Branch Library Management System Requirements Specification

Revision 2854, made 26/01/2012 by tws

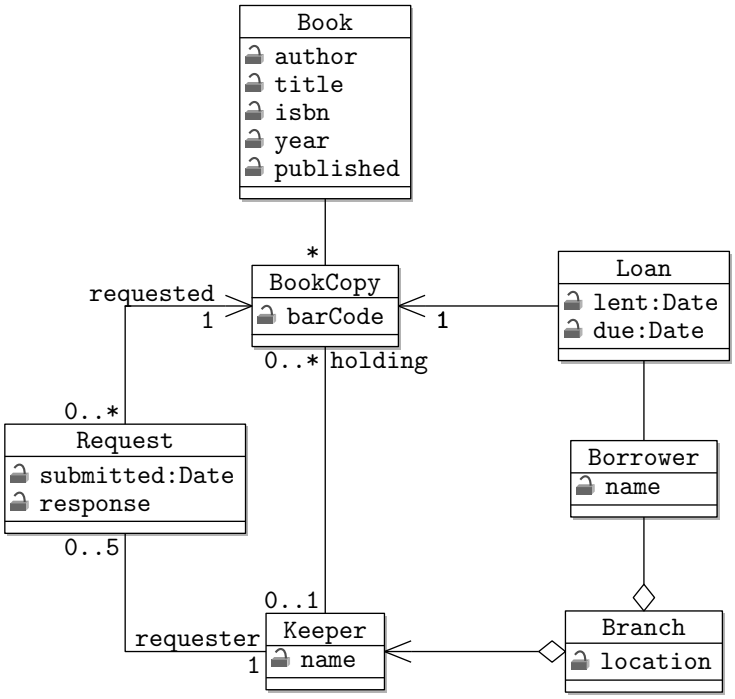
September 24, 2012

Contents

1 Problem Description

A university is considering the introduction of a branch library system in its departments to support staff and students in their work. A computer based system is needed for managing the movement of books from the central library and within the branch. Each branch will receive a collection of books from the main library, typically as a result of a request from a member of staff. These will then be available to loan out to members of staff and students.

2 Domain Model



The model illustrates the major domain elements for the branch library system.

The distributed branch is made up of a number of keepers who will hold the books from the central library. Keepers can make requests for books to be transferred from the library, which must be processed by the librarian. Books can be lent to borrowers.

Note that this is not a class diagram of a system design, since many elements are missing, attribute types are unspecified and operations have not been identified. The diagram represents the entities in the system domain that need to be represented, rather than how the system requirements will be realised in a design.

There are several issues resulting from the development of the domain model that need to be addressed by further requirements gathering.

1. What is the life-cycle of a loan - are loans kept after books are returned (e.g. for auditing purposes), or is a loan object destroyed when the book is returned to a keeper?

