**qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm**

|  |
| --- |
| Software Engineering  Assignment 1  Donncha Cassidy-Hand |

Table of Contents

[**Introduction 1**](#_Toc405555725)

[**Use Case Diagram 2**](#_Toc405555725)

[**Book Borrow & Reserve Class 3-4**](#_Toc405555725)

**Library Activity Diagram** [**5**](#_Toc405555725)

[**Book Borrow & Reservation Sequence Diagram 6-7**](#_Toc405555725)

[**Borrow & Reserve State Chart 8-9**](#_Toc405555725)

[**Code Snippets 10-12**](#_Toc405555725)

# Introduction

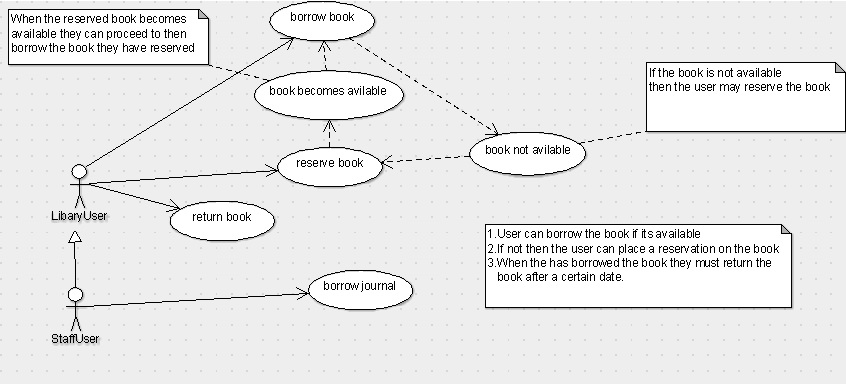
The topic I have chosen to investigate for this assignment is a Library with the ability for a user to be able to borrow a book, reserve a book and return a book. The user will be presented with these options allowing the system to be flexible and convenient for the user.

My goal is to have a argoUML file which will demonstrate using a detailed **Use Case Diagram**, **two Class Diagrams**, **Two State Charts**, and **One Activity Diagram.**

There will also be a demonstration of a PHP code that will allow the user to reserve or borrow a book and show how they can then unreserve a book and have their details associated with the book they have reserved or borrowed.

# Use Case Diagram

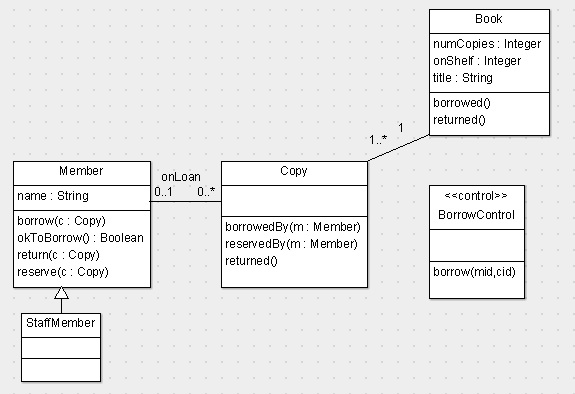
The use case diagram shows the routes that a user may take, these including been able to borrow a book, return a book and reserve a book. If the book is already borrowed then the user may reserve the book and when the book becomes available they may then proceed to borrow the book and then later return the book when it is due back.



# Book Borrow Class

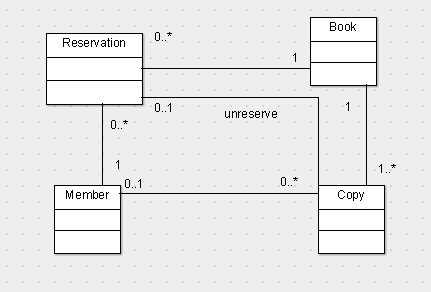
This class provides elements to allow the borrowing of a book that is used to represent both logical and model classes used to create tables. In this class the tables Book, Copy, Member and Staff member are been created. Within those tables they are given specific variables with an appropriate data type to allow the table to function as desired i.e. numCopies: Integer, is providing the number of copies within the book table and is given an integer data type which will provide a whole number.

Also there are relationship types been established between the tables within the class i.e. 1 to 1, 1 to many etc. A stereotype is been declared to allow a control for borrowing the book.



