Milestone 2:

Identify your stakeholders, and discover the actual needs of your customers.

Team CABS:

Marcus Blaisdell

Vivek Sandhinti

Joseph Crissey

Christian Asivido

**1. Identify Stakeholders:**

**Stakeholders:**

-        Librarians

-        Librarian assistants

-        Administrators

-        Students

-        Educators

-        Software developers

-        Software maintenance

-        WSU / the school using the system

**2. Use cases**

**Actors:**

-        Librarians - primary

-        Librarian assistants - primary

-        Students - primary

-        Educators - primary

-        Unregistered patron (public patron) - primary

-        Database server - secondary

-        Access terminal - secondary

**Librarians:** *(inherits from Librarian Assistant)*

-        Goals:

Maintain system

-        Preconditions:

Registered super user

-        Main tasks:

Add / delete books

-        Exceptions:

-        Variations:

-        Information – acquire; produce; change:

-        Inform about external changes:

-        Information desired:

-        Inform of changes:

**Librarian assistant:** *(inherits from Educator)*

Goals:

Assist patrons

Preconditions:

Registered as assistant user

Main tasks:

Checkout books

Check in books

Scan requested books for inter-library transfer

Reshelving

**Educator:** *(inherits from student)*

-        Goals:

Reserve book for a semester

Restrict it to in-library use, checkout for 2 hours, cannot leave library

-        Preconditions:

Registered educator patron

-        Main tasks:

Reserve books

-        Exceptions:

-        Variations:

-        Information – acquire; produce; change:

-        Inform about external changes:

-        Information desired:

-        Inform of changes:

**Students:** *(inherits from public patron)*

-        Goals:

Get books

-        Preconditions:

Registered patron

-        Main tasks:

Search for a book

Check out a book

Return a book

Place a hold on a book

Renew checkout

Request summit book

-        Exceptions:

Can’t check out a book if you have overdue books

Can’t check out reserved books

-        Variations:

-        Information – acquire; produce; change:

-        Inform about external changes:

-        Information desired:

-        Inform of changes:

-

Public patron: (base case)

-        Goals:

Look up a book

-        Preconditions:

-        Main tasks:

-        Exceptions:

-        Variations:

-        Information – acquire; produce; change:

-        Inform about external changes:

-        Information desired:

-        Inform of changes:

**Database Server:**

-        Goals:

Track books

-        Preconditions:

-        Main tasks:

Send notifications

-        Exceptions:

-        Variations:

-        Information – acquire; produce; change:

-        Inform about external changes:

-        Information desired:

-        Inform of changes:

1. Use Case Template for “Checkout a book”

|  |  |
| --- | --- |
| Primary Actor: | Patron |
| Goal in context: | Check out a book to read at leisure outside the library |
| Preconditions: | Patron has found a book they want to check out |
| Trigger: | Patron decides to check out a book |
| Scenario: | 1. Patron looks up a book through the access terminal 2. The book availability is determined 3. Book location is provided to patron 4. Patron locates book in library and decides to check out book 5. Patron visits librarian assistant to checkout book 6. Librarian assistant authenticates patron with their Cougar Card 7. Librarian assistant checks out book to patron 8. Patron leaves with book |
| Exceptions: | Patron has overdue books, they are not allowed to check out more books |
| Priority: | High, to be implemented in first iteration |
| When Available: | During library hours |
| Frequency of use: | High, hundreds of transactions per day |
| Channel to Actor: | Physical access to library |
| Secondary Actors: | Librarian assistant, database server |
| Channels to Secondary Actors: | Physical access to library, access to database server |
| Open Issues: | None |

Use Case Template for “Return a book”

|  |  |
| --- | --- |
| Primary Actor: | Patron |
| Goal in context: | Return a checked out book |
| Preconditions: | A book has been checked out |
| Trigger: | Book is no longer needed |
| Scenario: | 1. Patron has finished using book, book is no longer needed 2. Patron goes to physical library and puts book in return book slot |
| Exceptions: |  |
| Priority: |  |
| When Available: | 24 hours through internal and external drop boxes |
| Frequency of use: | Hundreds of times daily |
| Channel to Actor: | Physical access to drop box |
| Secondary Actors: | Library |
| Channels to Secondary Actors: | Physical library |
| Open Issues: | None |