2. What is the life–cycle of a request, is it disposed of once it has been processed by a librarian?

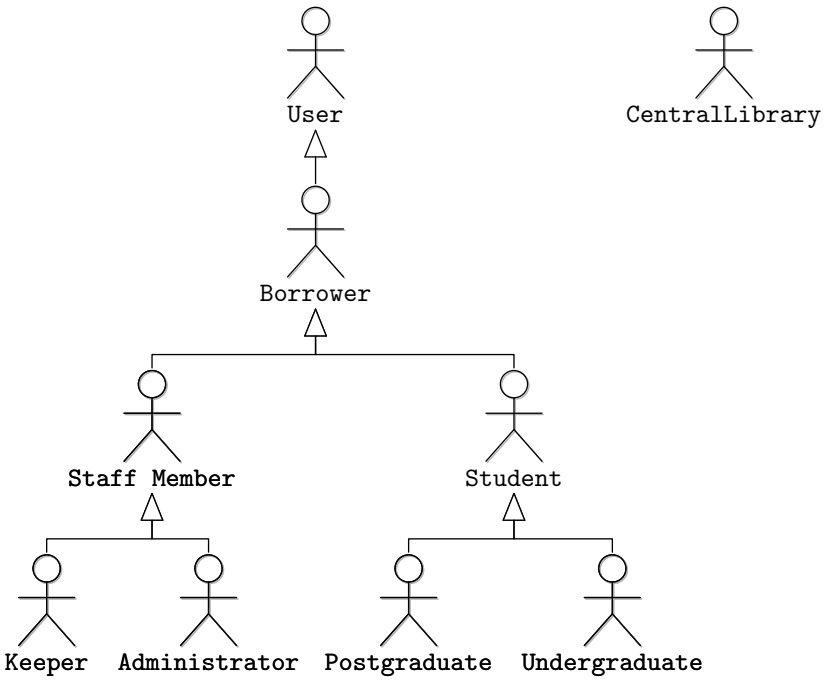


Figure 1: revised actors for the branch library system

3 Actors

Figure 1 illustrates the relationships between actor roles in the system. A short summary of the actors is given below:

User represents any actor on the system. Is able to search the library system for books and their locations

CentralLibrary handles the Central Library functions for distributing books to branches

Borrower a user who is registered as a borrower by the CentralLibrary

Student a type of borrower with limited borrowing rights

Staff Member a type of borrower with the potential to be a Keeper

Keeper a member of staff in the department hosting the branch, with the authority to request books from the Central Library and sub-lend books to borrowers

Administrator a member of staff in the department hosting the branch, appointed to manage keepers in the branch

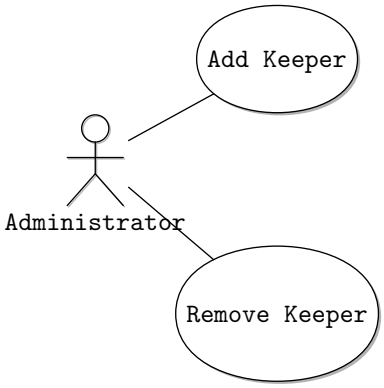
4 Use Cases

This section describes the required functionality for the Branch Library System as four groups of related use cases. The core use cases for the system are:

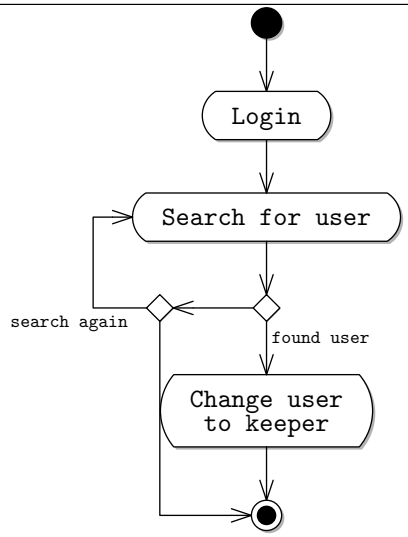
- common utility activities (Section 4.5);
  - Login
  - View Status
  - Logout
  - Search for book
  - Search for user
- keeper administration (Section 4.1);
  - Add keeper
  - Remove keeper
- transferring books to keepers in the branch from the Central Library (Section 4.2);
  - Request book for branch
  - Process request
  - Record receipt of book
- sub-lending books to borrowers by keepers (Section 4.3)
  - Record book lend
  - Request book return (keeper to borrower)
  - Record book return (Branch)
- returning books to the Central Library (Section 4.4)
  - Request book from Branch Library
  - Record book sent
  - Record book return (Central)
- some further use cases have also been identified as potentially useful. The clients do not envisage that these use cases should be implemented in the current construction phase and have not been fully described in the previous elaboration phase.
  - Transfer book between keepers
  - Record book return by borrower to the Central Library
  - Apply fine for late return.

