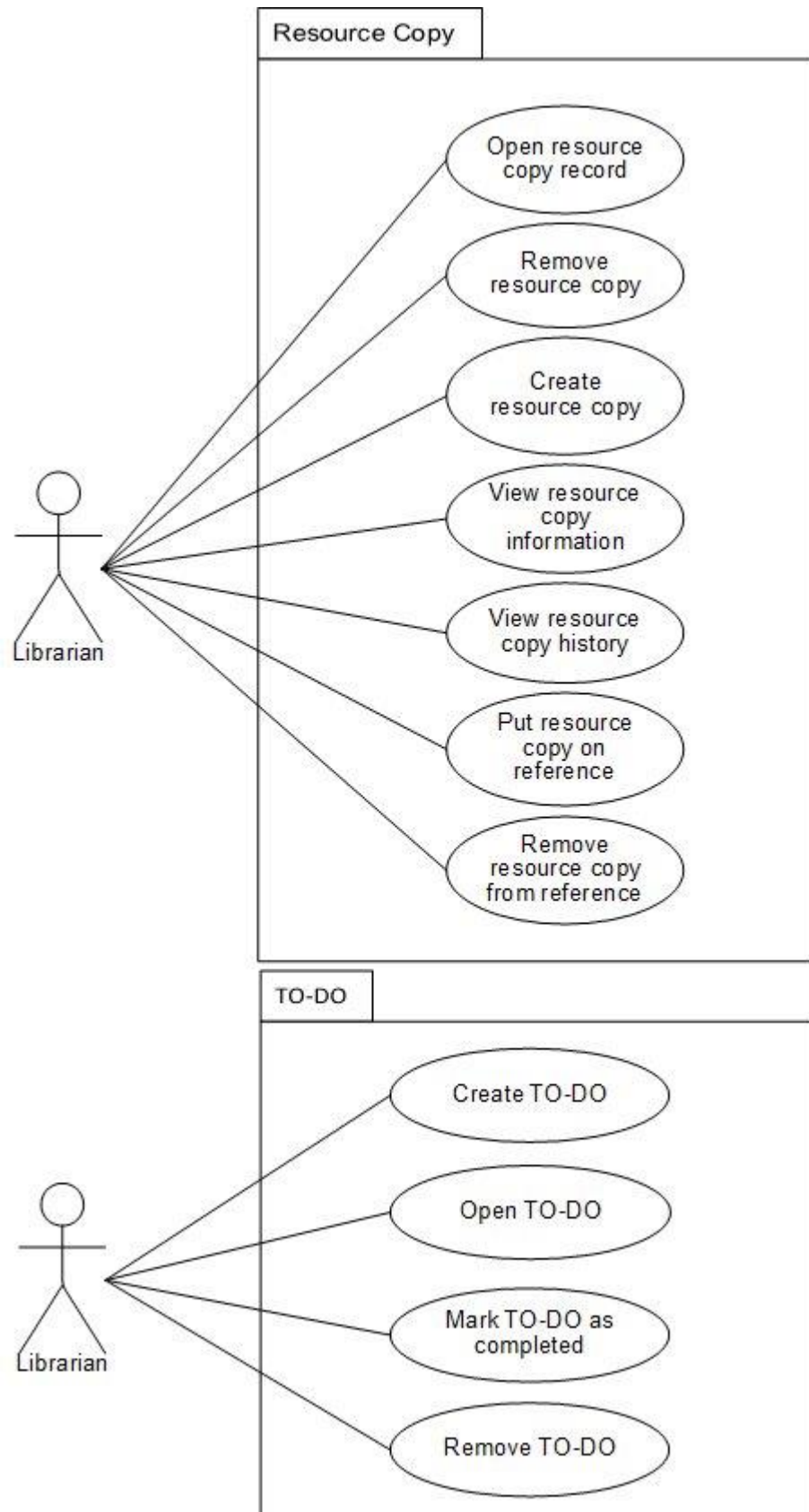


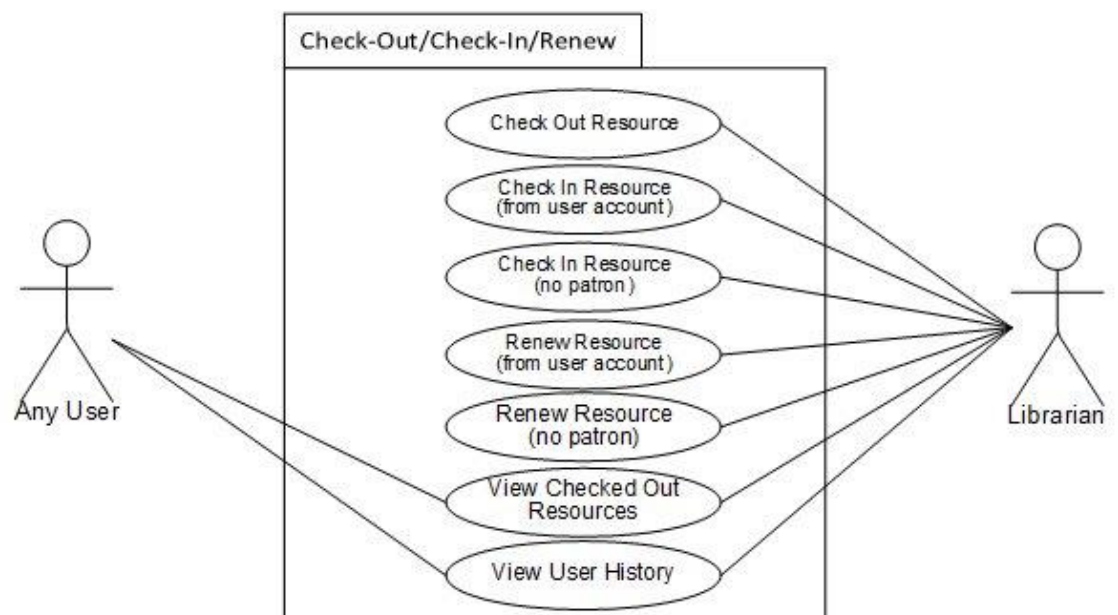
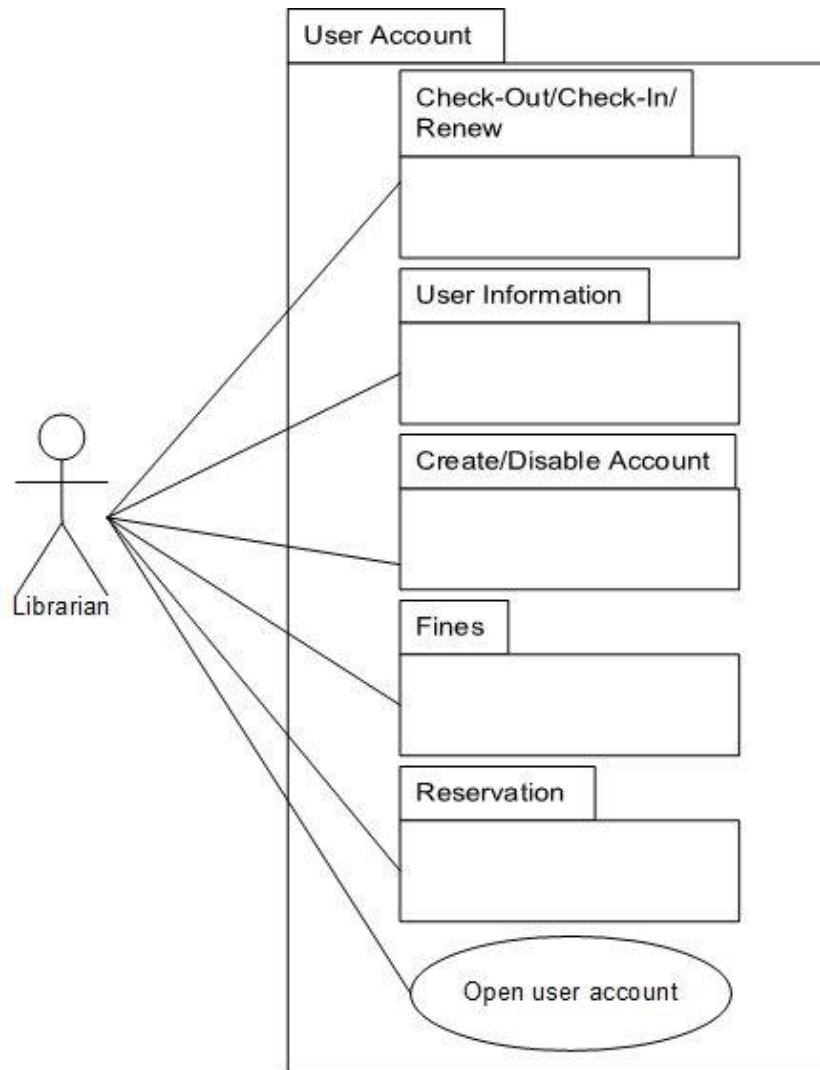
Client Program Use Cases

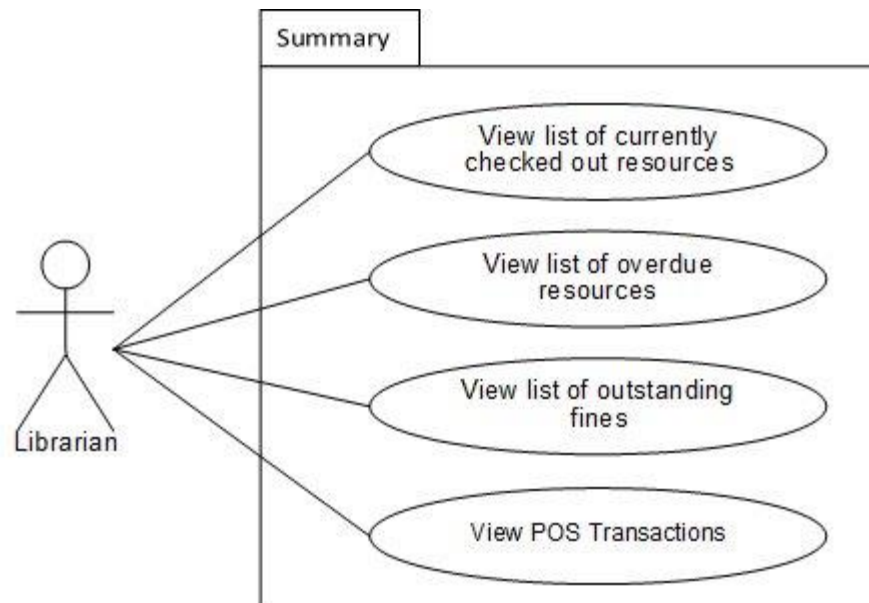
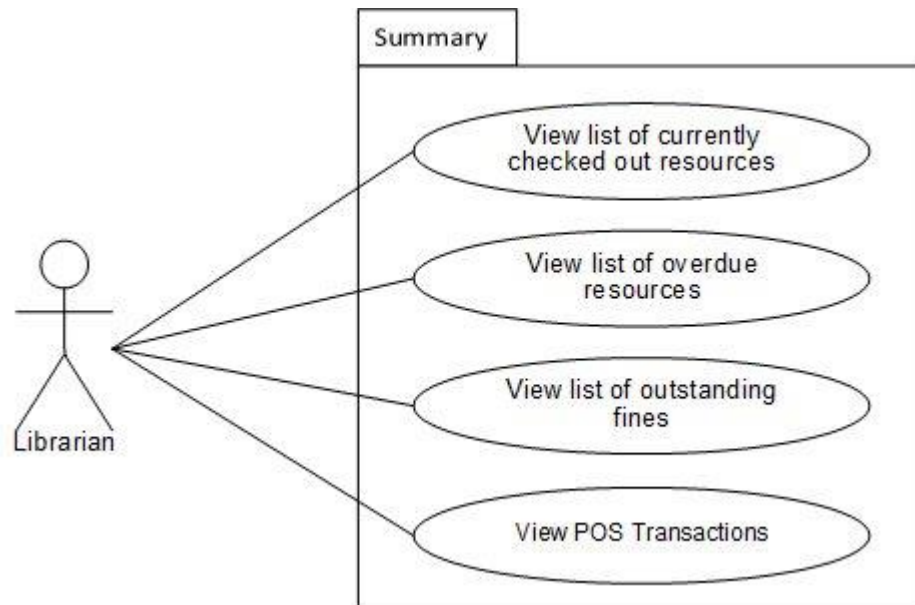