Use Case Template for “Place a book on hold”

|  |  |
| --- | --- |
| Primary Actor: | Patron |
| Goal in context: | Prevent a needed book from being checked out until patron can check out book |
| Preconditions: | Book is available  User is registered and authenticated to online system |
| Trigger: | Book has been located in online resource, patron needs the book but is not at the physical library |
| Scenario: | 1. Patron looks up book online 2. Book availability is determined to be available 3. Patron clicks a button on the web page to indicate they want to check out the book 4. A Hold is placed on the book so that if someone other than the registered patron tries to check it out, the librarian assistant will not check it out to them |
| Exceptions: | Patron has overdue books and is ineligible to check out books |
| Priority: | High, |
| When Available: | 24 hours via online interface |
| Frequency of use: | Daily, multiple times |
| Channel to Actor: | Via web interface |
| Secondary Actors: | Database server |
| Channels to Secondary Actors: | Via web interface |
| Open Issues: | None |

4. Use Case Template for Request a book from a different library

|  |  |
| --- | --- |
| Primary Actor: | Patron |
| Goal in context: | To check out a book that is not available at the current working library but is available at one in the area. |
| Preconditions: | Current working library must not have the book available  Different library in the area has book |
| Trigger: | Patron needs book, not at current library. System will check if at a local library. |
| Scenario: | 1. Patron looks up book 2. Book is unavailable at current working library 3. Patron requests book from local library |
| Exceptions: | Patron has overdue books and is ineligible to check out books. |
| Priority: | High |
| When Available: | 24 rs available online |
| Frequency of use: | Daily, multiple time |
| Channel to Actor: | Via web interface |
| Secondary Actors: | Database server |
| Channels to Secondary Actors: | Via web interface |
| Open Issues: | none |

5. Use Case Template for Send notifications

|  |  |
| --- | --- |
| Primary Actor: | Database Server |
| Goal in context: | Send patrons any information about overdue fees, dates, or other important information. |
| Preconditions: | There must be an alert in the system |
| Trigger: | If there is any alert and the date and time meets. |
| Scenario: | 1. Patron has checked out book 2. Patron has 2 days left to turn in book or renew, etc |
| Exceptions: | None |
| Priority: | High |
| When Available: | 24 Hours online |
| Frequency of use: | High, Hundreds or thousands everyday |
| Channel to Actor: | Online |
| Secondary Actors: |  |
| Channels to Secondary Actors: |  |
| Open Issues: |  |

6. Use Case Template for Renew checkout

|  |  |
| --- | --- |
| Primary Actor: | Patron |
| Goal in context: | Allow Patrons to extend the allowed time they have to borrow a book |
| Preconditions: | Patron has a book or media already checked out. Patron must also not have any overdue charges. |
| Trigger: | Patron selects that they would like to renew a book already checked out. |
| Scenario: | 1. Patron has a book or media checked out 2. Patron chooses that he/she would like to extend book. 3. Patron has no outstanding charges or overdues 4. System extends book due date |
| Exceptions: | 1. Patron has overdue charges or books |
| Priority: | High |
| When Available: | Must be in person during library hours with book or media in hand. |
| Frequency of use: | Hundreds of times per day |
| Channel to Actor: | Web interface |
| Secondary Actors: | Database server |
| Channels to Secondary Actors: | Web interface |
| Open Issues: | None |

7. Use Case Template for Search for book

|  |  |
| --- | --- |
| Primary Actor: | Librarian  Librarian assistant  Student  Educator  Public patron  Database server |
| Goal in context: | To find a book given a certain parameter or set of parameters |
| Preconditions: | Database must be populated |
| Trigger: | Person is looking for book |
| Scenario: | 1. Person accesses website  2. Person enters parameters  3. Web portal populates list of matching books |
| Exceptions: | Database is empty  Invalid input |
| Priority: | High |
| When Available: | Whenever database is available |
| Frequency of use: | High |
| Channel to Actor: | Whenever database is available |
| Secondary Actors: | None |
| Channels to Secondary Actors: | None |
| Open Issues: | None |