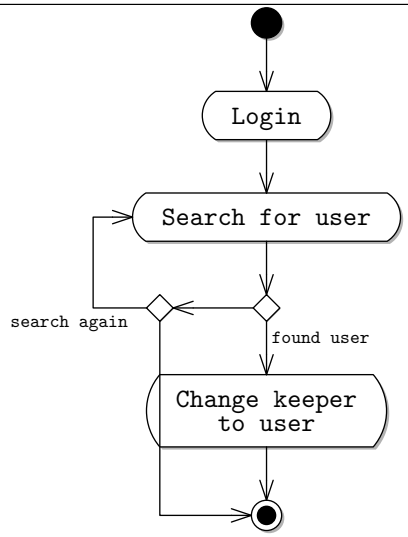
4.1 Keeper Administration

Use Case Diagram



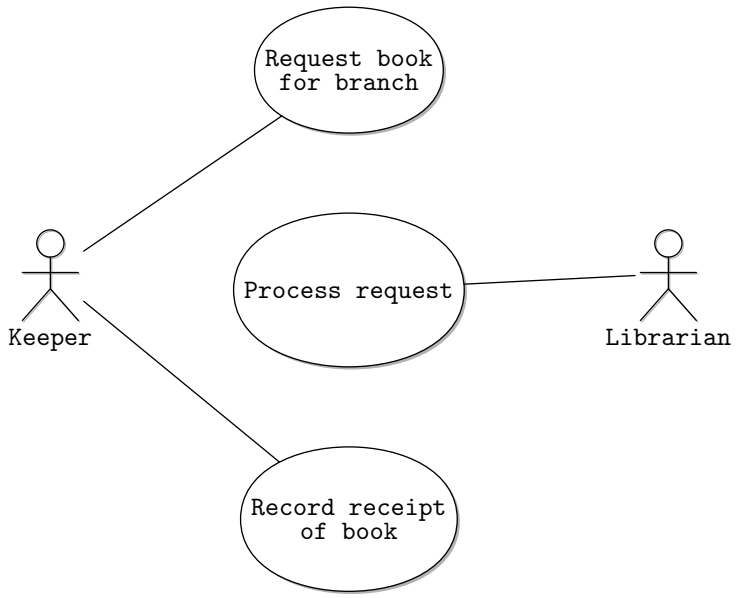
A member of staff appointed in the department as an administrator is responsible for adding and removing members of staff as keepers of books.

Use case	Add Keeper
Description	
Rationale	A member of staff appointed in the department as an administrator is responsible for adding and removing members of staff as keepers of books.
Priority	Must have
Status	Not implemented
Actors	• Administrator
Extensions	
Includes	• Login • Search for user
Conditions	**post** The member of staff is registered as a keeper able to receive and lend books.
Non-Functional Requirements	
Scenarios	
Risks	• The system may not be able to determine whether a user is a member of staff, eligible to be designated as a keeper.
User Interface	