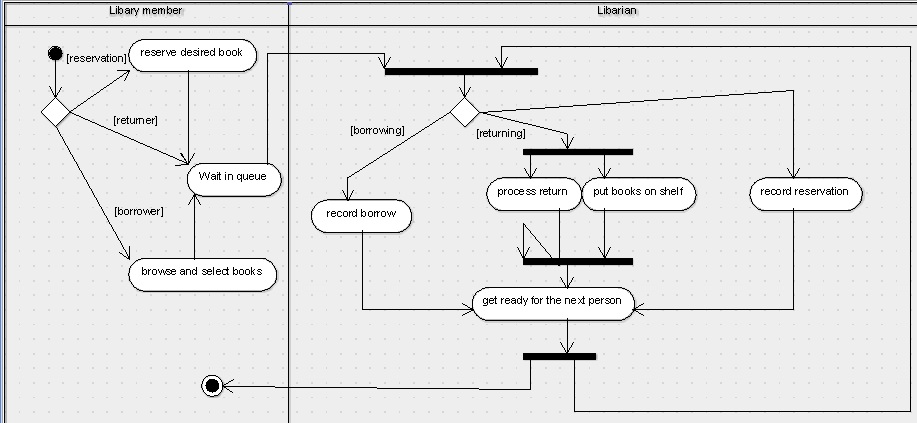
# Reservation Class

In the reservation class the appropriate declarations are been made to allow for the reservation of a book. There are four tables been created with the appropriate relationships to each other. It allows for the member to reserve a copy of a book and the unreservation of a copy.



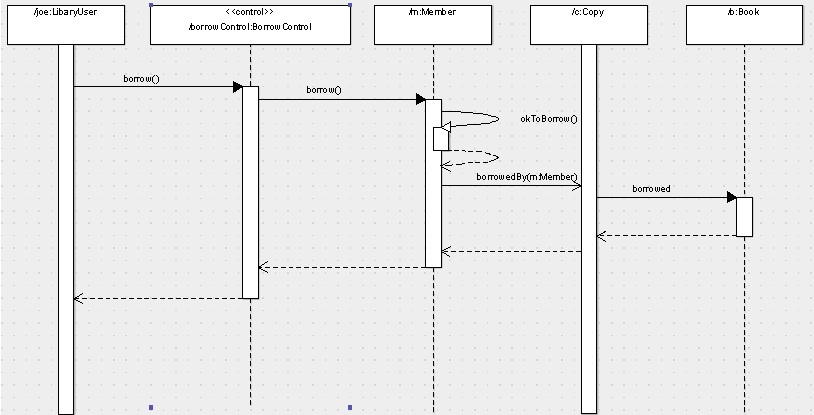
# Library Activity Diagram

The library activity diagram provides various sequences that the user must take with various options that they may choose from on the way. It allows for the user to initially reserve a book, borrow a book or wait in the queue but for example it all leads to them waiting in the queue anyway before the process may proceed. As the user progresses through the Activity Diagram they are presented with a number of choices and processes that they must go through before they can complete it.



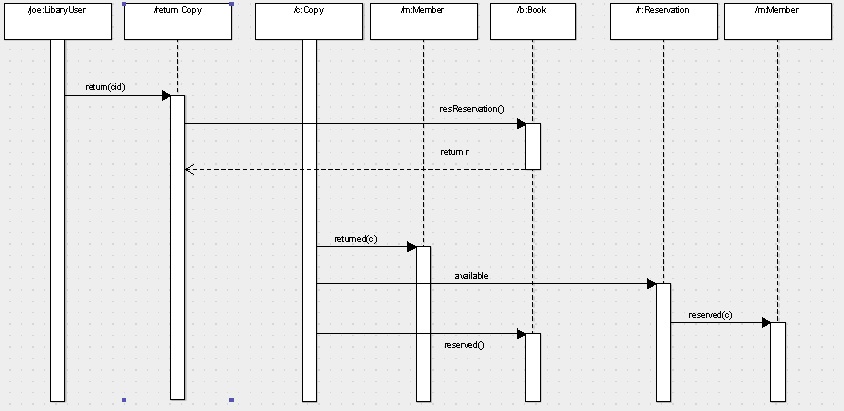
# Borrow Book Sequence Diagram

This sequence diagram shows the sequence in which an actor user (Joe) may be processed through the system to complete the borrowing of the book through a timeline with the calls, returns, create or destroys been in the appropriate places for the user to borrow or then return the a book.