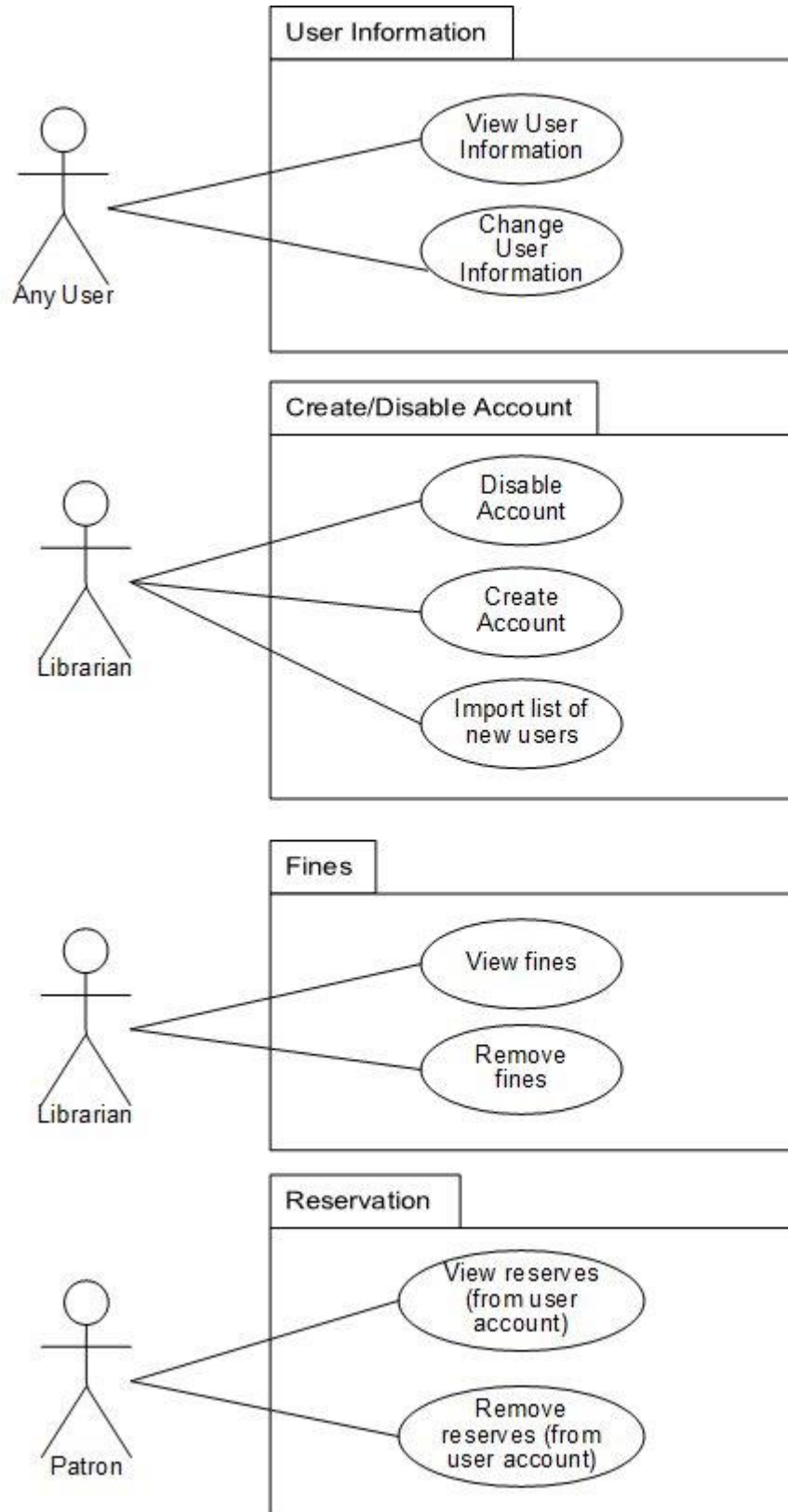




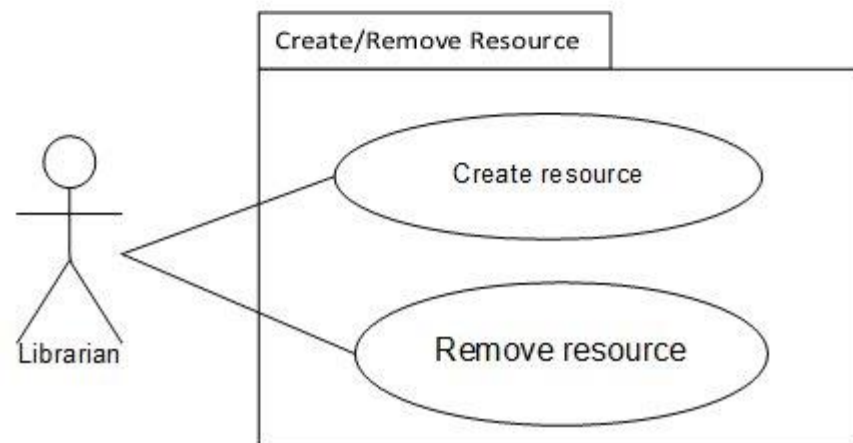
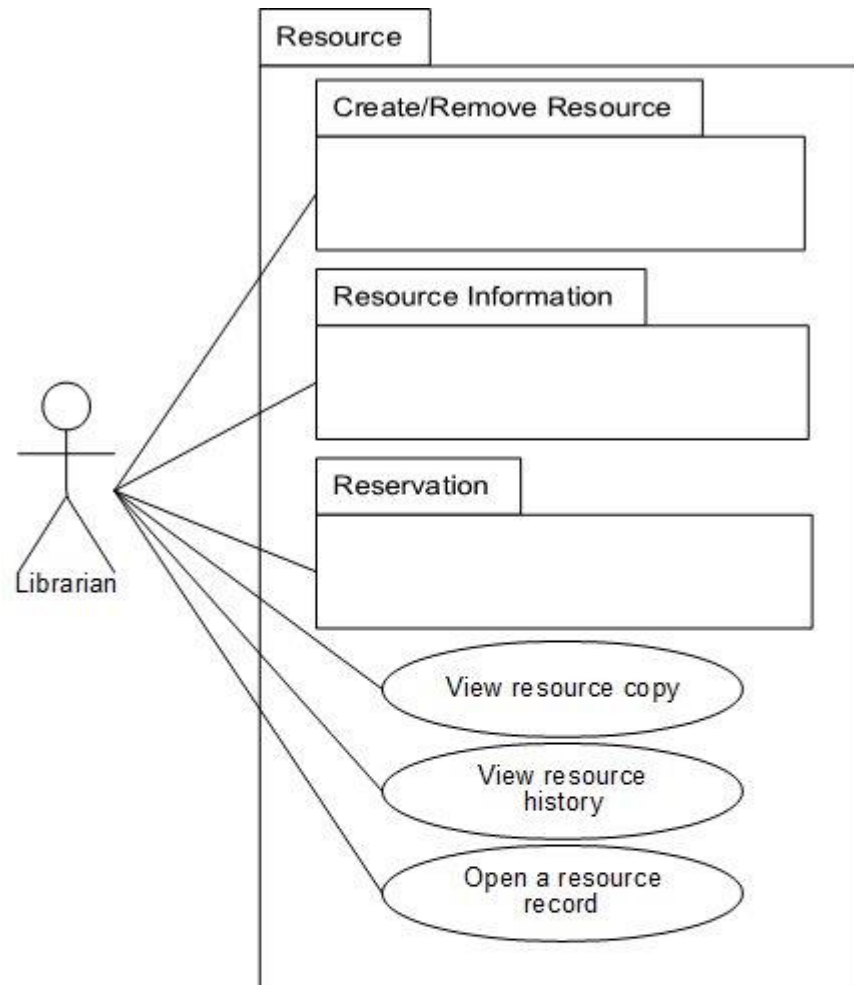


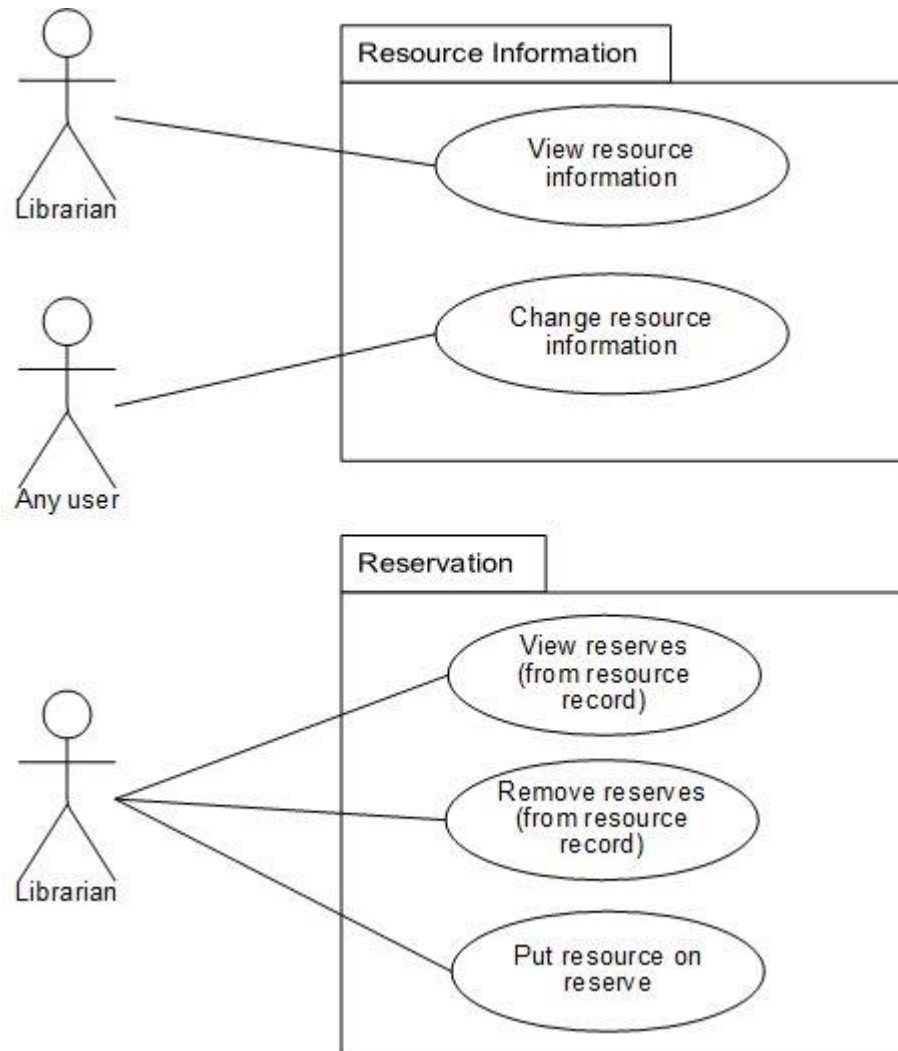




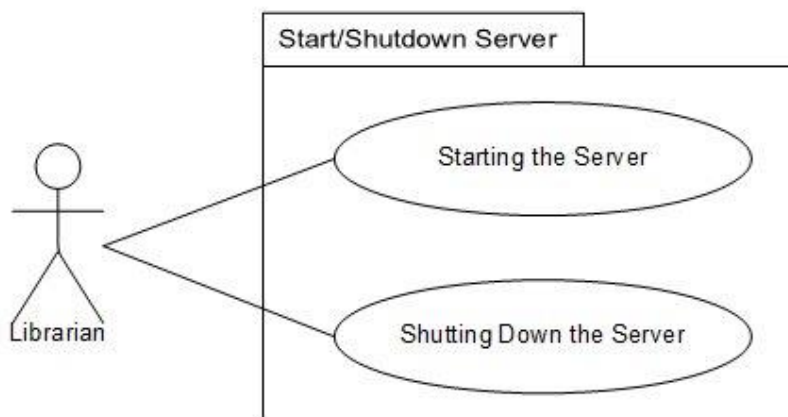
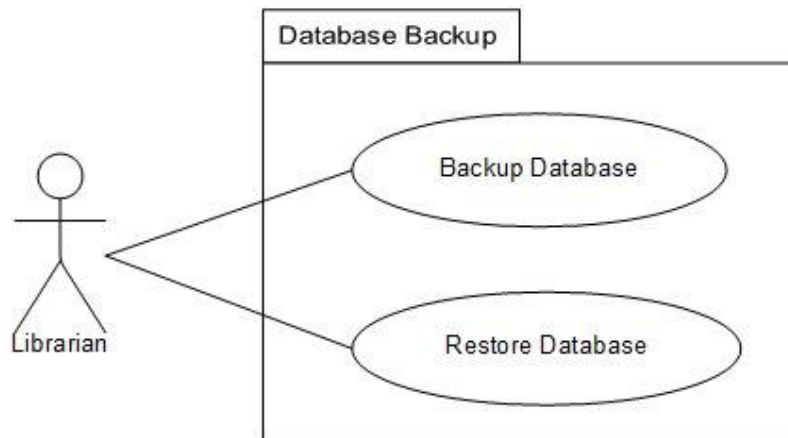
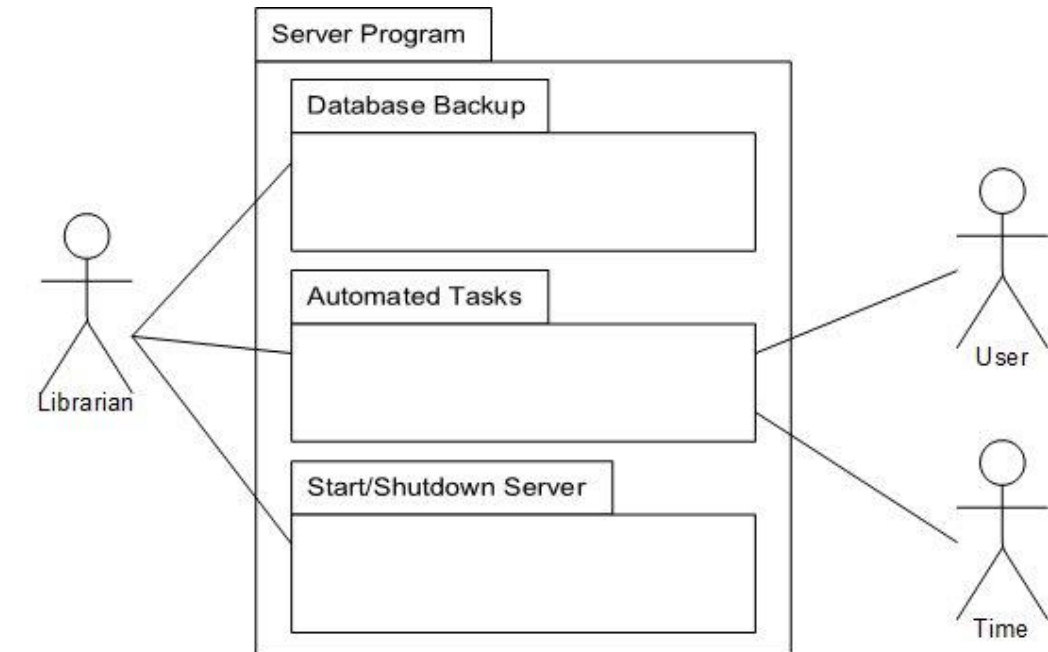


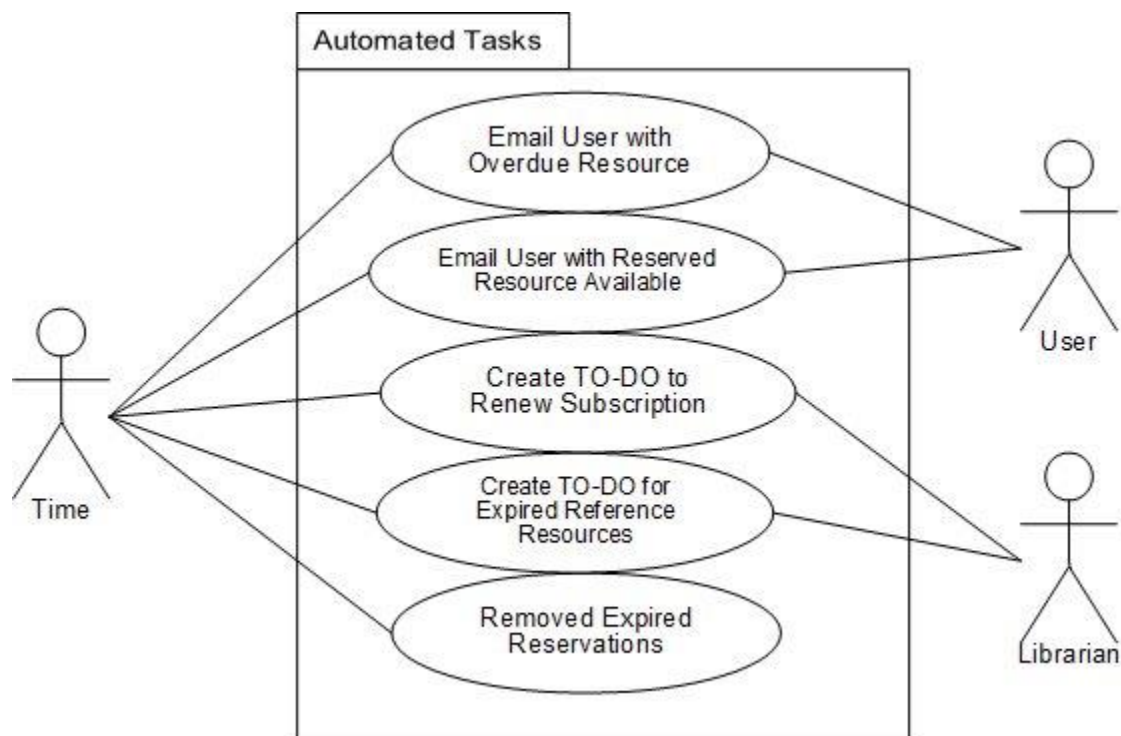
Reserves can be viewed/removed from either a user account (in terms of resource copies) or from a resource copy (in terms of user accounts)





Server Program Use Cases





Textual Use Cases

Resource Management

Name: **Create resource**

Participating Actors: Staff member

Entry Conditions:

- User is using the client program
- Client program is in Staff Mode
- The resource doesn't already exist.

Exit Conditions:

The resource is created in the system.

Flow Of Events:

1. The user chooses to add a new resource to the system.
 2. The system asks the user for information such as: resource type (book, music, etc.), title, etc.

Special Requirements: None

Name: **Create resource copy**

Participating Actors: Staff member

Entry Conditions:

- User is using the client program
- Client program is in Staff Mode
- The resource record is open

Exit Conditions:

The resource copy is added to the system.

Flow Of Events

1. The user clicks the "Resource copies" button or tab
 2. The system displays a list of the resource copies in the system.
3. The user clicks the "Add new resource copy" button
 4. The system ask the user whether the copy belongs to the library or if it is the property of a particular user (i.e.: a foreign resource). If it is the property of a particular user, then the staff member will select the user from a list.
 5. The system gives the user the resource copy ID of the new resource

Special Requirements: None

Name: **Change resource information**

Participating Actors: Staff member

Entry Condition:

If using client program, it is in Staff Mode

Exit Conditions:]

The user information is updated in the database

Flow of Events:

1. The resource record is opened, it **includes** Open Resource Record
2. User clicks on the "Resource information" button or tab
 3. The system displays the resource information
4. The user makes the appropriate changes and clicks the "save changes" button

Special Requirements: None

Name: **Create subscription**

Participating Actors: Staff member

Entry Conditions:

- Subscription does not already exist in the database
- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The new subscription is present in the user database

Flow of Events:

1. Staff members issues command "add new subscription"
2. Systems asks Staff member for additional info:
 - Subscription name
 - Copies available
 - Number of copies
 - Annual renew cost
 - Renew date

Special Requirements: None

Name: **Open a resource record**

Participating Actors: Anyone

Entry Conditions

User is using the client program

Exit Conditions

The resource record is displayed

Flow of Events

1. The user enters attributes of the resource they are looking for. This may also be the resource ID.
2. The system shows a list of matching resources.
3. The system displays an option for the user to search again.
4. The user selects the resource they are looking for.
5. The user opens the resource record

Special Requirements: None

Name: **Open a resource copy record**

Participating Actors: Staff member

Entry Conditions:

- User is using the client program
- Client program is in Staff Mode
- The resource record is open

Exit Conditions:

The resource copies record is opened

Flow of Events:

1. User clicks on the "Resource Copies" button or tab
2. The system displays a list of resource copies for that record
3. The user double clicks on one of the resource copies to display its record.

Special Requirements: None

Name: **Open a subscription record**

Participating Actors: Staff member

Entry Conditions:

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The subscription record is opened

Flow of Events:

1. The staff member enters attributes of the subscription they are looking for. This may also be the subscription ID.
2. The system shows a list of matching subscriptions.
3. The system displays an option for the user to search again.
4. The user selects the subscription they are looking for.
5. The user opens the subscription record

Special Requirements: None

Name: Put resource copy on reference**Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The resource has been placed on reference.