8. Use Case Template for Add a book

|  |  |
| --- | --- |
| Primary Actor: | Librarian  Database server |
| Goal in context: | To add a new book to library database |
| Preconditions: | Librarian has physical copy of book |
| Trigger: | Librarian receives new shipment of books |
| Scenario: | 1. Librarian receives shipment of books 2. Librarian logs into web portal 3. L accesses add book function 4. L inputs book information or selects existing book 5. L inputs number of books received 6. Web portal sends message to update database server 7. Web portal outputs barcodes 8. L prints and attaches barcodes |
| Exceptions: | L is unable to log in |
| Priority: | High |
| When Available: | When librarian is available |
| Frequency of use: | Whenever shipment is received |
| Channel to Actor: | Web portal |
| Secondary Actors: |  |
| Channels to Secondary Actors: |  |
| Open Issues: |  |

9. Use Case Template for Remove a book

|  |  |
| --- | --- |
| Primary Actor: | Librarian (L)  Database server |
| Goal in context: | To remove a book from the database |
| Preconditions: | L must have physical book |
| Trigger: | L wants to remove book |
| Scenario: | 1. L receives damaged book 2. L logs in to web portal 3. L accesses book removal function 4. L scans barcode 5. Web portal sends message to database server to delete book |
| Exceptions: | Book has already been removed  Book is missing/lost |
| Priority: | High |
| When Available: | When librarian is available |
| Frequency of use: | High |
| Channel to Actor: | Web portal |
| Secondary Actors: | None |
| Channels to Secondary Actors: | None |
| Open Issues: | What to do if book is missing/lost? |

10. Use Case Template for **Reshelving**

|  |  |
| --- | --- |
| Primary Actor: | Librarian  Librarian Assistant |
| Goal in context: | Print out the list of books needing to be reshelved, marking as they are shelved. |
| Preconditions: | Books must have been returned or brought to desk and added to physical reshelf location and to a reshelving list in the software. |
| Trigger: | Either the temporary location is full or at the end of the day the assistant shelves the books. |
| Scenario: | 1. Patrons periodically return books and/or take them off of shelves and return to front desk. 2. Librarian (Assistant) adds book to reshelving list and places book in reshelving location.. 3. Either the end of day is reached or the reshelving location reaches maximum capacity. 4. Librarian (Assistant) enters reshelving interface and looks at list of books. 5. Takes locations listed and shelves book to their appropriate locations.  Scan each book as it is shelved, removing each book from list. |
| Exceptions: | No books have been returned, no list is printed |
| Priority: | Low Priority - All other functions do not rely on reshelving.  Books can be reshelved without the software. |
| When Available: | During library hours. |
| Frequency of use: | A few times per day. |
| Channel to Actor: | Any access terminal, library or internet. |
| Secondary Actors: | None |
| Channels to Secondary Actors: | None |
| Open Issues: | Will people lose the book in the temporary shelving? |

11. Use Case Template for **Reserving a Book**

|  |  |
| --- | --- |
| Primary Actor: | Librarian  Librarian Assistant  Students  Educators |
| Goal in context: | Reserve a book so that it is available for use when you want to use it |
| Preconditions: | Book is available for checkout (then a hold is placed) |
| Trigger: | A book is needed by an educator. |
| Scenario: | 1. Educator needs to use a book periodically for class/research. 2. They then reserve a book for a certain amount of time for use when they need it. |
| Exceptions: | Book is unavailable. |
| Priority: | Medium Priority - Needed for educators but everything will work fine without. |
| When Available: | When a book is available. |
| Frequency of use: | A few times per semester. |
| Channel to Actor: | Access terminal over internet or library. |
| Secondary Actors: | None |
| Channels to Secondary Actors: | None |
| Open Issues: | None |

12. Use Case Template for **Book Availability**

|  |  |
| --- | --- |
| Primary Actor: | Any Human Actor |
| Goal in context: | Get whether a book is available and in what location. |
| Preconditions: | The database has been filled. |
| Trigger: | An actor has interest in the availability of a book. |
| Scenario: | 1. Patron needs a book. 2. Patron goes onto website. 3. Patron enters search parameters. 4. Clicks on appropriate book. 5. Checks the availability. |
| Exceptions: | None |
| Priority: | High Priority - Main function of library is to access books |
| When Available: | All times. |
| Frequency of use: | Many times daily. |
| Channel to Actor: | Any access terminal. |
| Secondary Actors: | None |
| Channels to Secondary Actors: | None |
| Open Issues: | None |