Use-case Descriptions:
1. LOGIN

Use-case Name:	Login
Actors:	Student, Lecturer, Librarian, Manager
Pre-condition:	There is an Internet connection and the user has accessed the library management system index page.
Post-condition:	The user is logged-in to correct page according to his/her role; and his/her username is displayed on upper-right corner of the page.
Summary:	Allow users to login into system to perform unique functions. Example: A lecturer logs in to suggest on which book to buy.
Basic Course (BC):	1. The user accesses the login screen 2. The use-case is activated when the user presses the login button after entering the username and password. 3. The system validates the username and password 4. If the credentials entered were correct the user is taken to the next page (Otherwise **EP.1** or **EP.2** or **AP.1**). The next-page may be a Student home-page, a Lecturer home-page, a Librarian's home-page, an Administrator's home-page or Manager's home-page depending on the one who logged in. (**Hint:** To create separate login links for each user).
Alternative Paths (AP):	**AP.1 Forgot my password** *1.* User clicks on the link: *"forgot password"* 2. System sends password reset link to user's email. 3. User is displayed with a page that tells him/her that a password reset link has been sent to his/her email. (Go to **AP.2**) **AP.2 User clicks password reset link** 1. User navigates to his/her email and clicks the password reset link. (**EP.3** Password reset link has expired) 2. The user is diplayed with a page to reset his or her password. 3. User confirms new password.
Exception Path (EP):	**EP.1 Username not found** 1. The system could not find the username *2.* Return to step 1 of BC and display error message: *"Username does not exist"* **EP.2 Username/ Password do not match**

	1. The username and password did not match *2.* Return to step 1 of BC and display error message: *"Incorrect username or password"* **EP.3 Password reset link has expired** *1.* User clicks on the link: *"re-send link"* 2. The user receives another email with password-reset link
System Rule (SR):	**SR.1 Username and Password constraints** 1. Students and lecturers will login with ARIS credentials. 2. For other users, the *username* will be their *payroll numbers* and *password* should consist of atleast 7 alphanumeric characters and atleast 1 non-alphanumeric character. 3. Each user must have a unique username.

2. ADD BOOKS

Use-case Name:	Add books into the database
Actors:	Administrator
Pre-condition:	Logged in as Administrator
Post-condition:	The book(s) is inserted
Summary:	Allows addition of books into the database
Basic Course (BC):	1. The administrator clicks the link/button: *"Add books"* from the side navigation bar. 2. The admin chooses between two options: *add a book* (adding one book at a time) and *add multiple books* (adding multiple books from an excel file). 3. If the admin chooses to add one book at a time. He/she is displayed with a form to enter the book's metadata. (else **AP.1**) 4. The admin clicks *submit* button to insert the book into the database. 5. The data are validated; if no validation errors, admin receives confirmation of successful insertion. (else **EP.1**) 6. Admin clicks *refresh* button to add the next book.
Alternative Paths (AP):	**AP.1 Add multiple books** 1. Admin is displayed with an interface to upload an excel file. 2. Admin navigates to his/her computer and selects the file to upload. 3. The file is uploaded. (incase; **EP. 3 or EP. 4**)

	4. Data are validated, if all information in the file pass, the books are inserted into the database. (else **EP. 2**) 5. Admin receives confirmation of successful insertion.
Exception Paths (EP):	**EP.1 Validation errors (adding single book)** 1. Admin is displayed with an error that book could not be inserted due to invalid or missing information, with the field(s) causing errors indicated. *2.* Admin corrects the error(s) and clicks *submit.* **EP. 2 Validation errors (adding several books)** 1. Admin is displayed with an error that books could not be inserted due to invalid or missing information in the file. 2. Admin corrects the errors in the file and uploads again. **EP. 3 Error when uploading** 1. Admin is displayed with an error message that the file could not be uploaded due to problems such as network. 2. Admin investigates and corrects suspected problems and re-tries to upload. **EP. 4 Canceling uploading** 1. Admin clicks *cancel* button to cancel uploading the file. 2. The file stops being uploaded. **3.** Return to **AP. 1 step 1**.
System Rule (SR):	**SR. 1 Fields for creation of a book entry** • Bar code (mandatory), Title (mandatory), Author (mandatory), Publisher (mandatory), Year published (mandatory), ISBN (mandatory), Pages (mandatory), Subjects, Keywords (mandatory) and available copies (mandatory) **SR. 2 Validation rules** • Each book must have a unique identification number (bar code).

