1. **Some combinations should clearly not be allowed e.g. an undergraduate in a Ph.D degree program.**

We have stored a list of all possible valid combinations in a table called Program. For any new student, only these combinations will be shown in the drop down while entering the details.

1. **Restricting course reserved materials to only appropriate students**.

We are handling this by showing only the list of books that the student is allowed to request when he goes to the publication module. The data is fetched by joining a few table:

Enrolled, Course, BlockBook, Student

1. **Book may be reserved for a course by a faculty member for a period up till 4 months.** We have created a separated module which is accessible only by faculty in which the faculty can block books for the course he/she teaches. Course table has a list of books assigned to each course, so only those books are shown to the faculty.
2. **Reserved books can be checked out for maximum of 4hrs and by only students of the class for which the book is reserved.**

Firstly the student is able to view only books which are restricted by its enrollment, as mentioned above. The checkout duration is actually calculated by looking up a table called Checkout, which has a list of check out durations in various scenarios. The calculation of due time/date is done by a **trigger** which is triggered each time an entry is made in the reservation table.

1. **Calculation of all due dates for all resources.**

The checkout duration is actually calculated by looking up a table called Checkout, which has a list of check out durations in various scenarios. The calculation of due time/date is done by a **trigger** which is triggered each time an entry is made in the reservation table.

1. **Waitlist for the resources.**

We have created separate tables for each type of resources which maintains the waitlist and clears the waitlist as and when needed.

1. **Faculty always have priority on the wait queue of publications.**

We maintain all patrons in the same waitlist queue, but before clearing any student’s waitlist, we check if there is a faculty in the same queue. If there is a faculty we send the notification to the faculty.

1. **Late Fee**

We have two tables, lateFeeReference and CalcLateFee. lateFeeReference stores the late fee calculation rules for each of the resources. CalcLateFee stores the actual calculated late fee against each reservation. Late fee gets calculated only when a patron returns the resources after comparing the check-in date and the duedate. All these late fee calculations are done using a **trigger**. As soon as there is an update in the reservation table with a checkin date the **trigger** runs and makes an entry in the CalcLateFee table in case there is a late fee applicable.

1. **Due date reminders**

We have created a **procedure** to generate all notifications related to upcoming duedates.

1. **Late fee reminders**

We have created a **procedure** to generate all late fee reminders.

1. **Account deactivation**

We have created a **procedure** to check all the patrons who have not cleared their dues for more than 90 days and this proc itself deactivated their account.

1. **A conference room can be booked only by faculty while a study room can be booked by both student and faculty.**

We do not allow such a request at the interface itself.

1. **If patron is not authorized for room type, an error message is presented.**

We do not allow such a request at the interface itself.

1. **If a room is not “checked out” by 1 hour after reserved start time, the reservation is automatically cancelled.**

We have created a **procedure** for the same.

1. **Camera reservation confirmation**

We have created a **procedure** which runs on every Friday and checks if the requested camera is returned and the first person in the queue gets a notification.

If the camera is not returned, then the reservations for the camera are cancelled by the procedure and the patron at que pos 1 is informed.

1. **If a patron fails to checkout the camera by 10am**.

We have created a **procedure** for the same. If the camera is not checked out by 10am, then the second person in the queue get the notification to collect the camera and the first person gets a notification that his/her reservation is canned.