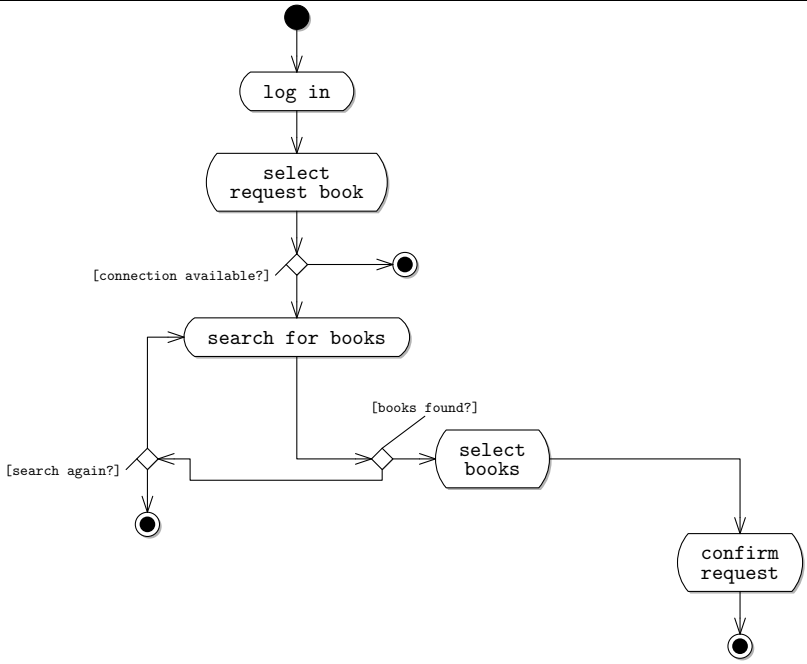
Use case	Remove Keeper
Description	
Rationale	A member of staff appointed in the department as an administrator is responsible for de-authorising members of staff as keepers.
Priority	Should have
Status	Not implemented
Actors	
Extensions	
Includes	• Login • Search for user
Conditions	**pre** The keeper to be removed must exist in the system. **pre** The keeper must not have any books, or have lent books to users. **post** The member of staff is de-authorised to keep and loan books.
Non-Functional Requirements	
Scenarios	
Risks	• The system may not be able to determine whether a user is a member of staff, eligible to be designated as a keeper. • Keepers may need to be de-authorised while they still have books on loan to other users.
User Interface	

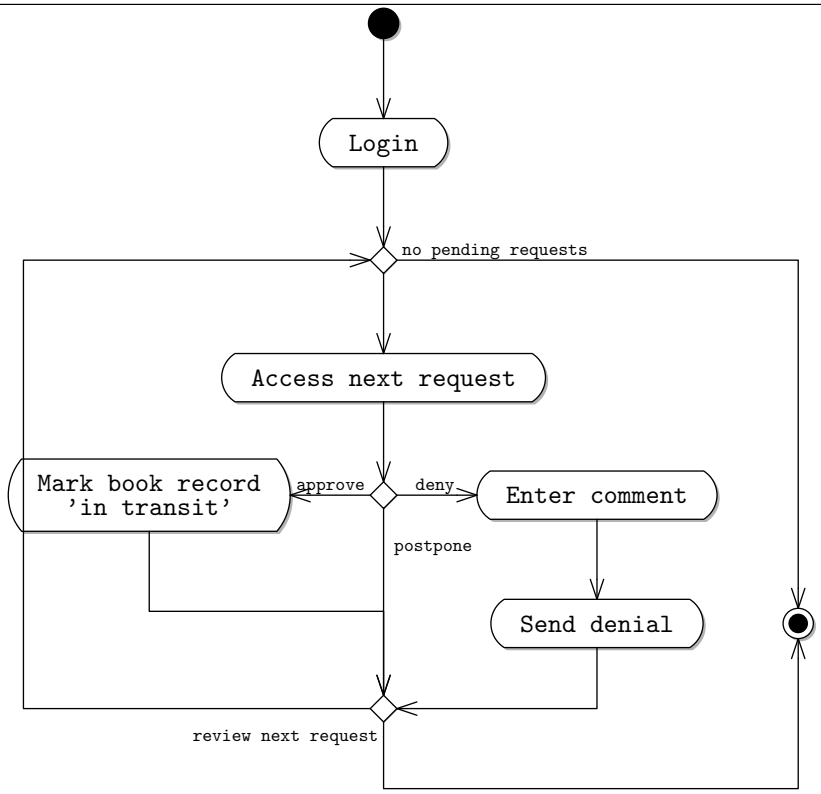
4.2 Transferring Books from the Central Library to the Branch

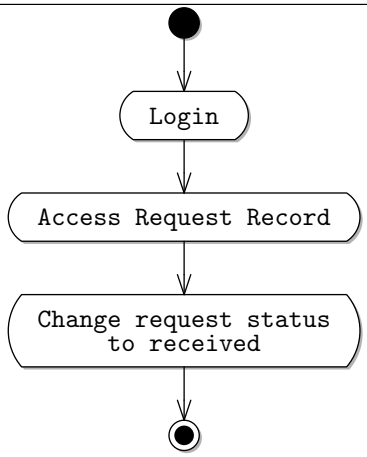
Use Case Diagram



A keeper can request books to be transferred from the Central Library to the branch (in the possession of the keeper). The librarian must process each request. When the book arrives in the branch, the keeper must record receipt of the book before it can be sub-lent.