Flow Of Events:

1. User opens the resource copy record (see Open Resource Copy Record use case)
2. The user clicks the "Put on reference" button.
 3. The system prompts the user for additional information, including:
 - Class code (i.e.: CPSC 2301)
 - Class section (i.e.: 001)
 - Class name (i.e.: software engineering)
 - Instructor user name
 - How long to keep the resource on reference

Special Requirements:

If another user attempts to open the particular resource record, then they will either:

- Have to wait until the record has been saved and closed.
- Be able to open it but not edit it (i.e.: read only).

Name: Remove resource**Participating Actors:** Staff or Administrators**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The resource is removed along with each resource copy

Flow of Events:

1. User opens the resource record (see Open Resource Record use case)
2. The user clicks the "remove resource" button.

Special Requirements:

- The resource shouldn't be able be queried or updated while it is being removed by the user.
- A new edition of resource should exist for outdated resources.

Name: Remove resource copy**Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The resource copy is removed from the system.

Flow Of Events:

1. User opens the resource record (see Open Resource Record use case)
2. The user clicks the "Resource copies" button or tab
 3. The system displays a list of the resource copies in the system.
4. The user selects one or more resource copies and clicks the "Remove resource copy" button

Special Requirements: None

Name: Renew subscription**Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode
- Subscription is less than two months of expiring

Exit Conditions:

The selected subscription is renewed

Flow of Events:

1. User opens the subscription record (see Open Subscription Record use case)
2. The staff member clicks the "Renew Subscription" button.
3. The user is prompted for additional information (ie: expiration date).

Special Requirements: None**Name: Remove subscription****Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The subscription is removed from the database

Flow of Events:

1. User opens the subscription record (see Open Subscription Record use case)
2. Staff member issues command "cancel subscription"

Special Requirements: None**Name: Open a subscription record****Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The subscription record is opened

Flow of Events:

1. The staff member enters attributes of the subscription they are looking for. This may also be the subscription ID.
 2. The system shows a list of matching subscriptions.
 3. The system displays an option for the user to search again.
4. The user selects the subscription they are looking for.
5. The user opens the subscription record

Special Requirements: None

Name: Take resource copy off reference

Participating Actors: Staff member

Entry Conditions:

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The resource has been taken off reserve.

Flow of Events:

1. User opens the resource record (see Open Resource Record use case)
2. The user clicks the button "Remove from Reference"

Special Requirements: None

Name: View Resource Copies

Participating Actors: Staff member

Entry Conditions:

- The resource exists in the system
- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The resource's copies are shown

Flow of Events:

1. The user opens the resource record (see Open Resource Record use case)
2. User clicks on the "Resource Copies" button or tab
3. The system displays a list of the copies of the resource

Special Requirements: None

Name: View Resource Copy History

Participating Actors: Staff member

Entry Conditions:

- User is using the client program
- Client program is in Staff Mode
- The resource copy record is open

Exit Conditions: The resource copy's history is shown

Flow of Events:

1. The user opens the resource copy record (see Open Resource Copy Record use case)
2. User clicks on the "History" button or tab

Special Requirements: None

Name: View resource copy information**Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode
- The resource copy record is open

Exit Conditions:

The resource copy's information is shown

Flow of Events:

1. User clicks on the "Information" button or tab

Special Requirements: None**Name: View resource history****Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The resource's history is shown

Flow of Events:

1. The user opens the resource record (see Open Resource Record use case)
2. User clicks on the "History" button or tab
3. The system displays the check-out/in history of the resource

Special Requirements: None**Name: View resource information****Participating Actors:** Any user**Entry Conditions:**

User selects a resource

Exit Conditions:

The resource's information is shown

Flow of Events:

1. The user opens the resource record (see Open Resource Record use case)
2. User clicks on the "Information" button or tab

Special Requirements: None**Name: Send email to users with overdue resources****Participating Actors:** User**Entry Conditions:**

The resource borrower has an overdue resource.

Exit Conditions:

The resource borrower is notified via email of their overdue resources.

Flow of Events:

1. The server program determines a resource is overdue and the borrower should be notified.
2. The system sends an email to the user as a reminder to return the resource.
3. The user receives the email notification.

Special Requirements: None

Name: Send email to user when reserved resource is available

Participating Actors: User

Entry Conditions:

Resource has been returned

Queue has a valid reservation

Exit Conditions:

Email sent to next user

Flow Of Events:

1. The server program determines a resource is available and that the next user in the reserved queue has not yet been notified.

2. The system sends an email to the user to the user.

Special Requirements: None

Circulation of Resources

Name: Signing into the client

Participation Actors: User

Entry Conditions:

It **extends** "Wrong Credentials" when username and password do not match.

Exit Conditions:

The user is signed in

Flow of Events:

1. The user runs the client (see Starting Client case)

2. The user clicks the login button.

3. The user is prompted for their login information.

4. The user enters their login information

5. If the user enters the wrong credentials, he cannot log-in (see Wrong Credentials use case)

Special Requirements: None

Name: Signing out of the client

Participating Actors: User

Entry Conditions:

The client is running and someone is signed in.

Exit Conditions:

User is signed out

Flow of Events:

1. User clicks the "sign out" button

2. The systems signs the user out of the client

Special Requirements: None

Name: Check In Resource (from user account)**Participating Actors:** Staff member**Entry Conditions:**

Using client program

Client program is in Staff Mode

Exit Conditions:

Check in information is present in the database

Flow of Events

1. The user account is opened (see Open User Account use case)
2. User clicks on currently “checked out” button or tab
 3. The client displays a list of currently checked out items for the account
4. The user selects one or more resources and clicks “check in”

Special Requirements:

Only a staff member can check in a resource for a student, a faculty member or other staff member

Name: Check In Resource (no patron)**Participating Actors:** Staff member**Entry Conditions:**

Using client program

Client program is in Staff Mode

Exit Conditions:

Check in information is present in the database

Flow of Events:

1. staff member clicks on “check in (no patron)” button
2. Staff member enters resource copy ID
 3. client displays information about resource (book info, check out date, return dates)

Special Requirements:

Only a staff member can check in a resource for a student, a faculty member or other staff member

Name: Check Out Resource**Participating Actors:** Staff member (Librarian)**Entry Conditions:**

Using client program

Client program is in Staff Mode

It extends Refuse loan when: 1. The resource has been requested by another patron 2. Patron has outstanding fines equals to \$5.00**Exit Conditions:**

Librarian has checked out the resource for the patron

Flow of Events:

1. The user account is opened (see Open User Account use case)
2. Librarian clicks on "Checkout" button or tab.
3. Librarian enters the resource ID.
 4. The system checks whether the patron has the maximum number resources on loan, whether the patron has any outstanding fines, whether the resource is on reserve for another patron, and whether the patron has any overdue resources. If any of these conditions are true, then the patron is refused the loan (see the Refuse Loan use case). Otherwise, the loan is approved.

Special Requirements: None**Name: View checked out resources****Participating Actors:** User**Entry Conditions:** The user exists in the system

User is using the client program

Exit Conditions:

The user's list of checked out resources is shown

Flow of Events:

1. The user account is opened (see Open User Account use case)
2. User clicks on the "checked-out resources" button or tab

Special Requirements: None**Name: View list of currently checked out resources****Participating Actors:** Staff Member**Entry Conditions:**

User is using the client program

Client program is in Staff Mode

Exit Conditions:

The list of currently checked out resources is displayed

Flow of Events:

1. User clicks on the "Summary" button
 2. The system presents several summaries to choose from.
3. The user clicks on "View list of currently checked out resources"

Special Requirements: None

Name: Refuse loan**Participating Actors:** Staff member**Entry Conditions:**

This use case extends the Check Out Resource, Renew Resource (from user account), and Renew Resource (no patron) use cases.

Exit Conditions:

The loan is not approved. No changes are made to the database.

Flow Of Events:

1. The system notifies the staff member that the loan has not been approved.
2. The user clicks OK and is returned to the previous screen.

Special Requirements: None**Name: View List of overdue resources****Participating Actors:** Staff Member**Entry Conditions:**

User is using the client program

Client program is in Staff Mode

Exit Conditions:

The list of overdue resources is displayed

Flow of Events:

1. User clicks on the "Summary" button
 2. The system presents several summaries to choose from.
3. The user clicks on "View List of Overdue Resources"

Special Requirements: None**Name: Put resource on reserve****Participating Actors:** User**Entry Conditions:**

- Resource is not available (checked out)
- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The resource is placed on reserve for the Patron

Flow Of Events:

1. The resource record has been opened (see Open Resource Record use case)
2. User clicks the "put on reserve" button
 3. If in Staff mode, the system prompts for a library ID.

Special Requirements: None

Name: View Reservations (From Resource Record)**Participating Actors:** Staff member**Entry Conditions:**

User is using the client program

Client program is in Staff Mode

Exit Conditions:

The list of reserved resources is shown as a list of users, sorted in order of first come first served

Flow Of Events:

1. The user opens the resource record (see Open Resource Record use case)
2. The user clicks the "Reserve" button or tab

Special Requirements: None**Name: View reservations (from user account)****Participating Actors:** Any user**Entry Conditions:**

User is using the client program

Exit Conditions:

The list of reserved resources is shown

Flow of Events:

1. The user account is opened (see Open User Account use case)
2. The user clicks the "Reserve" button or tab

Special Requirements: None**Name: Remove reservation (from resource record)****Actors:** Library Staff**Entry Conditions:**

- User is using the client program

- Client program is in Staff Mode

Exit Conditions:

The resource copy is no longer on reserve for the given patron.

Flow Of Events

1. The staff member brings up the page for the resource (see Open Resource Record use case)
2. The staff member clicks on a button labeled "List of reserves".
3. The staff member selects the user or users to remove from the queue.
4. The staff member clicks a button labeled "Remove".

Special Requirements: None

Name: Remove reservation (from user account)**Participating Actors:** Any user**Entry Conditions:**

User is using the client program

Exit Conditions:

The selected user's reserves are removed from the system

Flow of Events:

1. The user account is opened (see Open User Account use case)
2. User clicks on the "Reserves" button or tab
 3. A list of the user's reserved resources is displayed
4. The user selects one or more resources and clicks the "remove" button

Special Requirements: None**Name: Remove expired reservations****Participating Actors:** Automated Server Function**Entry Conditions:**

- Patron or Librarian made a reservation
- System sent notification that resource was available
- Reservation expired

Exit Conditions:

Resource's reservation queue is updated

Flow of Events:

1. System verifies unclaimed reservation
2. System removes patron from the queue and promotes next patron to the top of the queue
- 3) System emails message to the next patron in the queue (if any)

Special Requirements: None**Name: Renew resource (from user account)****Participating Actors:** Staff member**Entry Conditions:**

- Using client program
- Client program is in Staff Mode
- It **extends** Refuse loan when:
 1. Resource has been requested by other patron,
 2. Patron has outstanding unpaid fines equals to \$5.00

Exit Conditions:

Renew details is present in the database

Flow of Events:

1. The user account is opened (see Open User Account use case)
2. staff member clicks on "checked out" button or tab
 3. System displays list of currently checked out items for the account
4. staff member selects one or more resources and clicks "renew" button
 5. The system checks whether the patron has any outstanding fines, whether the resource is on reserve for another patron, and whether the patron has any overdue resources. If any of these conditions are true, then the patron is refused the loan (see the Refuse Loan use case). Otherwise, the loan is approved.

Special Requirements: None

Name: Renew resource (no patron)**Participating Actors:** Staff member**Entry Conditions:**

- Using client program
- Client program is in Staff Mode
- It **extends** Refuse loan when:
 1. Resource has been requested by another patron
 2. Outstanding fines are not equal to \$5.00

Exit Conditions:

Renew details is present in the database

Flow of Events:

1. staff member clicks on "renew (no patron)" button
2. Staff member enters resource copy ID
 3. The system checks whether the patron has any outstanding fines, whether the resource is on reserve for another patron, and whether the patron has any overdue resources. If any of these conditions are true, then the patron is refused the loan (see the Refuse Loan use case). Otherwise, the loan is approved.

Special Requirements: None**Name: View resources to be shelved****Participating Actors:** Staff member**Entry Conditions:**

There are resources that have been checked in, but not yet shelved

Exit Conditions:

List of resources is displayed on screen.

Flow of Events:

1. Staff member logs in, selects the "Shelving" option
 2. The system displays a list of resources to be shelved.

Special Requirements: None**Name: Remove fines****Participating Actors:** Staff member**Entry Conditions:**

- User has outstanding fees to pay
- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

User's fines are removed from their account

Flow of Events:

1. The user account is opened (see Open User Account use case)
2. Staff member clicks on "Fines" button or tab
 3. The system displays a list of the fines for that user's account
4. The staff member selects one or more fines and clicks "remove"
5. Staff member clicks the "print" button to print a receipt

Special Requirements: None

Name: **View fines**

Participating Actors: Any user

Entry Conditions:

User is using the client program

Exit Conditions:

The user database is updated with the new information

Flow of Events:

1. The user account is opened (see Open User Account use case)
2. User clicks on the "Fines" button or tab

Special Requirements: None

Name: **View List of outstanding fines**

Participating Actors: Staff Member

Entry Conditions:

User is using the client program

Client program is in Staff Mode

Exit Conditions:

The list of outstanding fines is displayed

Flow of Events:

1. User clicks on the "Summary" button
 2. The system presents several summaries to choose from.
3. The user clicks on "View List of Outstanding Fines"

Special Requirements: None

Patron Management

Name: Create account

Participating Actors: Staff member

Entry Conditions:

User is using the client program

Client program is in Staff Mode

Exit Conditions:

The new user account is present in the user database

User is assigned library number

Flow Of Events:

1. Staff member issues command "add new user".
2. Staff member is prompted for additional info:
 - First and last name
 - Username
 - Email
 - Password (and password confirmation)
 - User type (Administrator, staff member, faculty, student)

Special Requirements:

The staff member must have sufficient authority to add a user

Name: Disable account

Participating Actors: Administrator, Staff Member

Entry Conditions:

User is using the client program

Client program is in Staff Mode

Exit Condition:

User account is disabled

Flow of Events:

1. The user account to be disabled is opened (see Open User Account use case)
2. Actor Clicks disable user button
 3. The system verifies that the actor has permission to disable the account (only an Administrator or Staff Members can disable a user and faculty members, and only an administrator can disable staff). If not, then the request to disable the account is rejected (see Refuse account disable use case). Otherwise, the account is disabled.

Special Requirements: None

Name: View user information

Participating Actors: Any user

Entry Conditions:

User owns account

Staff has selected a user

Exit Conditions:

The user information is shown

Flow of Events:

1. The user account is opened (see Open User Account use case)
2. User clicks on the "User information" button or tab

Special Requirements: None

Name: Change user information**Participating Actors:** Any user**Entry Conditions:**

User owns account

Staff has selected a user

Exit Conditions:

The user information is updated in the database

Flow of Events:

1. The user account is opened (see Open User Account use case)
2. User clicks on the "User information" button or tab
 3. The user information is displayed
4. The user makes the appropriate changes and clicks the "save changes" button

Special Requirements: None**Name: View user history****Participating Actors:** Any user**Entry Conditions:**

- User owns the account

- Staff has selected a user

Exit Conditions:

The user's history is shown

Flow of Events:

1. The user account is opened (see Open User Account use case)
2. User clicks on the "History" button or tab

Special Requirements: None**Name: Import list of new users****Participating Actors:** Administrator**Entry Conditions:**

- The list of new users is of csv format with specific attributes and file layout.
- The administrator is logged onto their user account.
- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

All of the users are added to the system and given appropriate access.

Flow of Events:

1. Administrator chooses to import a list of new users.
 2. The system reminds the user of the required format for the csv file.
 3. The system prompts the user for the file to import.
4. The administrator locates and choose the file to import.
 5. The system checks the file to ensure correct usage.
 6. The system adds the users.
7. The administrator confirms that the new users have been added.

Special Requirements: None

Name: **Open user account**

Participating Actors: User

Entry Conditions:

User is using the client program

Exit Conditions:

The user account is opened

Flow of Events:

1a. If in Patron mode, then the user clicks the “my account” button.

1b. If in Staff mode, the user clicks the “search for user” button and enters attributes of the user they are looking for (this may also be the user’s library number).

2b. The user selects the user they are looking for.

Special Requirements: None

Program Administration

Name: **Starting Server**

Participating Actors: Staff Member

Entry Conditions:

The server is not running.