3. UPDATE/ DELETE BOOKS

Use-case Name:	Update book metadata/ Delete book from the database
Actors:	Administrator
Pre-condition:	Logged in as Administrator
Post-condition:	The book is updated or deleted from the database
Summary:	Allows books to be updated or deleted
Basic Course (BC):	1. Admin clicks on *"Update/Delete Book" link/ button* from the side navigation menu. 2. *Update/Delete book page/ interface* opens. 3. The admin enters the ISBN/ Title of the book to be viewed on the search box provided and clicks search. 4. The search result is displayed with three buttons attached to the result: *Delete, Update* and *Cancel* buttons. 5. If admin clicks *update button* from the displayed result . The admin is displayed with a form containing current metadata of the book. He or she may edit/update any part as required. (else **AP. 1 or EP. 1 or EP. 2**) 6. The admin clicks *update* to persist the changes to the database. (else **AP. 2**) 7. Admin is given a confirmation of successful update operation.
Alternative Paths (AP):	**AP. 1 Delete button is clicked** 1. A dialog is displayed asking the user to confirm deletion. 2. If admin clicks *okay.* The book is deleted from the database and user is given feedback of successful deletion. *3.* User clicks *okay.* **4.** Admin is taken back to fresh page of **step 2 of BC.** **AP. 2 Cancel button is clicked** 1. Admin clicks on cancel button instead of update button. Therefore the changes will not be persisted to the database. And he/she is taken to **step 2 of BC above** (In other words he will be taken back to the *Update/Delete book page*).
Exception Paths (EP):	**EP.1 No results were found** 1. Admin may have wrongly typed the ISBN/ Title and thus a feedback: *"No*

	results were found" is provided to the admin. **2.** Admin clicks on *cancel button* and he/she is taken back to **step 2 of BC.** **EP. 2 Cancel button is clicked** **1.** The admin is loaded with a fresh page of **step 2 BC.**
System Rule (SR):	• New information must obey existing constraints for the *book entity* for the update to be successful.

4. SEARCH/ VIEW BOOKS

Use-case Name:	Search/View books
Actors:	Student, Lecturer, Administrator, Librarian and Library Manager
Pre-condition:	User has accessed the *CoICT Library* index page (which is also the *opac* page)
Post-condition:	User successfully searches and views the books
Summary:	Allows users to search for books and view the results
Basic Course (BC):	1. User clicks the *search* button from the side navigation menus to bring in the search interface 2. User enters information about the book such as *title, author* or *keywords from the title* and clicks search (**EP. 1**) 3. User is displayed with the results 4. User selects the correct result 5. User is displayed with some important meta-data of the book such as: title, author, year, publisher, its loan category, where to find it and whether it is currently available or not. 6. User clicks *okay* or *cancel* 7. User can go to *step 2 above* to search for another book
Alternative Paths (AP):	-
Exception Paths (EP):	**EP. 1 No matches were found** 1. User is displayed with the message: *"no matches were found"*. This could be a result of a typing error or the book could be unavailable in the catalogue 2. User clicks *okay* or *cancel* 3. User can go to **step 2 of BC** to search for another book

System Rule (SR):	-

5. VIEW REPORTS

Use-case Name:	View Reports
Actors:	Student, Lecturer, Librarian, Administrator and Manager
Pre-condition:	The users have logged in to their respective home pages
Post-condition:	Users can view or read the reports
Summary:	Allows authorized library users to view and read the periodic library reports
Basic Course (BC):	1. User accesses the opac (index) page 2. User logs in from the opac page 3. User accesses the home-page corresponding to his or her role 4. User clicks the *"view report"* link/button from the side navigation menus 5. User is taken to *report* page 6. User reads/views the report
Alternative Paths (AP):	-
Exception Paths (EP):	-
System Rule (SR):	-

6. RESERVE A BOOK

Use-case Name:	Reserve a book(s)
Actors:	Student, Lecturer
Pre-condition:	They have logged-in to their respective home pages
Post-condition:	They have reserved books of their choices
Summary:	Allows students and lecturers to reserve books. Users will be notified by email or text SMS when the book is returned.
Basic Course (BC):	1. User (Student/Lecturer) has accessed opac (index) 2. User has searched for the book on the opac 3. User checks the search result for the book and finds that the book is currently unavailable, meaning that all copies have been borrowed by other users 4. User logs-in (to be able to reserve the book) 5. User clicks on *"reserve book"*

	link/button from the side navigation menus 6. User is taken to the *reserve* page. 7. User searches for the book using the search functionality provided 8. User clicks *reserve* button on the search result containing the book of preference
Alternative Paths (AP):	-
Exception Paths (EP):	-
System Rule (SR):	-

7. RENEW A BOOK

Use-case Name:	Renew a book
Actors:	Librarian (For now lecturer acts as librarian)
Pre-condition:	• The book can be renewed iff either it was not reserved or enough copies of it are available to meet all reserve requests without compromising the renew request • Logged in as Librarian (*or lecturer for now)*
Post-condition:	Successfully renewed a book or not depending on the first pre-condition above
Summary:	Allows students and lecturers to renew books
Basic Course (BC):	1. Librarian logs in to perform renew for the user 2. Librarian clicks on *"search"* button 3. Librarian enters the *"id_no"* of the user and clicks search. 4. Librarian selects the user from the search result. (else **EP. 2**) 5. The user is displayed plus all the books he or she holds 6. Librarian clicks *renew* on the book that needs to be renewed 7. If no reserves are compromising, renewal will be successful (else **EP. 1**) 8. Librarian is notified of success or not and feedback is provided to user accordingly 9. Librarian clicks *okay* and is taken to **step 2 above** to renew more books if any.
Alternative Paths (AP):	-
Exception Paths (EP):	**EP. 1 Compromising reserves** 1. There are compromising reserves, system notifies the librarian. 2. Librarian clicks *okay* and is taken to **step 2 of BC**