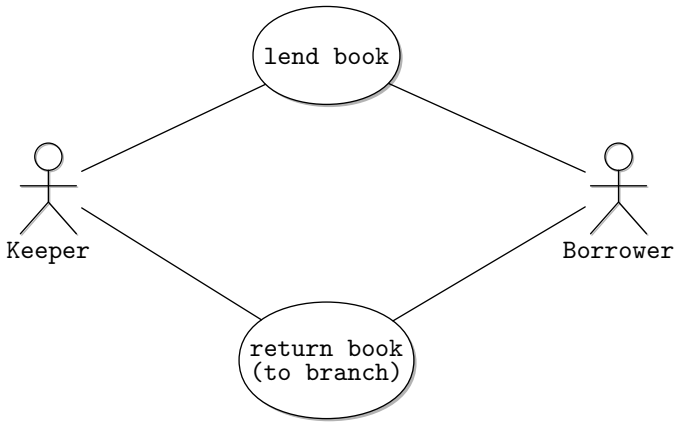
Use case	Request book for branch
Description	
Rationale	Keepers need to be able to request books to be transferred into the branch library.
Priority	Must Have
Status	Not implemented.
Actors	• Keeper
Extensions	
Includes	• Login • Search for book
Conditions	**post** A request for the transfer of the requested book is stored for processing.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Process Request
Description	
Rationale	Librarians need to process requests for books from the branch keepers. Keepers need to be notified of the Librarian's decision to approve or deny requests.
Priority	Must Have
Status	Not implemented
Actors	• Librarian • Keeper
Extensions	
Includes	• Login
Conditions	**post** Any request of deny decisions are recorded. **post** The requesting keeper is notified of the librarian's decision.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

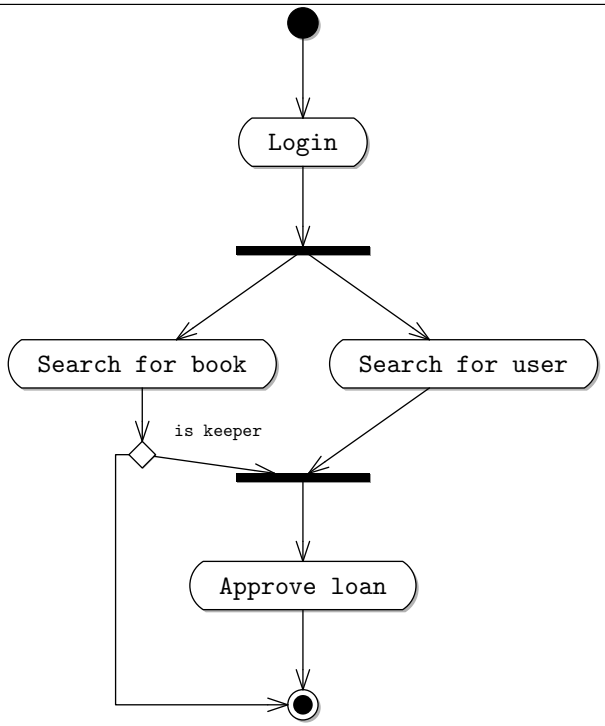
Use case	Record receipt of book
Description	
Rationale	The keeper is required to explicitly acknowledge receipt of a book before they lend it out to a borrower. The use case supports the tracking and auditing of a book's location.
Priority	Must Have
Status	Not implemented
Actors	• Keeper • Librarian
Extensions	
Includes	• Login
Conditions	**pre** The book must be recorded as 'in transit'. **post** The book is recorded as in the possession of the keeper.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