Exit Conditions:

The server is started

Flow of Events:

1. Staff member launches server program.

2. The system runs the server.

Special Requirements: None

Name: **Shutting Down Server**

Participating Actors:

Staff member

Entry Conditions:

The server is running

Exit Conditions:

Server is shut down

Flow of Events:

1. Staff member closes the server program.

2. The system shuts down the server and saves current state.

Special Requirements: None

Name: Create backup of database**Participating Actors:** Staff Member**Entry Conditions:**

No terminals are logged in

Exit Conditions:

The database is duplicated on the server

Flow of Events:

1. Staff Member logs in to server, selects the "database backup" option.
2. The contents of the databases are duplicated and stored. Once this is finished, the oldest backup file is deleted

Special Requirements: None**Name: Restore backup of database****Participating Actors:** Staff Member**Entry Conditions:**

No terminals are logged in

Exit Conditions:

The backup is restored

Flow of Events:

1. open user account
2. Staff Member logs in to server, selects the "database restore" option.
3. Server presents a list of backups to choose from
4. User selects backup and clicks "restore" button

Special Requirements: None**Name: Starting client****Participating Actors:** Staff Member**Entry Conditions:**

The client program is not already running

Exit Conditions:

The client program is running

Flow of Events:

1. The user launches the program

Special Requirements: None**Name: Shutting Down Client****Participating Actors:** Staff member**Entry Conditions:**

The client program is already running

Exit Conditions:

The client program is not running

Flow of Events:

1. The user closes the program

Special Requirements: None

Name: View POS transactions**Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The list of transactions for the desired POS is displayed.

Flow of Events:

1. User clicks on the "Summary" button
 2. The system presents several summaries to choose from.
3. The user clicks on "View POS transactions"
 4. The user is asked for the POS to view
5. The user clicks the desired POS

Special Requirements: None**Name: Create TO-DO****Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The TO-DO is created.

Flow of Events:

1. The user clicks on the TO-DO button.
 2. The system displays a list of active TO-DOs
3. The user clicks on the "Create TO-DO" button
 4. The user is prompted for additional information.

Special Requirements: None**Name: Create TO-DO for expired reference resources****Participating Actors:** Automated Server Function**Entry Conditions:**

A resource has been automatically taken off reference (time expired) and must be removed from the reservation section

Exit Conditions:

A TO-DO is created to renew the expired reference resource.

Flow of Events:

1. The system periodically checks for expired reference material
2. The system creates a new TO-DO for each item found.

Special Requirements: None

Name: Create TO-DO to renew a subscription**Participating Actors:** Automated Server Function**Entry Conditions:**

A subscription must exist on the database and its renew date must be expired or expiring soon.

Exit Conditions:

A TO-DO is created to renew the subscription

Flow of Events:

1. System periodically checks for subscription renewals.
2. When renew date hits a TO-DO is created to renew all subscriptions for that day/year

Special Requirements: None**Name: Open TO-DO****Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The TO-DO is opened.

Flow of Events:

1. The user clicks on the TO-DO button.
 2. The system displays a list of active TO-DOs
3. The user double-clicks on the appropriate TO-DO to open it.

Special Requirements: None**Name: Remove TO-DO****Participating Actors:** Staff member**Entry Conditions:**

- User is using the client program
- Client program is in Staff Mode

Exit Conditions:

The TO-DO is removed.

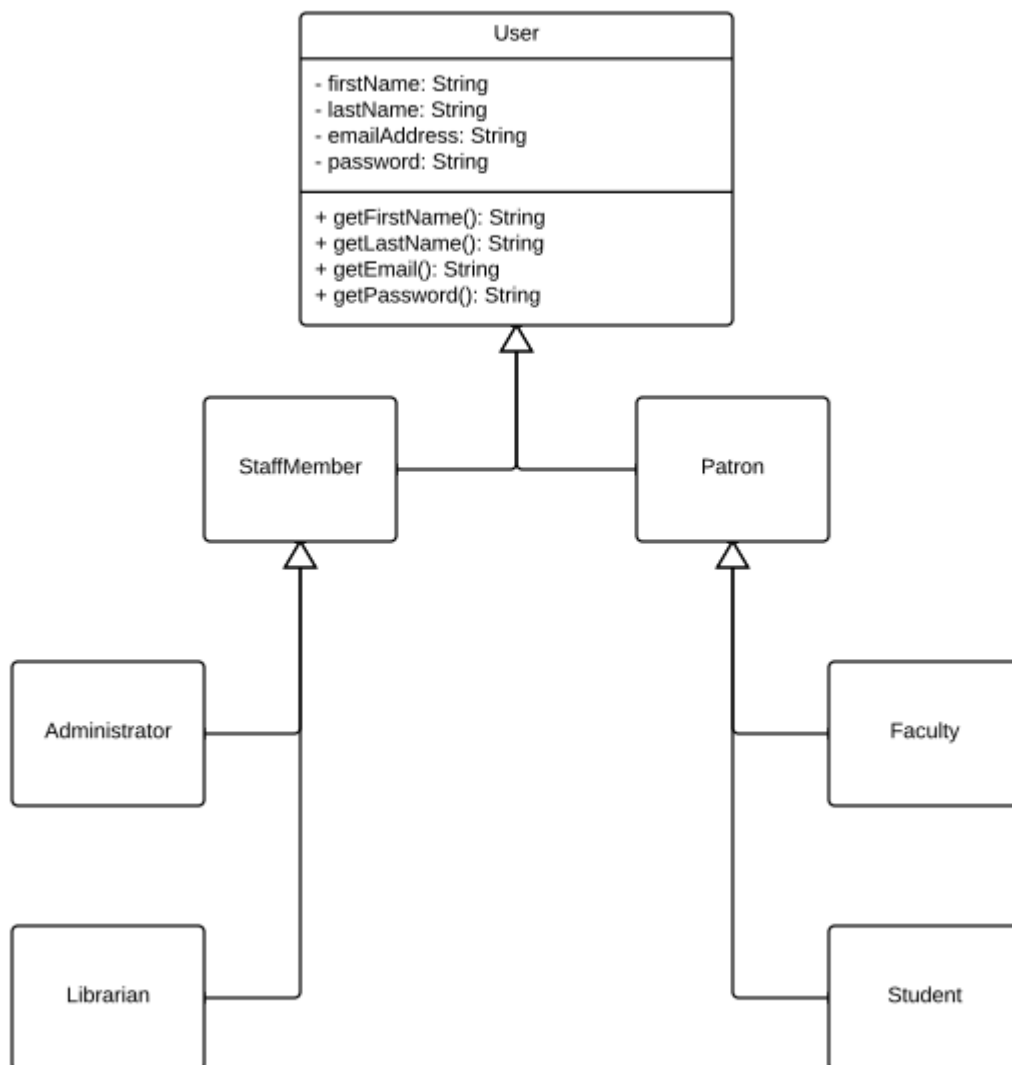
Flow of Events:

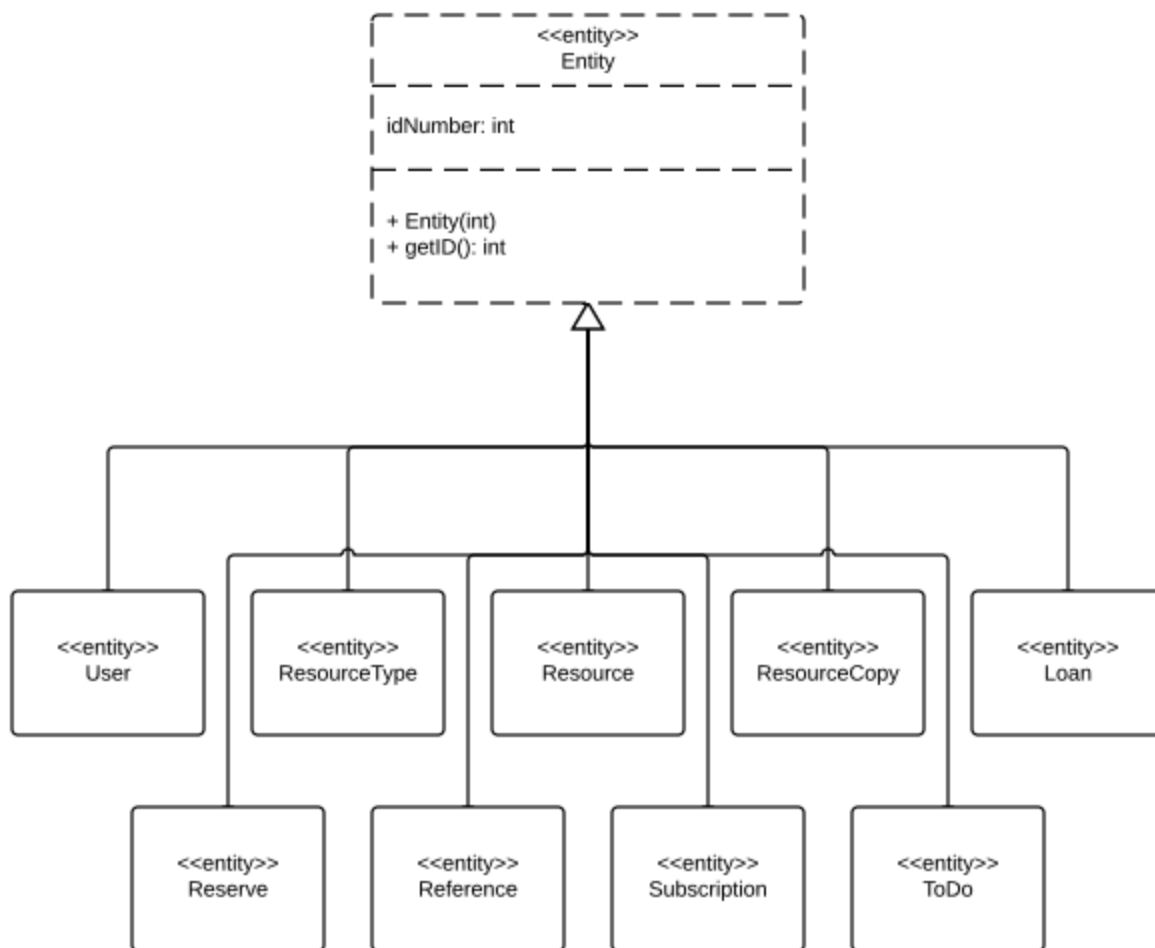
1. The user clicks on the TO-DO button.
 2. The system displays a list of active TO-DOs
3. The user selects one or more TO-DOs and clicks the "Remove" button.