	EP. 2 No results were found 1. Librarian wrongly typed the *id_no* of the user. The message: "*no matches were found"* is provided 2. The Librarian clicks *cancel* and is left at **step 2 of BC**
System Rule (SR):	-

8. LEND BOOKS

Use-case Name:	Tie books to users (Lend books)
Actors:	Librarian
Pre-condition:	• Logged in as Librarian • Borrower has been found in the system
Post-condition:	Book successfully tied to user
Summary:	Allows librarians to tie books to users
Basic Course (BC):	1. User goes to librarian to borrow a book 2. Librarian clicks *search* button 3. Librarian enters the *id_no* of the user into the search box and clicks search 4. Librarian selects the user from the search result 5. Librarian enters the book to be tied to user 6. Librarian clicks the *lend* button 7. Librarian is notified of successful process 8. Librarian clicks *okay* 9. Librarian can return to step 2 above to tie more books 10. If no more books to be tied, librarian clicks *okay* and is finished with the user
Alternative Paths (AP):	-
Exception Paths (EP):	-
System Rule (SR):	• Database to be updated at the beginning of each new academic year, to ensure that only valid users continue to have access to the system services. Example: Fetching from ARIS student and lecturer information each new academic year.

9. UNTIE BOOKS FROM USERS

Use-case Name:	Untie books from users
Actors:	Librarian
Pre-condition:	Logged in as Librarian
Post-condition:	Successfully unties books from users
Summary:	Allows librarian to untie a book from the user who returns it
Basic Course (BC):	1. User returns book to Librarian 2. Librarian logs in 3. Librarian clicks search button to initiate search process 4. Librarian enters the *id_no* of the user to search for him or her 5. Librarian selects the user from the search result 6. Librarian clicks "*untie"* button on the particular book to untie it from the user (Also the circulation module is updated to reflect the change. Example: If book was unavailable before, then now it is changed to available) 7. Librarian receives confirmation of successful untie and updating 8. Librarian clicks *okay*
Alternative Paths (AP):	-
Exception Paths (EP):	-
System Rule (SR):	• Librarian must verify that the bar-code of the book being returned matches with the bar-code that was tied to the user earlier.

10. SUGGEST BOOK(S) TO BUY

Use-case Name:	Suggest book(s) to buy
Actors:	Lecturer
Pre-condition:	Logged in as Lecturer
Post-condition:	Successfully suggests a book to buy
Summary:	Allows Lecturers to suggest important books to buy
Basic Course (BC):	1. Lecturer clicks on *suggest* menu from the side navigation menus. 2. The user is displayed with a form where he or she will be required to fill in important information about the book he or she is suggesting should be bought.

	Such important information includes: *Title of the book, ISBN, Edition, Year, Author, Co-author, Publisher* 3. User clicks on *submit* button to submit the suggestion 4. The form input are validated and if there are no errors the data are saved to the database and a notification of success is sent to user (else **EP. 1**) 5. User (Lecturer) clicks *okay* 6. User may return to step 2 to continue suggesting
Alternative Paths (AP):	-
Exception Paths (EP):	**EP. 1 Form input errors** 1. User is displayed with input errors on specific form fields 2. User corrects errors 3. User clicks submit
System Rule (SR):	-

11. SEND MEETINGS ALERTS

Use-case Name:	Send meetings alerts
Actors:	Library manager
Pre-condition:	Logged in as library manager
Post-condition:	Successfully sends meetings alerts
Summary:	Allows library managers to send meetings alerts to library staff (librarians and administrators) . The alerts will be sent as email/ text SMS notifications to the staff.
Basic Course (BC):	1. Library manager clicks on *send meeting alert* link from the side navigation bar. 2. Manager is displayed with the form to fill in important information about the meeting. The information include: day, time, venue and the agenda of the meeting. 3. Manager clicks *submit* button 4. Form data are validated 5. If everything goes well a notification email/text SMS is sent to library staff. Also the manager receives success confirmation. (else **EP. 1**)
Alternative Paths (AP):	-
Exception Paths (EP):	**EP. 1 Form validation errors** 1. User is displayed with errors on specific

	form fields 2. User corrects the errors 3. User clicks submit
System Rule (SR):	-

12. SEARCH/ VIEW USERS

Actors: Librarian, Administrator

Coming soon

13. PROVIDE STUDENT/ LECTURER DETAILS

Actors: ARIS

Coming soon