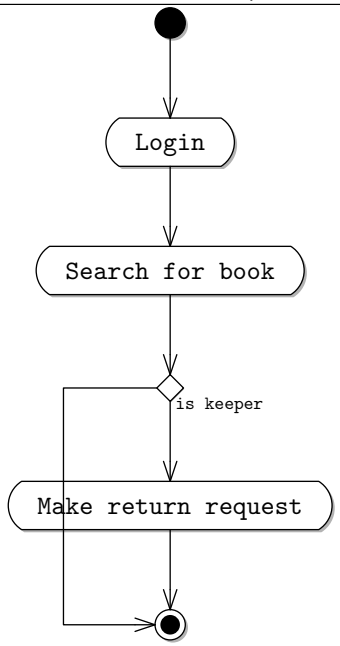
4.3 Sub-Lending Books

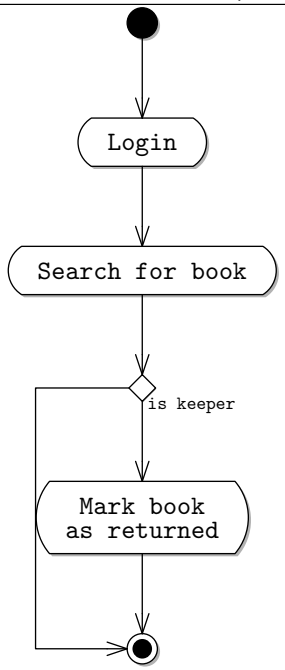
Use Case Diagram



Keepers can sub-lend books in their possession to borrowers.

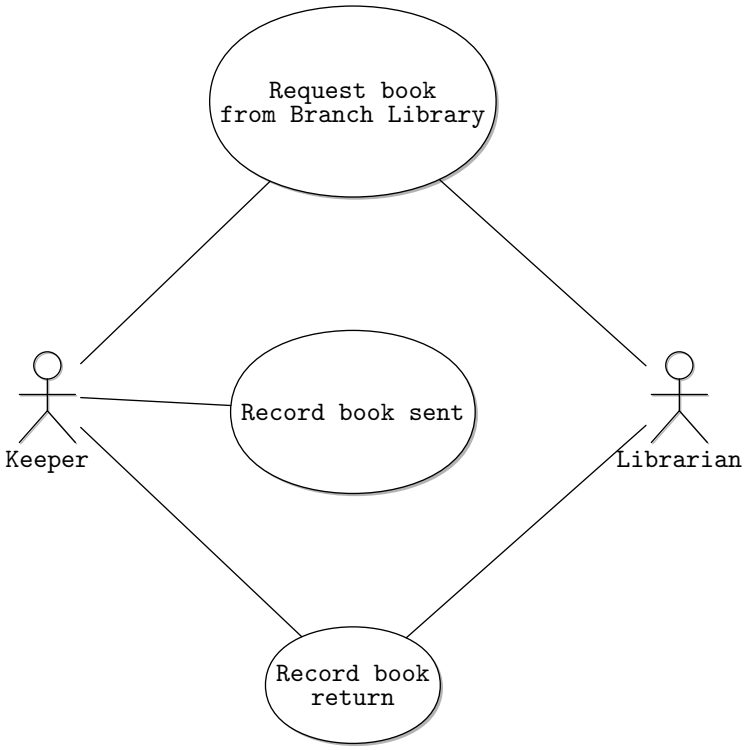
Use case	Record book lend
Description	
Rationale	Keepers need to be able to document the lending of books to borrowers. Borrowers are notified when they have been assigned a book.
Priority	Must have
Status	Not implemented
Actors	• Keeper • Borrower
Extensions	
Includes	• Login • Search for book • Search for borrower
Conditions	**post** The status of the book is changed to 'lent' to named borrower.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Request book return (to Branch)
Description	
Rationale	Keepers need to be able to request that books are returned by a borrower.
Priority	Must have
Status	Not implemented
Actors	• Keeper • Borrower
Extensions	
Includes	• Login • Search for book
Conditions	**pre** Book status is set to "lent to named borrower" **post** The borrower recieves notification that the keeper wishes for the book to be returned.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