Special Requirements: None

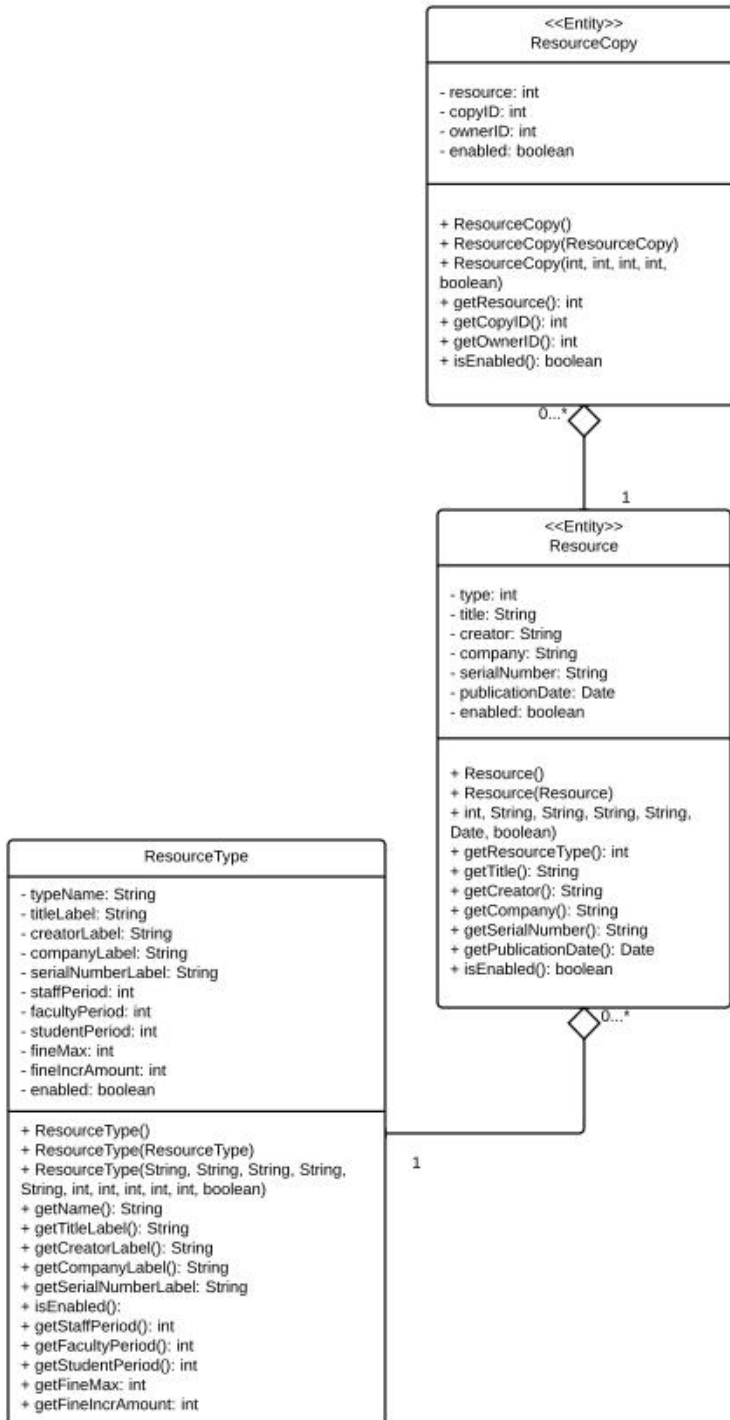
3.4.3. Object model

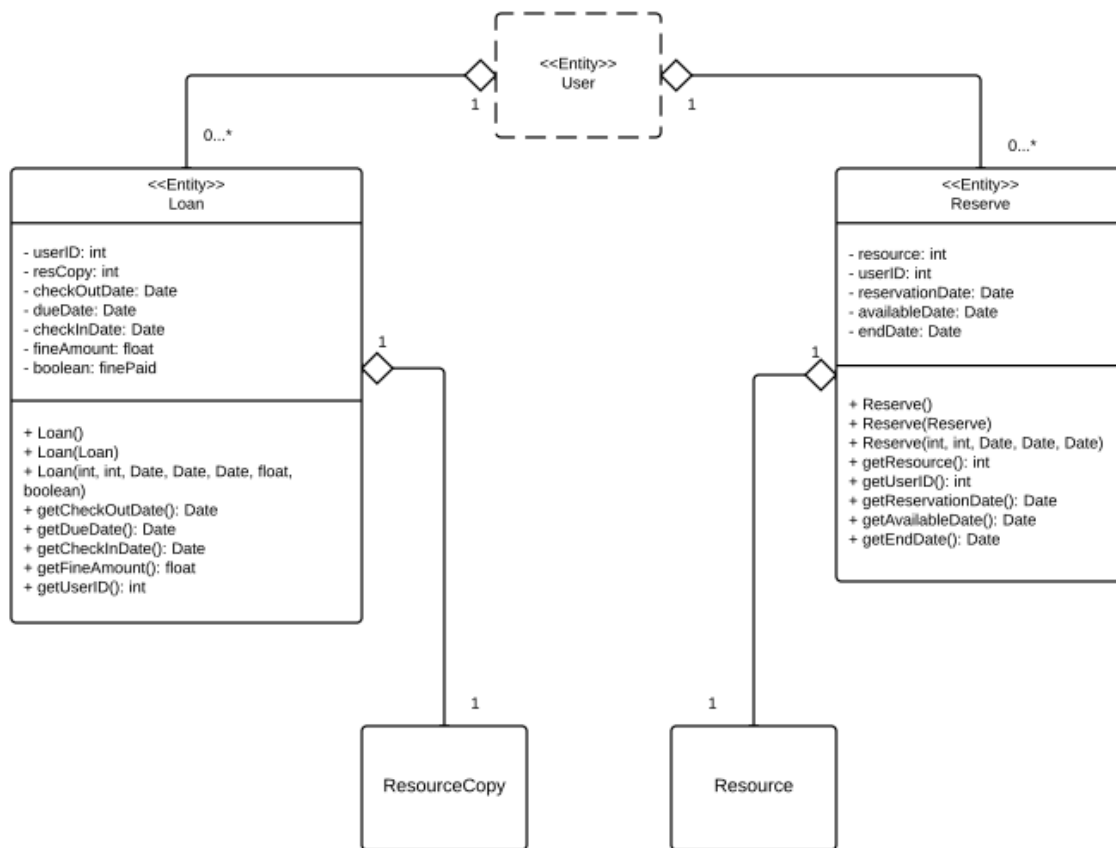
User Objects

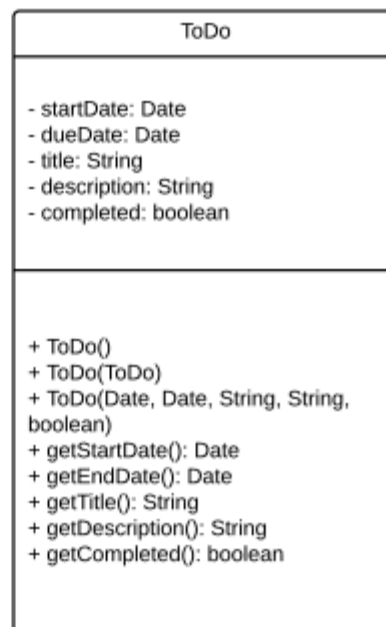
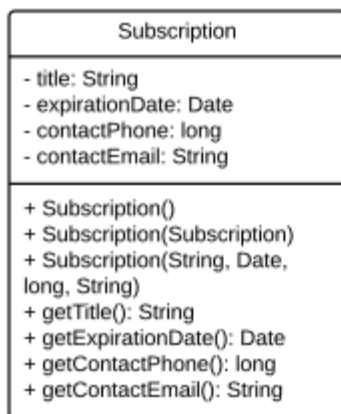
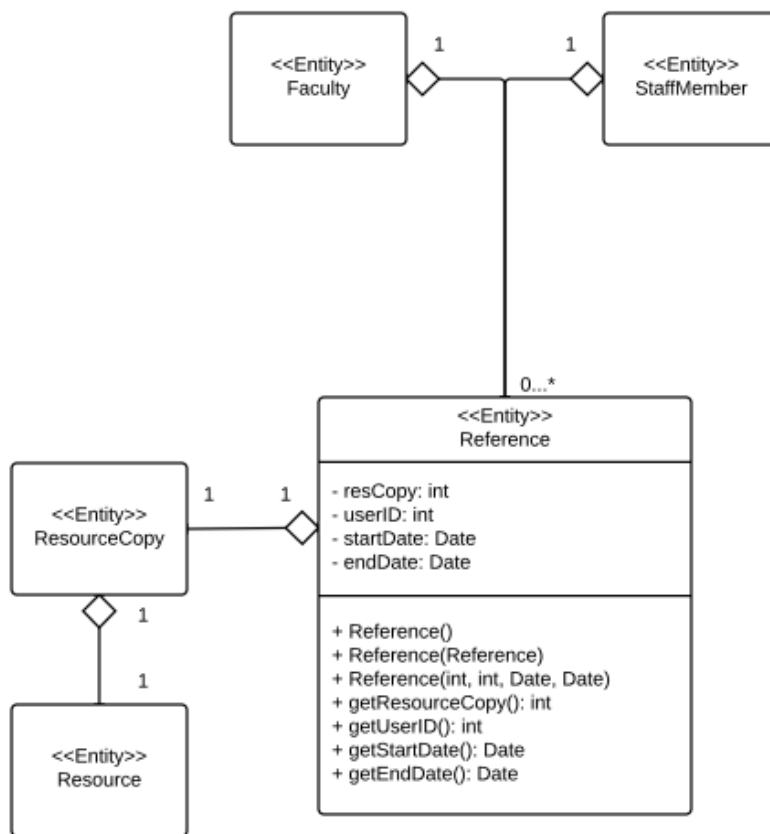




Application Structure Objects





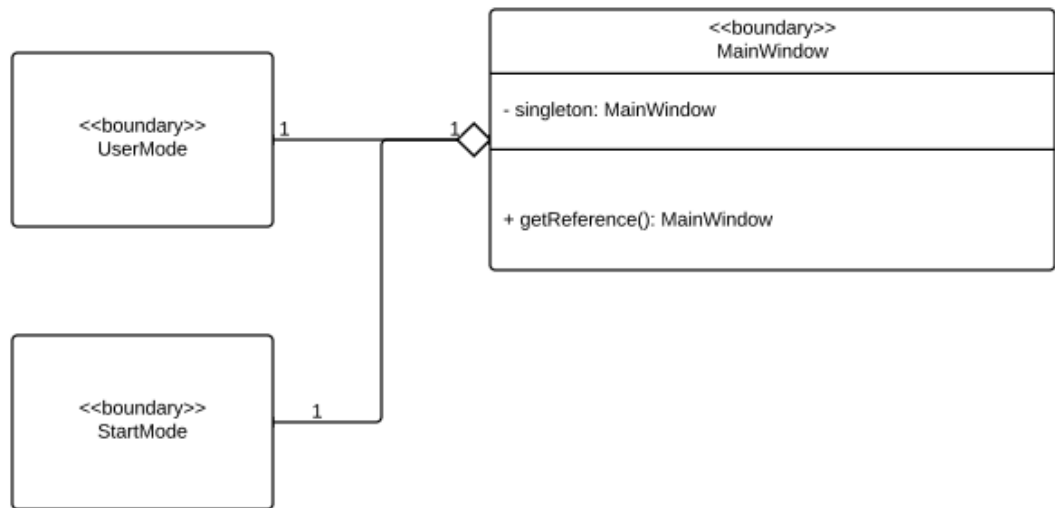


3.4.4. Class Diagrams

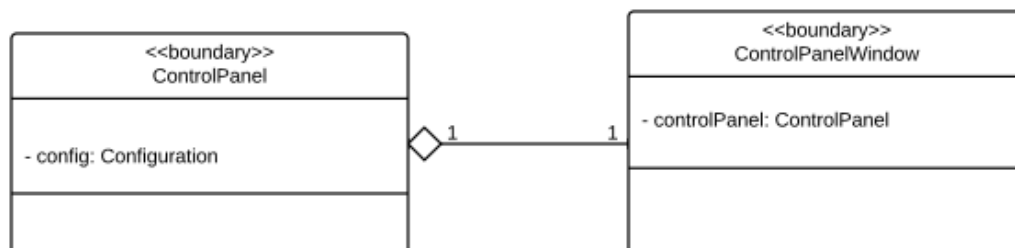
Package: client.userinterface.datatypes



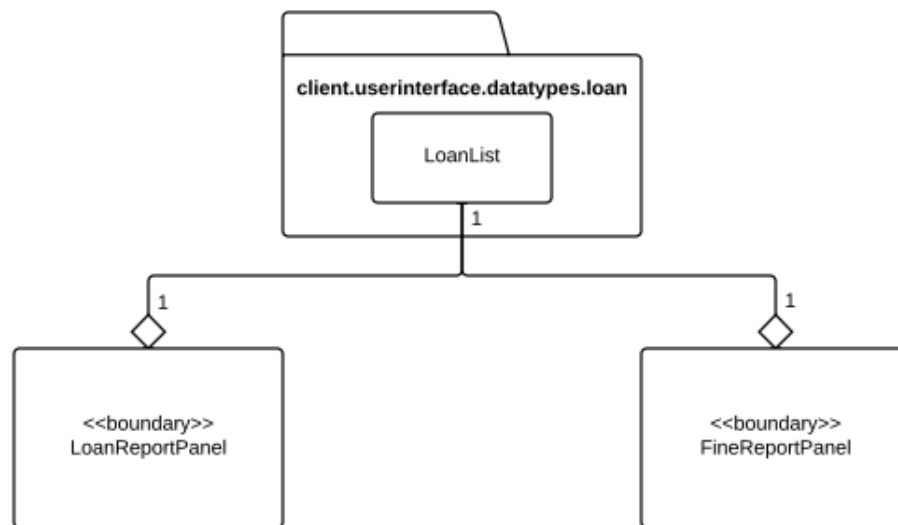
Package: client.userinterface.mainwindow



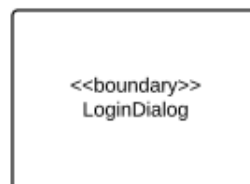
Package: client.userinterface.controlpanel



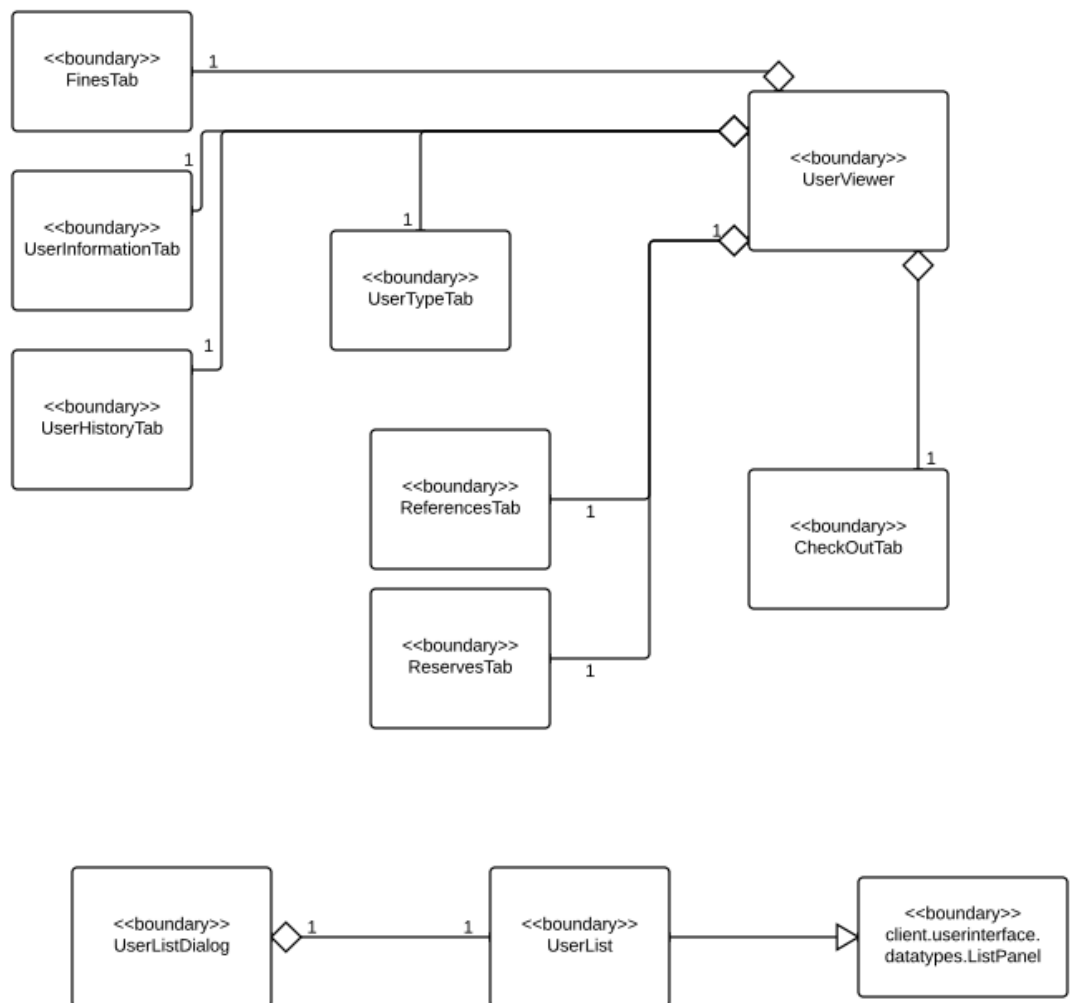
Package: client.userinterface.datatypes.report



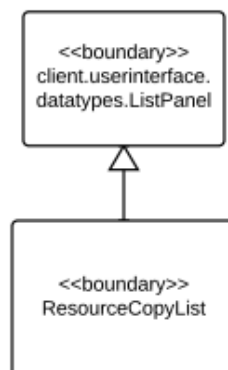
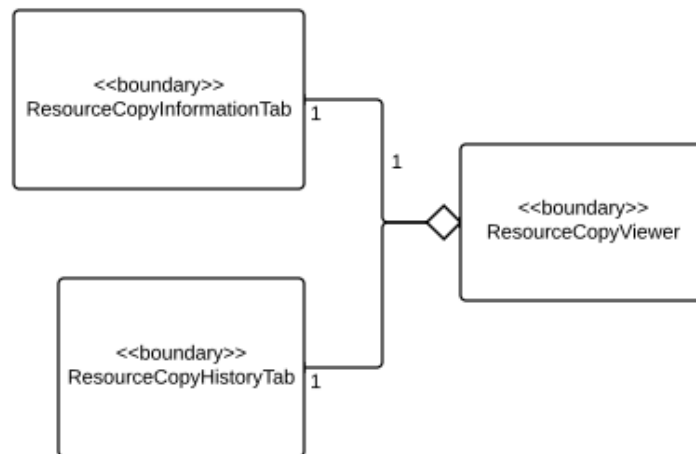
Package: client.userinterface.login



Package: client.userinterface.datatypes.user



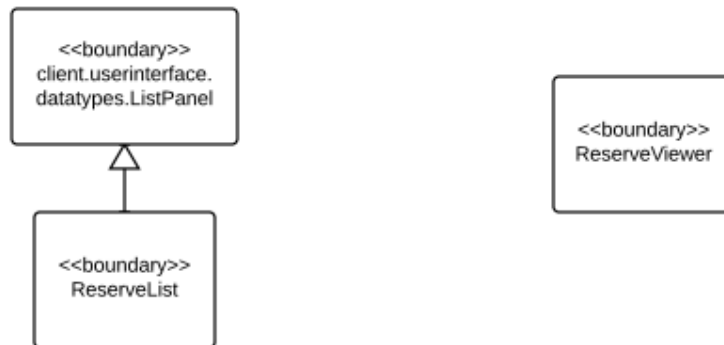
Package: client.userinterface.datatypes.resourcecopy



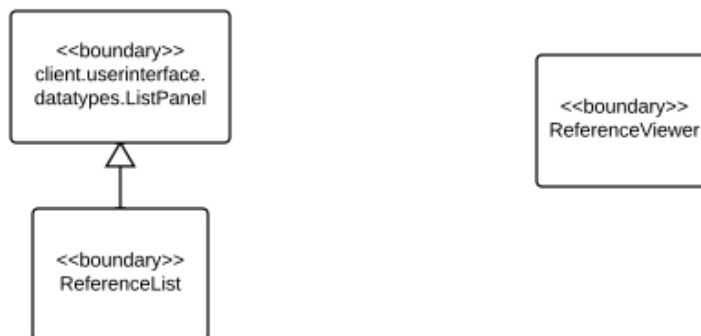
Package: client.userinterface.datatypes.loan



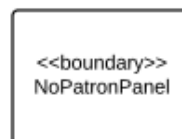
Package: client.userinterface.datatypes.reserve



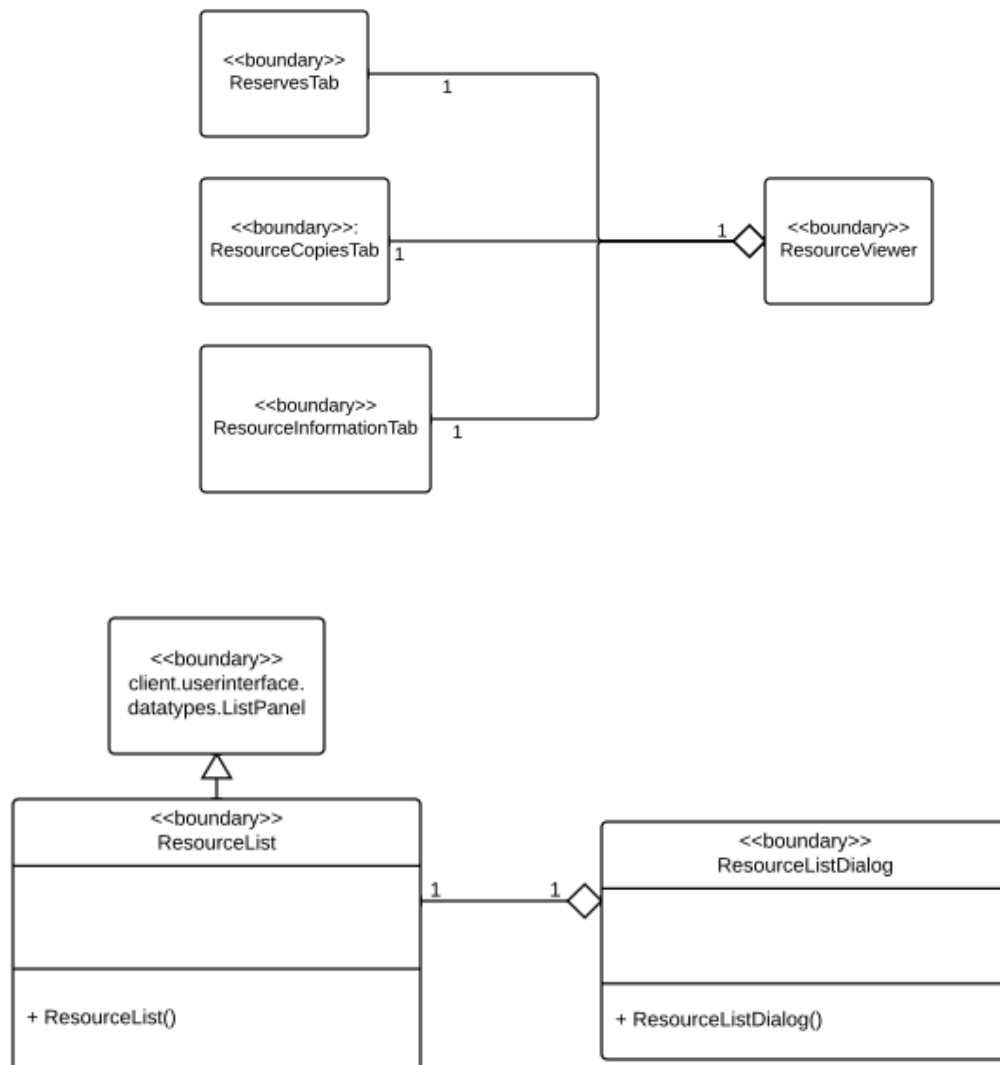
Package: client.userinterface.datatypes.reference