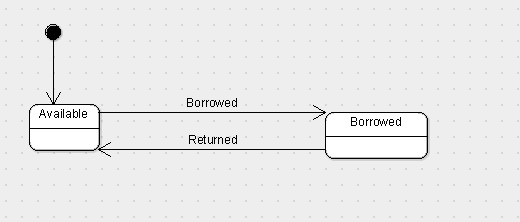
# Reserve Book Sequence Diagram

The reserve book sequence diagram allows the actor user(joe) to be passed through a timeline to proceed to the reservation and borrowing of a book by using the appropriate calls, returns, creates and destroys to complete the timeline.



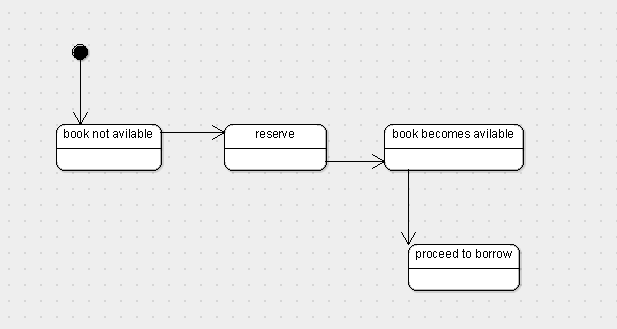
# Borrow State Chart

The following state chart shows the initial or terminal state and transitions that will be made to borrow a book, and then return a book once that book has been borrowed.



# Reserve State Chart

The reserve state chart shows the initial or terminal state processing or making transitions if not the book is not available to be able to go through the transitions to then reserve the book and proceed to borrow it for when it becomes available.

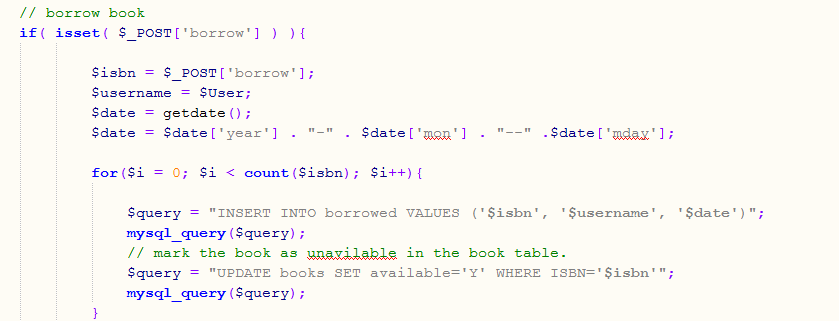


# Code Snippets (PHP)

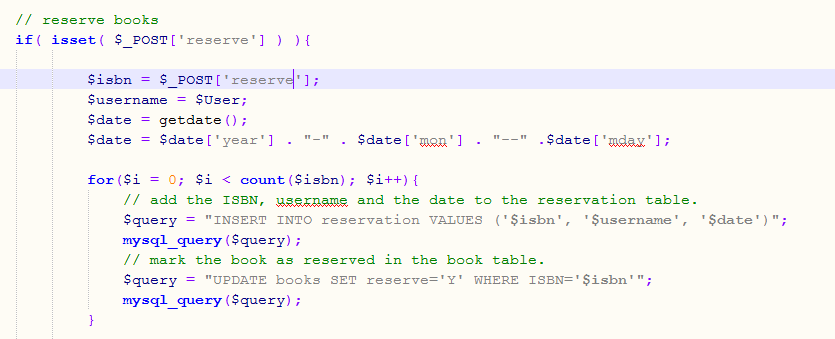
The following PHP code allows the users (member) to borrow a book. If the book is not available then the user may reserve a book provided that the book is not already reserved as well. It will take the user details and date and apply them to the book when it’s reserved or booked.

After the user has either reserved the book they may unreserved the book.

The code below allows the user to borrow the book. When the user borrows the book their user details, books details and date are associated with the book and then the book is made unavailable.



The code below allows the user to reserve a book if it’s already borrowed. When the user borrows the book, again the user’s details, book’s details and the current date is associated with the book and the reservation is then made unavailable for another person to book it.



The code below takes all the details of the books associated with the user and allows for the user to unreserve a book. When the book is unreserved the details of the book are then unassociated with the user.