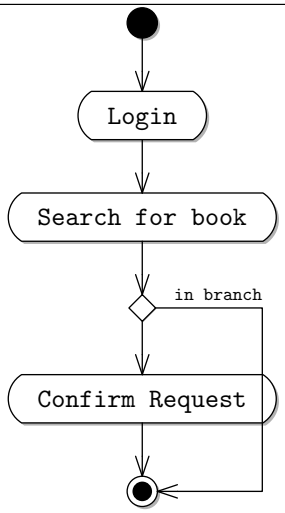
Use case	Record book return (to branch)
Description	
Rationale	Keepers need to be able to document the collection of books from borrowers. Borrowers are notified when a book they have borrowed is recorded as having been returned to the book's keeper.
Priority	Must have
Status	Not implemented
Actors	• Keeper • Borrower
Extensions	
Includes	• Login • Search for book • Search for user
Conditions	**pre** The keeper is responsible for the book. **post** The status of the book is changed to in possession of the keeper.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

4.4 Return books to the Central Library

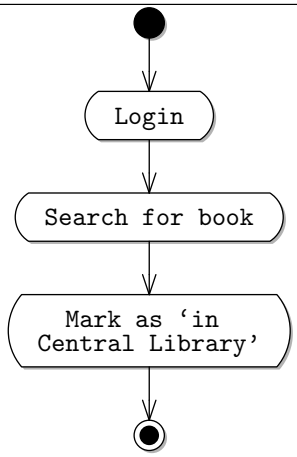
Use Case Diagram



Librarians in the Central library are able to request that books stored by branches be returned. Book keepers are responsible for sending books back to the library and can document this action on the system. The librarian records books as having been received back at the Central Library.

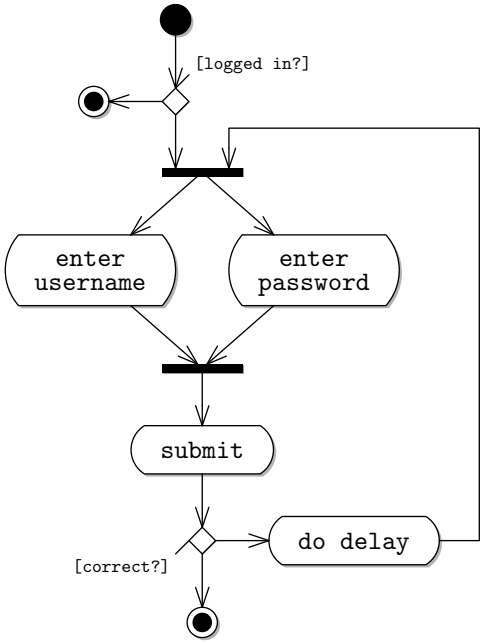
Use case	Request book from Branch
Description	
Rationale	Administrators need to request books be returned from a branch, when a central library borrower needs the book, for example.
Priority	Must Have
Status	Not implemented
Actors	• Librarian • Keeper
Extensions	
Includes	• Login • Search for book
Conditions	**post** A return request is recorded for the book.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Record book sent
Description	includegraphicsfigures/refinement/book-to-central/record-book-sent-1
Rationale	This feature allows keepers to document when a book has been sent back to the Central Library.
Priority	Should have. This feature is useful for tracking purposes but is not part of the core work flow of the system.
Status	Not implemented.
Actors	• Keeper
Extensions	
Includes	• Login • Search for book
Conditions	**pre** A return request has been made for the book. **pre** The book is recorded as in the keeper's possession. **post** Book is recorded as being 'in transit' to the Central Library.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Record book return
Description	
Rationale	Supports book tracking by allowing the librarian to record when a book is received back at the central library.
Priority	Must Have
Status	Not implemented
Actors	• Librarian
Extensions	
Includes	• Login • Search for book
Conditions	**pre** A return request has been made for the book. **pre** The book is recorded as 'in transit' or in possession of a Keeper. **post** Book is recorded as being in the Central Library.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