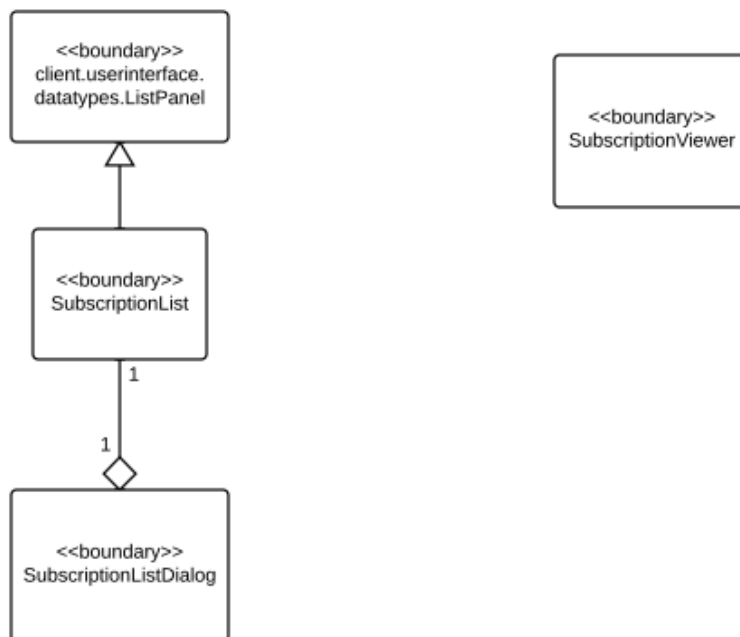
Package: client.userinterface.datatypes.nopatron



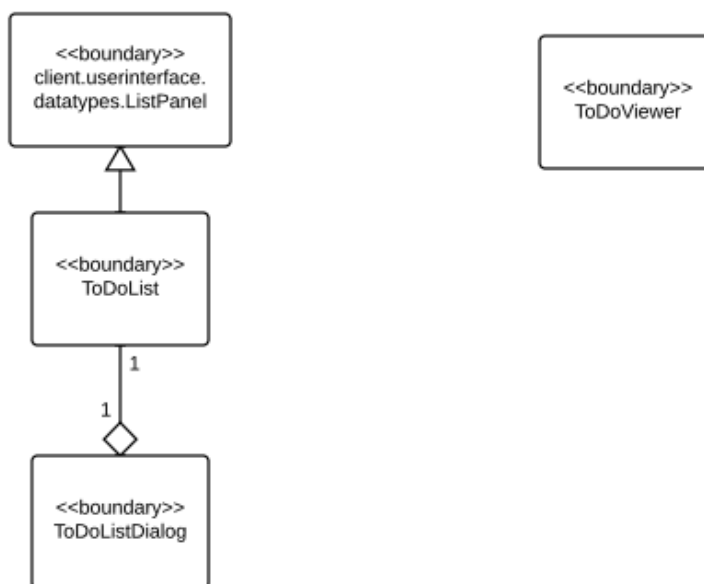
Package: client.userinterface.datatypes.resource



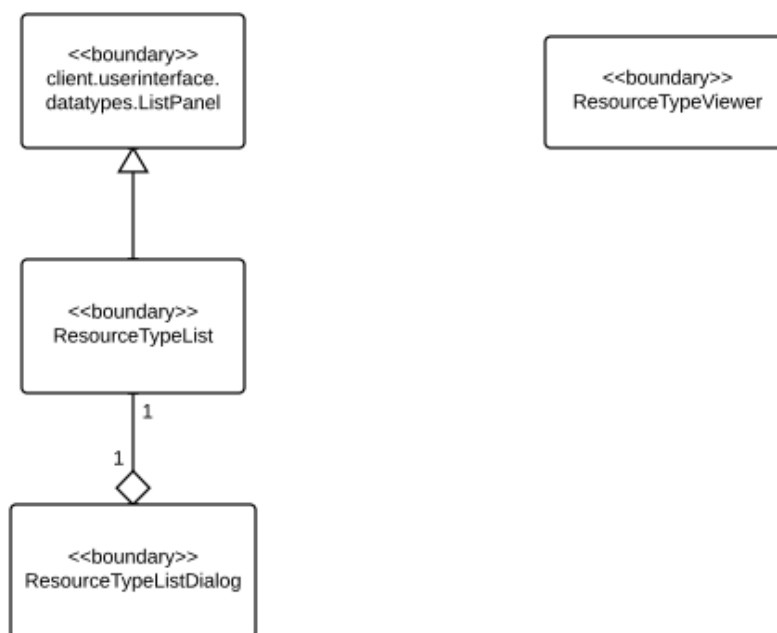
Package: client.userinterface.datatypes.subscription



Package: client.userinterface.datatypes.todo



Package: client.userinterface.datatypes.resourcetype

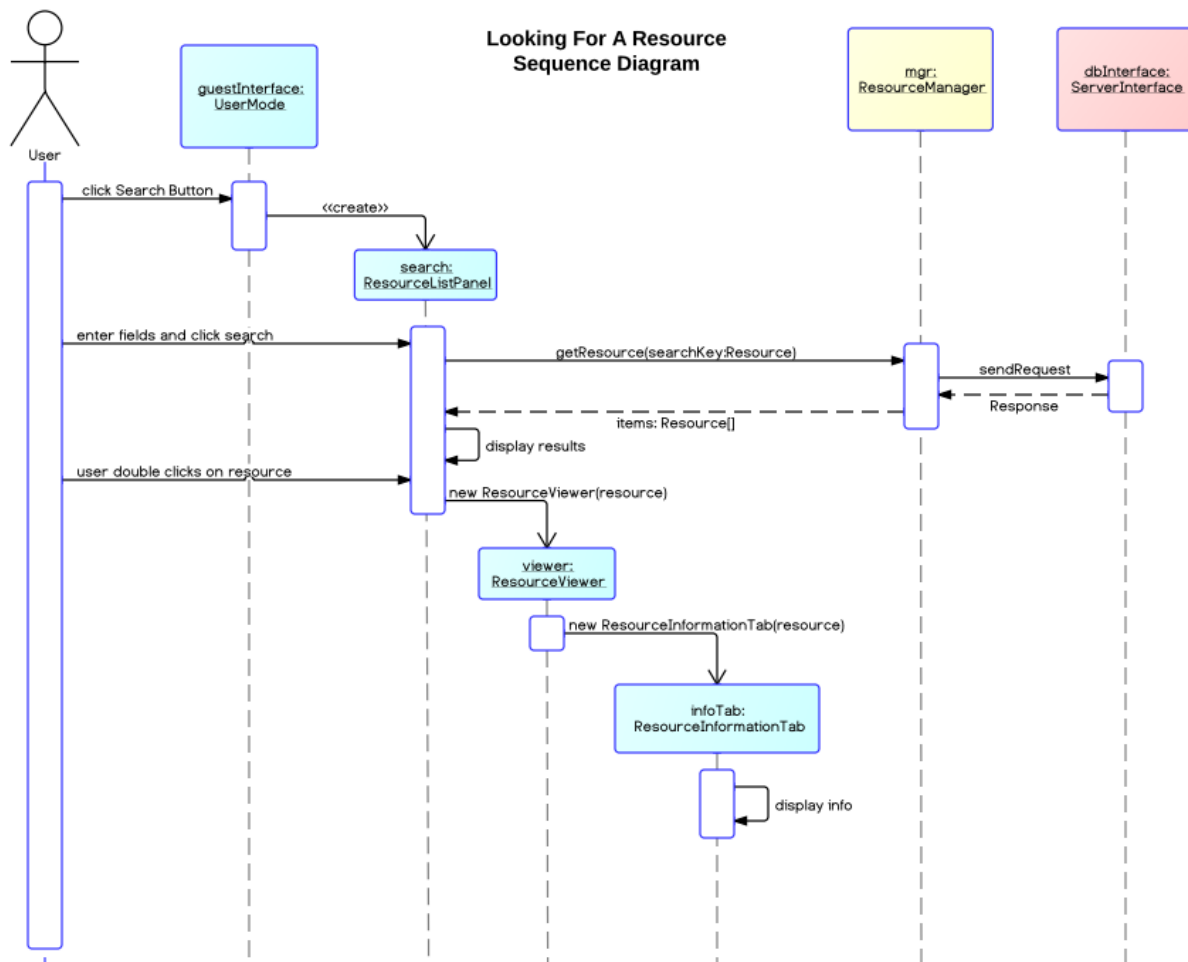


3.4.5. Dynamic Models

Sequence diagrams described in the use case models and scenarios

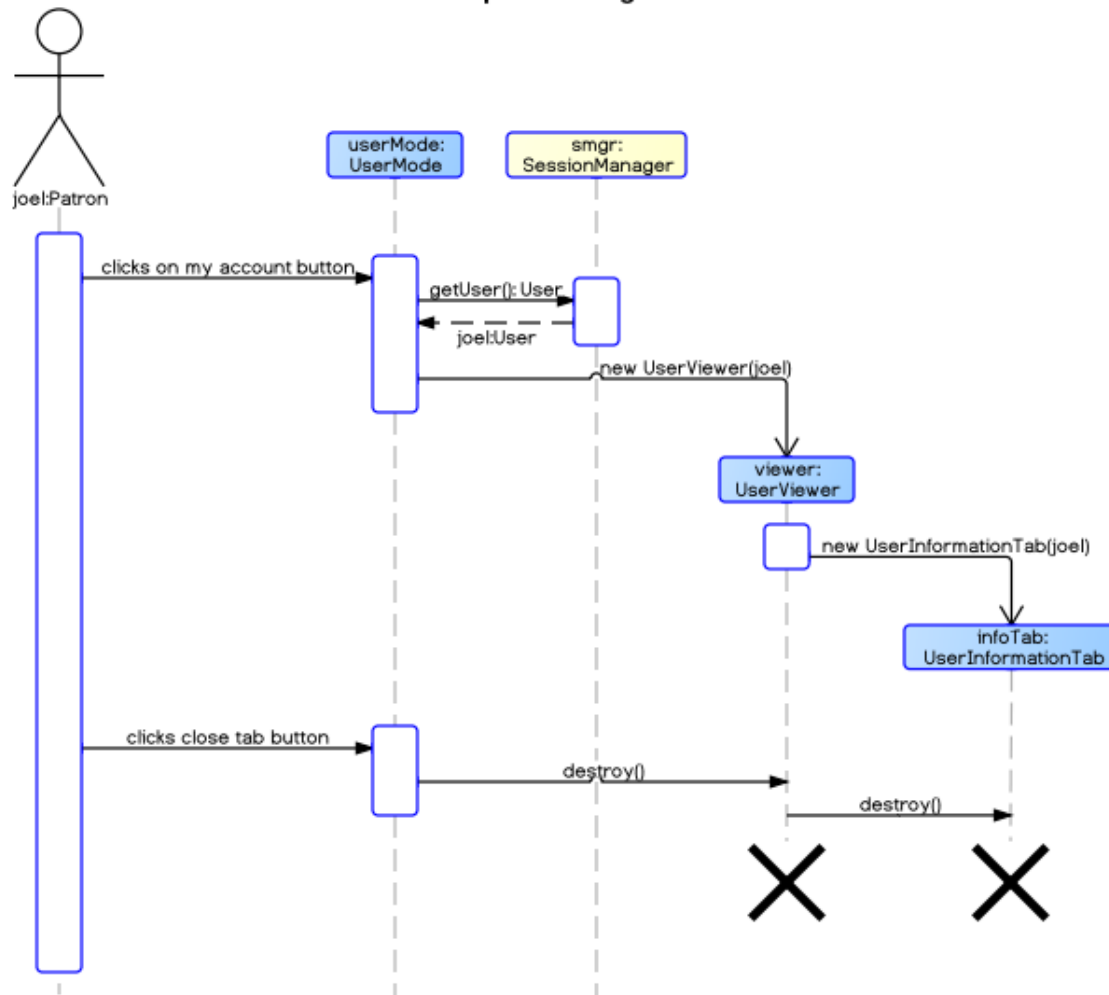
3.4.5.1. Sequence Diagrams

The following pages contain the most relevant sequence diagrams for the library system:

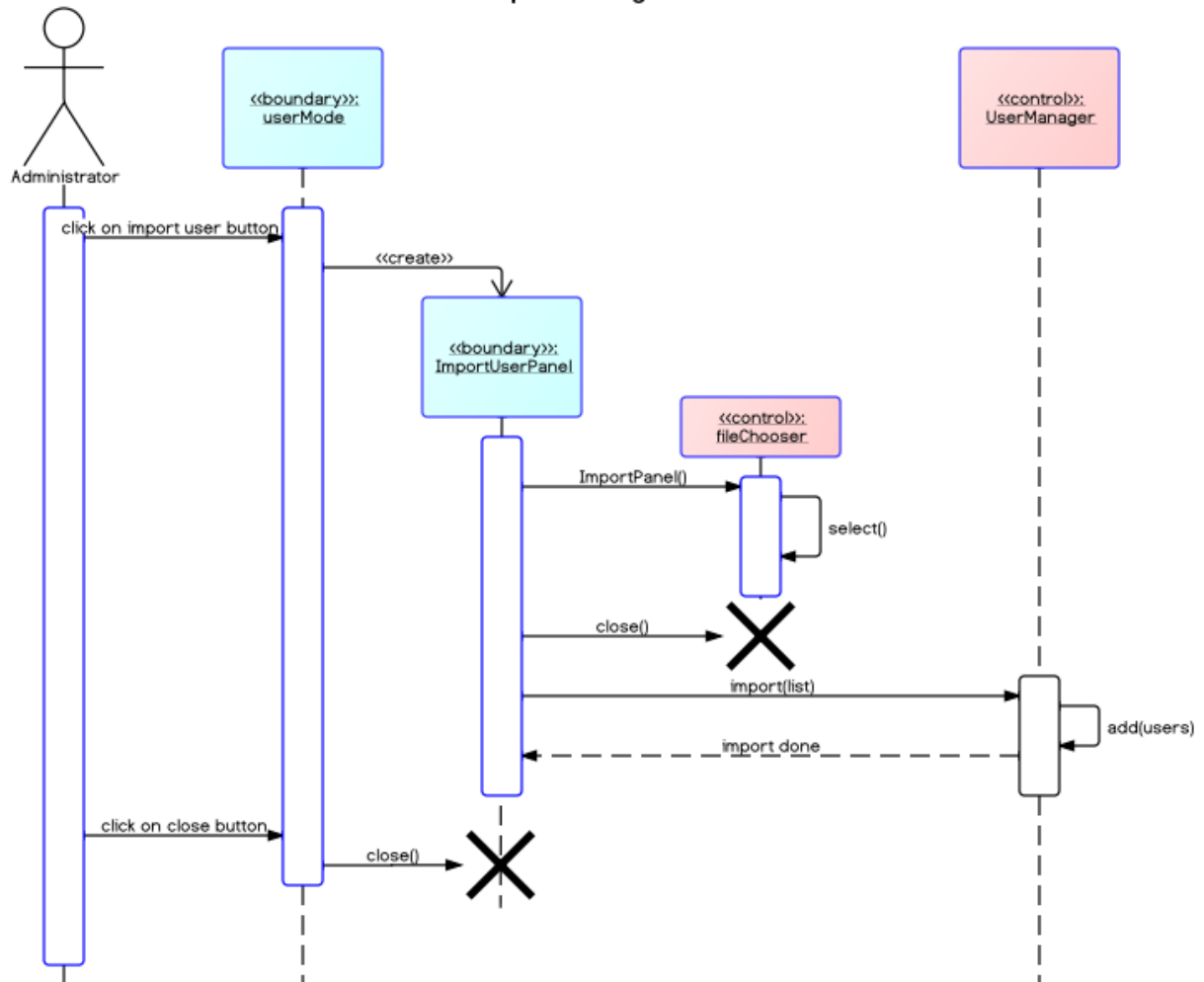


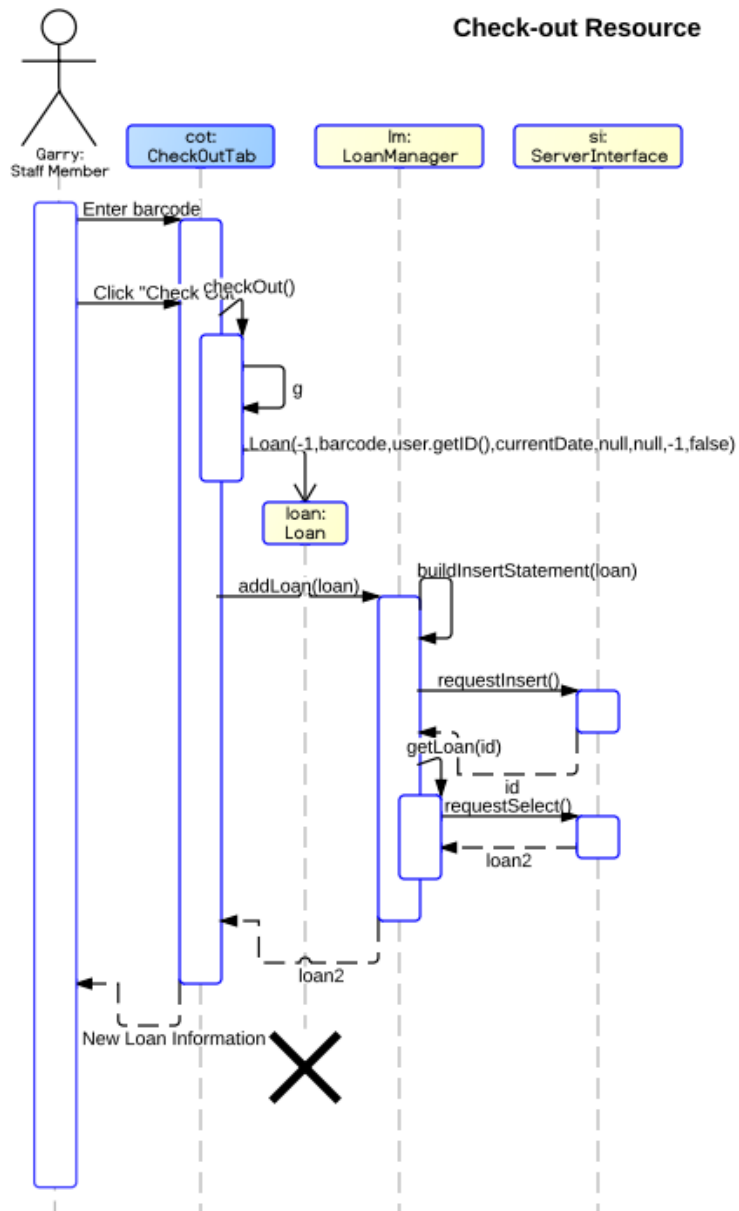
A student logs into a client to view their account information

Viewing a Users Information Sequence Diagram

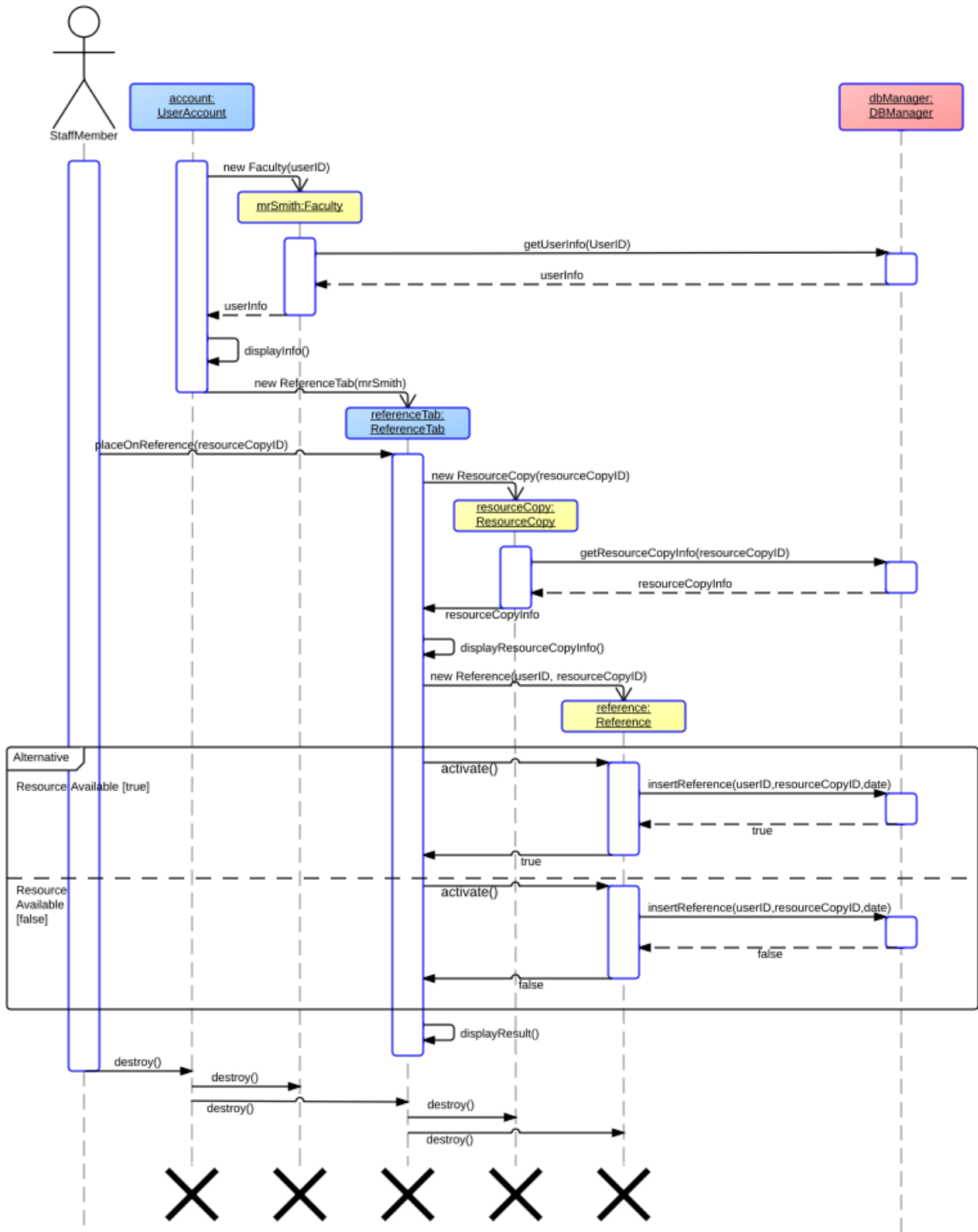


Importing New Users

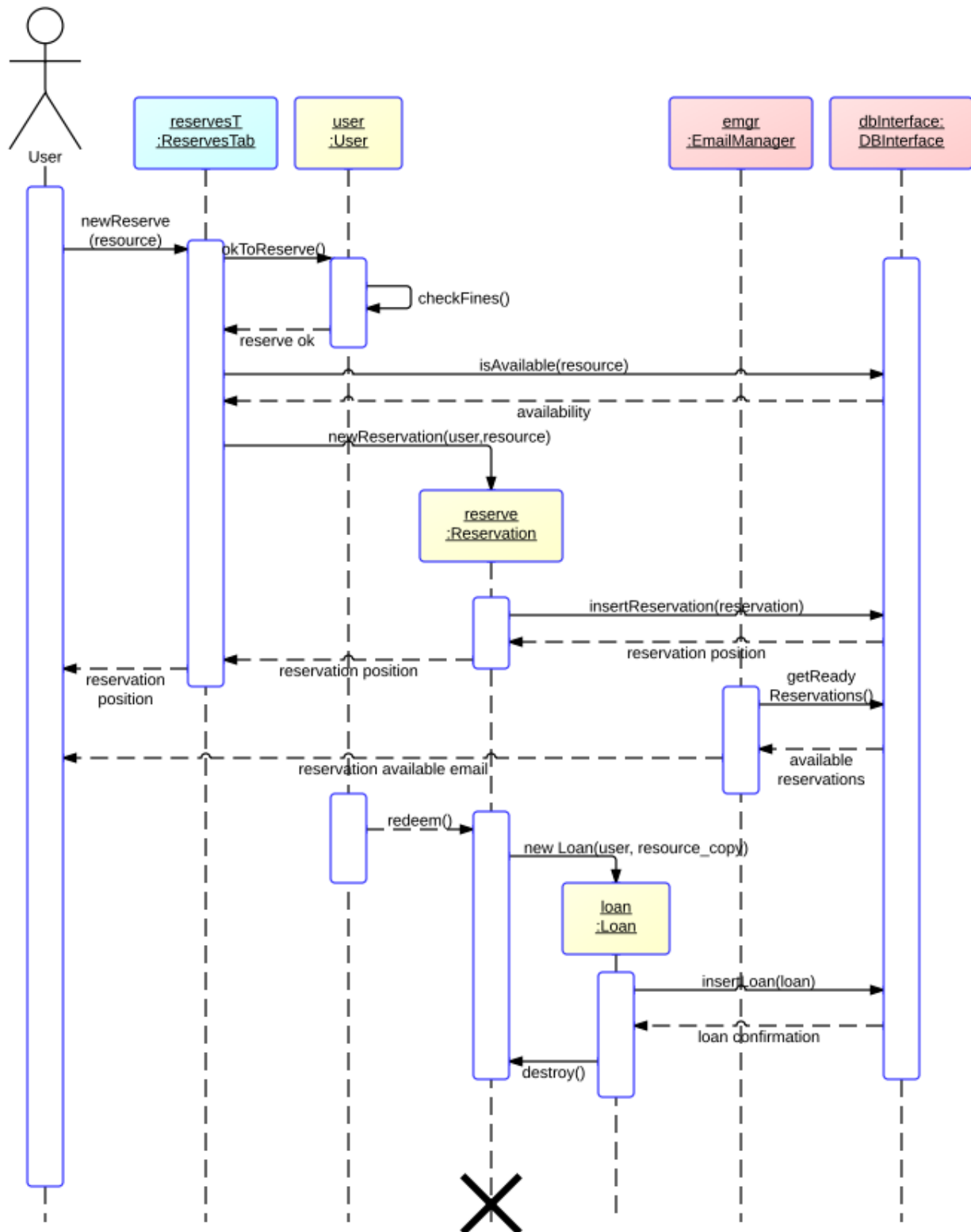
Importing New Users
Sequence Diagram



Put Resource on Reference



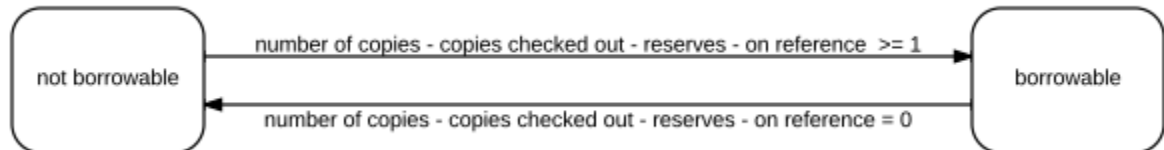
Reservation and Notification



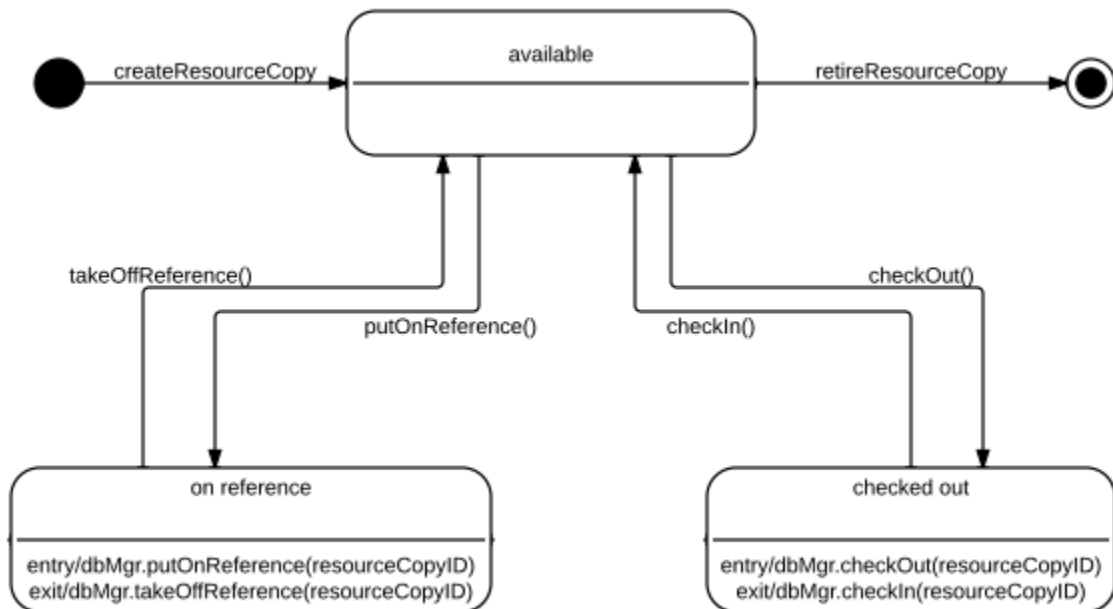
3.4.5.2. .System State Diagrams

The following are the most relevant system state diagrams for the library system:

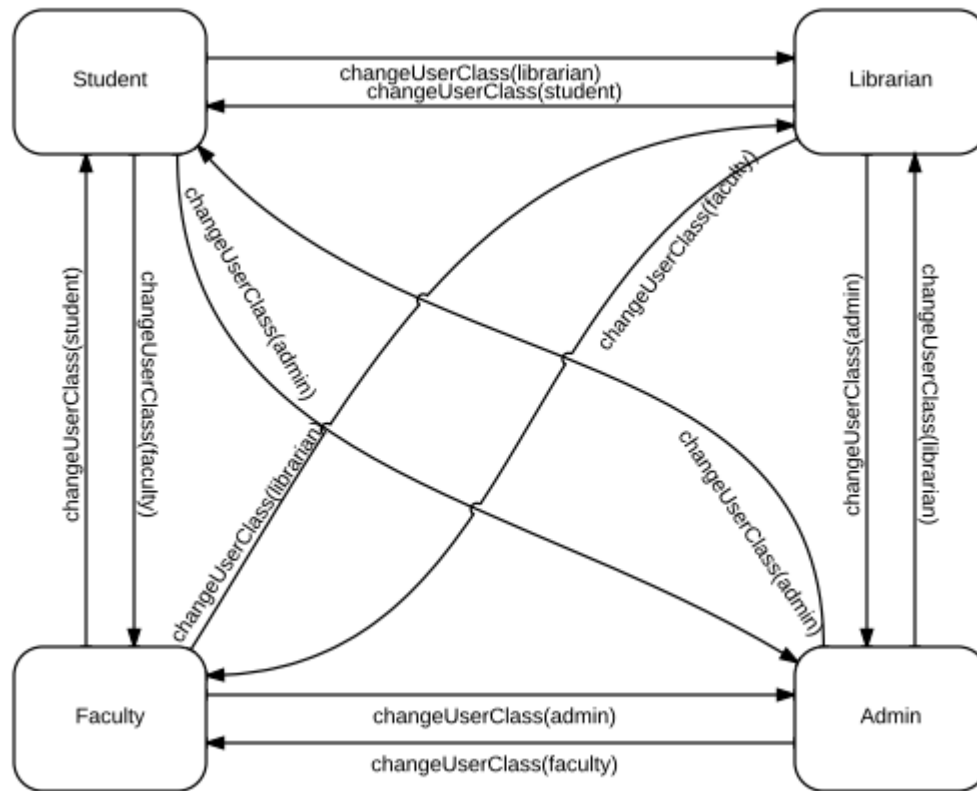
State Diagram for Resource Class



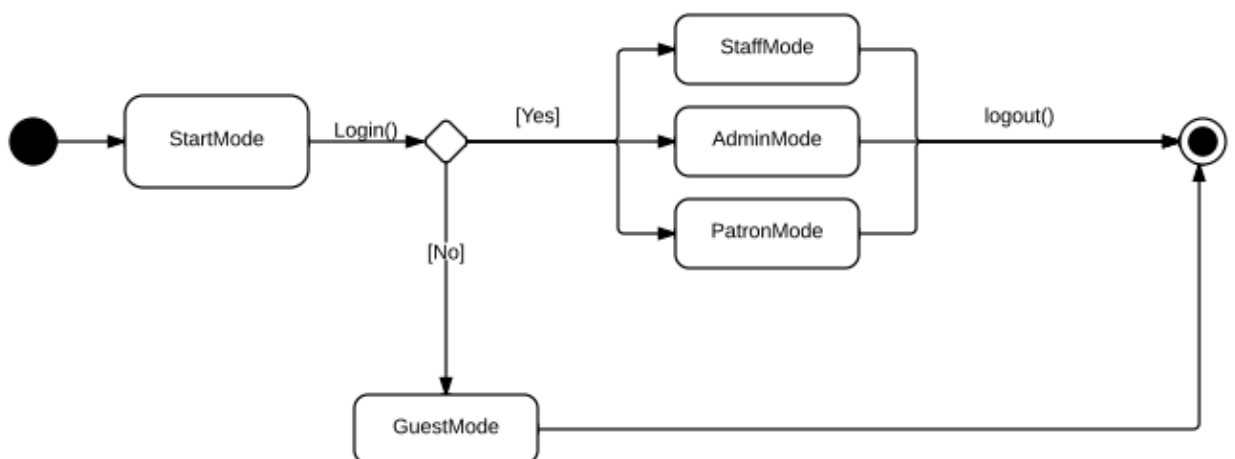
State Diagram for Resource Copy Class



System State Diagram for User Class

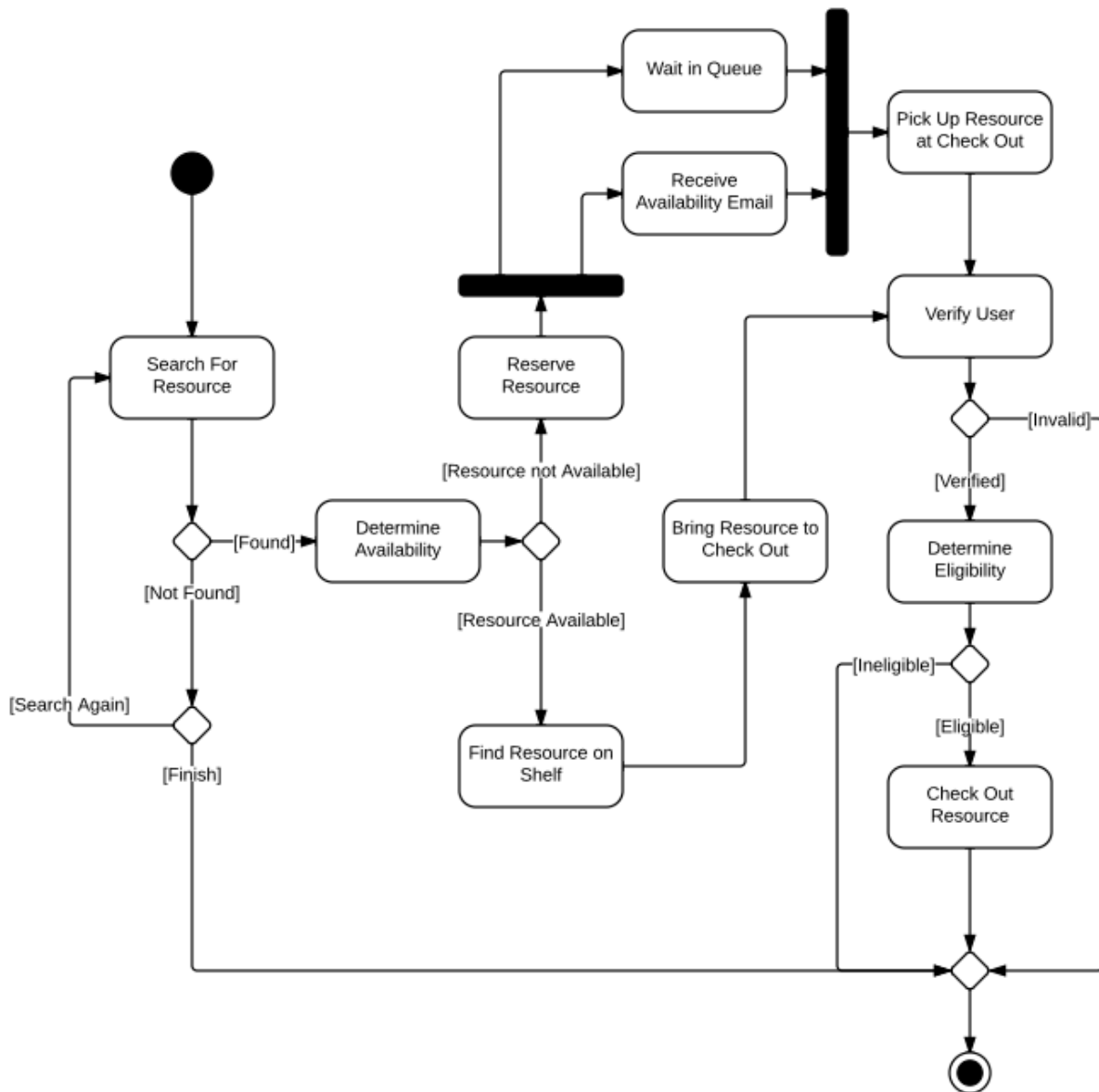


System State Diagram for GUI Mode



3.4.5.3. Activity Diagrams

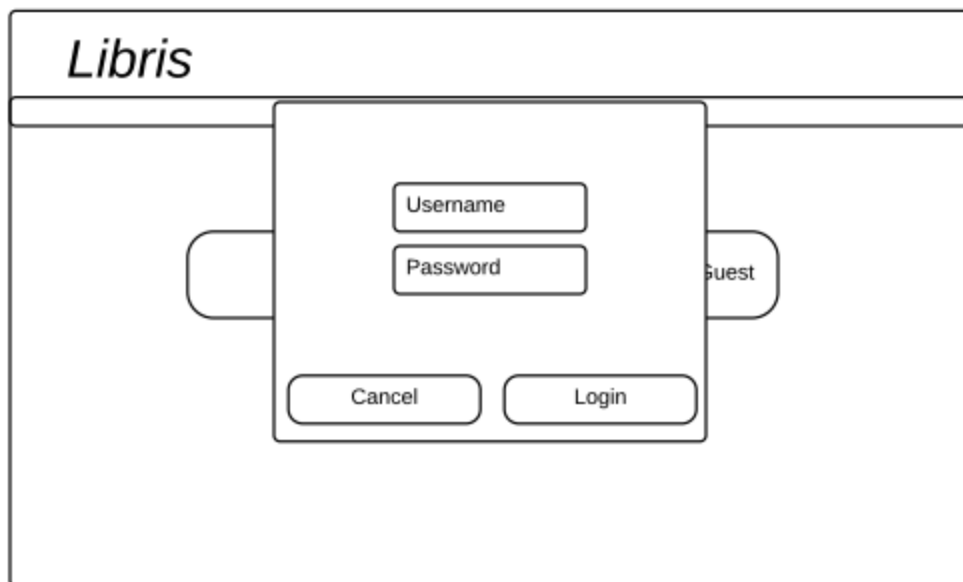
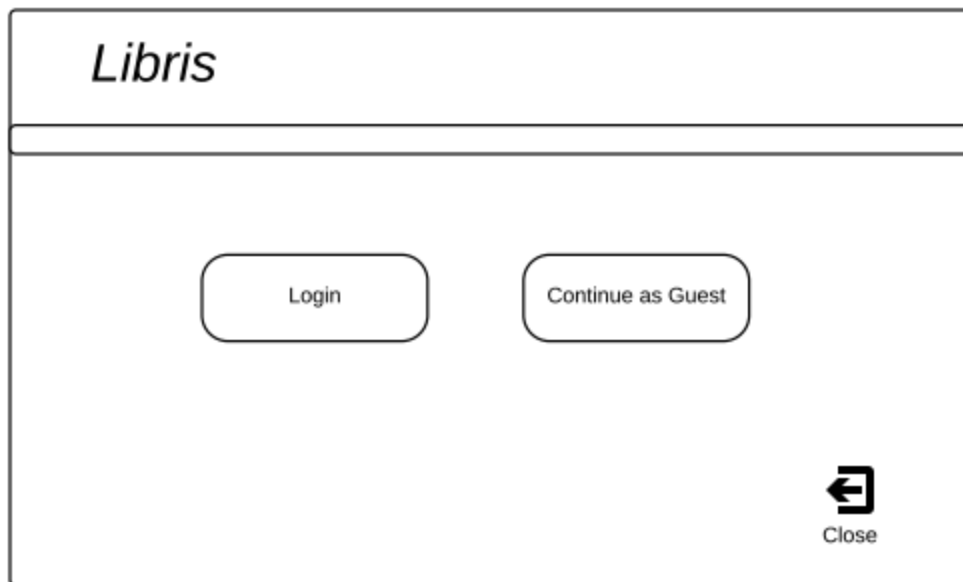
The following is the activity diagram is for the process of loaning a resource, one of the usual activities at a library:





3.4.6. User Interface


The following images display the user interface for Libris Library Management System:

Start Mode



Guest Mode

 
Back Search

Search - Book Search 

Title:

Author:



Release Date:

Publisher:

Software Engineering, 6th Edition

The complete Illustrated History of the First and ...

Rainmaking Presentations: How to Grow Your ...

 
Back Search

Resource: Software Engineering

Information

Title:

Author


Publisher


Released

Copies Available

Copies On Reference

Patron Mode


Logout


Search

My Account

Information

Fines

History

References

Reserves

First Name:

John

Last Name:

Doe

Email Address:

JohnDoe@gmail.com

User ID:


xxxxxxxxxxxxxxxxxxxx


Read Only

Read Only

Faculty Only

Enter Patron Barcode or Resource ID


Logout


Search

Resource: Software Engineering

Information

Reserves

Title:

Software Engineering 6th Edition

Author

Ian Sommerville

Publisher

Pearson Education

Released

2011

Copies Available

0

Copies On Reference

1

Enter Patron Barcode or Resource ID

Staff Mode

 Logout  Add Resource  Add User  Subscriptions  Reports  ToDo  Search

User Account: John Doe

Information

Fines

History

Check Out

References

Reserves

First Name:

John

Last Name:

Doe

Email Address:

JohnDoe@gmail.com








User ID:

xxxxxxxxxxxxxxxxxxxx

 Delete User

Only for Patrons

Enter Patron Barcode or Resource ID

 Logout  Add Resource  Add User  Subscriptions  Reports  ToDo  Search

Resource: Software Engineering

Information

Reserves

Copies

Title:

Software Engineering 6th Edition

Author

Ian Sommerville

Publisher

Pearson Education

Released

2011

Copies Available


0

Copies On Reference

1








Resource ID:

xxxxxxxxxxxxxxxxxxxx

 Delete Resource

Enter Patron Barcode or Resource ID

Staff Mode (cont.)








 Logout
  Add Resource
  Add User
  Subscriptions
  Reports
  ToDo
  Search

Subscriptions

Subscription Name	Ordered By	Renew Date
Popular Mechanics	Bob	Jan 01/2013
IEEE	James	Dec 31/2012
Make Magazine	James	Nov 01/2012

Add Subscription
Remove Subscription
Open Subscription

Enter Patron Barcode or Resource ID

 Logout
  Add Resource
  Add User
  Subscriptions
  Reports
  ToDo
  Search

Reports

Checked Out

Fines








Copies

Resource	ID	Issued to	Return Date
Software Engineering	1001241539	James Smith	Nov 21/2012
Operating System Concepts	5324518751	Gavin Free	Dec 01/2012
Discrete Mathematics	8426513729	Geoff Ramsey	Oct 04/2012
Unix	2476935214	Burnie Burns	Nov 13/2012

Check In
Renew

Enter Patron Barcode or Resource ID

Staff Mode (Cont.)








 Logout  Add Resource  Add User  Subscriptions  Reports  ToDo  Search

ToDo

Name	Issued by	Due date
Need more paper towels!	James	Nov 01/2012
Import new users for term start	Susan	Jan 02/2012
Email clients about new programs	James	Dec 01/2012

Add ToDo
Remove ToDo
Open ToDo

Enter Patron Barcode or Resource ID








 Logout  Add Resource  Add User  Subscriptions  Reports  ToDo  Search


Subscription - Popular Mechanics Magazine

Title: Popular Mechanics Magazine
Publisher: Bill Congdon
Description: Popular Mechanics is a service magazine covering a variety of information on home improvement, automotive neds, electronics, computers, ...
Renew Date: Jan 01/2013

Enter Patron Barcode or Resource ID

Staff Mode (Cont.)

 Logout  Add Resource  Add User  Subscriptions  Reports  ToDo  Search








Report - Overdue Resources 


Resource	Checked out by	ID	Return Date
Absolute C++	Gavin Free	2135425784	Oct 03/2012
Unix	Burnie Burns	5421352455	Sept 28/2012

Check In

Open Resource

Enter Patron Barcode or Resource ID

 Logout  Add Resource  Add User  Subscriptions  Reports  ToDo  Search

ToDo - Need More Paper Towels! 

We need more paper towels for both the upstairs and downstairs washrooms. Please put a new order in for the same amount as last month.

Due Date: Nov 01/2012









James

Completed

Remove

Enter Patron Barcode or Resource ID

Administrator Mode

 Logout  Add Resource  Add User  Import Users  Subscriptions  Reports  ToDo  Search

User Account: John Doe

Information

Fines

History

Check Out

User Type

References

Reserves

First Name:

John

Last Name:


Doe

Email Address:









JohnDoe@gmail.com

User ID:

xxxxxxxxxxxxxxxxxxxx

 Delete User

Enter Patron Barcode or Resource ID

 Logout  Add Resource  Add User  Import Users  Subscriptions  Reports  ToDo  Search

Resource: Software Engineering

Information

Reserves

Copies

Title:

Software Engineering 6th Edition

Author

Ian Sommerville

Publisher

Pearson Education

Released

2011

Copies Available


0

Copies On Reference

1

Resource ID:

xxxxxxxxxxxxxxxxxxxx

 Delete Resource

Enter Patron Barcode or Resource ID

4. Definitions, Acronyms and Abbreviations

Browser Screen(s): The front end screen for guests patrons to search for resources.

Catalog: Group of operations concerning creation, edition, import, export and search on full resources catalog.

Circulation: Term used to grouped checkout, check-in, renewal, reservations, reference and fines management.

Client Program: The program running on browsing workstations scattered throughout the library.

Guest: Any external person that visits the library who is not allowed to checkout resources or anyone who searches the catalog and does not log in to the system.

History: A list of all records for a given resource or patron.

Librarian: Any employee of the library with or without access to the system, also known as Staff Member. Librarian serve patrons or help patrons do research tasks.

Librarian Interface: Screen for the librarians to serve patrons or to perform administrative tasks. Depending on their job, different libraries get to see less or more options.

Libris: Latin for books.

LSM: Library Management System

Patron: Anyone using the system who is allowed to checkout resources of the system.

Reserve: A resource can be requested by other patron when it is already checked out.

Reservation: Resources on reserve.

Reference: A resource is placed on reserve for a predetermined period of time by a faculty member. Patrons may only consult reference items while in the library.

Renewal: To extend the period of a resource reaching its due date.

Resource: An item the library carries, i.e.: books, magazines, videos, etc.

Resource copy: A specific copy of a given resource.

Server Program: The application running on the server machine.

Staff Member: A more generic way to identify librarians and other employees of the library that have access to the system.

To-Do: Tasks staff members need to accomplish during a specific date.

5. References and Acknowledgements

Langara College Library. Sierra Tour. September 2012. Tour to the Langara Library's LSM software application. <<http://youtu.be/J4H4veWIFsY>>

Jaywil Software Development. Resource Mate. October 2012. Library Management System for Schools and Colleges. <<http://www.resorucemate.com>>

Evergreen. Evergreen Open-source library system. Open source LSM software. <<http://www.open-ils.org/>>

Our most sincere gratitude to the Langara Library, the Burnaby, the North Delta, and the Vancouver Public Libraries staff for their invaluable information provided to us to help us understand diverse library requirements during the requirements elicitation stage of this project.