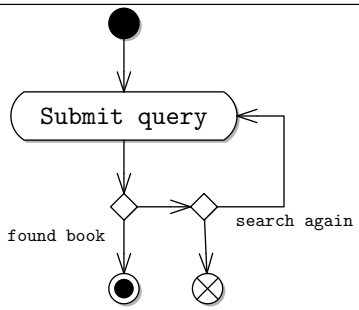
4.5 Common shared Use Cases

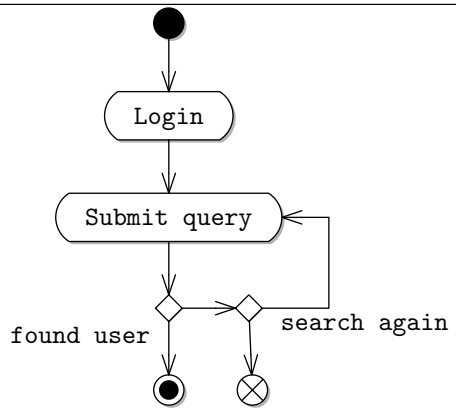
This section describe a number of use cases which are included elsewhere in the model.

Use case	Login
Description	
Rationale	The login use cases permits different levels of access to the system. A username is the unique user ID provided by the Central Library. A password must be up to eight alpha numeric characters.
Priority	Must have
Status	Not implemented
Actors	• Librarian • Borrower
Extensions	
Includes	
Conditions	**post** The user is logged in if correct credentials are provided
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	View Status
Description	A user's status (books borrowed, books lent, privileges, requests for book returns) is displayed.
Rationale	Users' need to be able to view their status within the system.
Priority	Must have
Status	Not implemented
Actors	• User
Extensions	
Includes	
Conditions	**pre** The user is logged in.
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Logout
Description	The logout use cases changes a users account to logged out, requiring them to re-authenticate to the system. Logout is either invoked by the user or by the internal inactivity timer actor.
Rationale	The logout use case allows a user to leave the system to prevent unauthorised access from an unattended terminal.
Priority	Must have
Status	Not implemented
Actors	• Librarian • Borrower • Inactivity Timer
Extensions	
Includes	
Conditions	**pre** The user is logged in **post** The user is logged out
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Search for book
Description	 Minimally, a user may search by: • an initial substring for the title of a book. For example, all books whose title begin with the term "Software Engineering" would be included in the search result for the query "Software Eng"; • the exact identifier of a book.
Rationale	Allows the branch library to be queried for locations of books with keepers. Note that searching books does not require a user to log in to the system.
Priority	Must have
Status	Not implemented
Actors	• User
Extensions	
Includes	
Conditions	
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

Use case	Search for user
Description	
Rationale	Allows the branch library to be queried for information on users by keepers.
Priority	Must have
Status	
Actors	• Keeper
Extensions	
Includes	
Conditions	
Non-Functional Requirements	
Scenarios	
Risks	
User Interface	

5 Non Functional Requirements

The system will need to be able to interact with the central library's existing database of books and users.

1. Any data must be stored in such a way that the storage medium can be changed without changing the system's architecture.
2. The system must interact with the Central Library's database of books.
3. The system must interact with the Central Library's database of users. Access control must be implemented with reference to this database.
4. The user interface to the system should emphasise simplicity and must be minimally functional for the purposes of the final demonstration.