DESIGN DOCUMENT

This is a library management system desktop application which serves the purpose of managing various following actions. The only user of this application is the librarian who is allowed to interact with the database and make appropriate changes after logging in with a specific username and password. The library database involved with the system consists of various tables such as :

- borrower(card_no, ssn, fname, lname, address, phone)
- book(isbn, title)
- author(author_id, fullnmae, title, fname, lname, mname, suffix)
- book-author(isbn, author_id)
- book_copies(book_id, branch_id, no_of_copies)
- library_branch(branch_id, branch_name, address)
- book_loans(loan_id, isbn, branch_id, card_no, date_out, due_date, date_in)
- fines(loan_id, fine_amt, paid)

Each table has their individual primary key and some foreign keys in order to maintain interaction between the tables.

- Graphical User Interface (GUI) and Overall Design

All interface with the Library (queries, updates, deletes, etc.) are done from a graphical user interface. The GUI application interfaces with the Library database via an appropriate MySQL connector. Initial database creation and population has been done from command line or other admin tool.

- Book Search and Availability

Using GUI, we are able to search for a book, given any combination of ISBN, title, and/ or Author(s).The query supports substring matching. We display the following in search results:

• ISBN • Book title • Book author(s) • Branch info - branch_id - branch_name

- Book Loans Checking Out Books

Using GUI, we are able to check out a book, given the combination of BOOK_COPIES(Isbn, branch_id) and BORROWER(Card_no), i.e. create a new tuple in BOOK_LOANS. A new unique primary key for loan_id is generated. The date_out is today’s date. The due_date is 14 days after the date_out.

- Each BORROWER is permitted a maximum of 3 BOOK_LOANS. If a BORROWER already has 3 BOOK_LOANS, then the checkout (i.e. create new BOOK_LOANS tuple) fails and returns a useful error message.

- If the number of BOOK_LOANS for a given book at a branch already equals the No_of_copies (i.e. There are no more book copies available at your library_branch), then the checkout fails and returns a useful error message.

Checking In Books : Using your GUI, we are able to check in a book and are able to locate BOOK_LOANS tuples by searching on any of book_id, Card_no, and/or any part of BORROWER name. Once located, there is a way of selecting one of potentially multiple results and a button (or menu item) to check in (i.e. enter a value for date_in in corresponding BOOK_LOANS tuple).

- Borrower Management

Using GUI, we are able to create new borrowers in the system. All name, SSN, and address attributes are required to create a new account (i.e. value must be not null). Query automatically generates new card_no primary keys for each new tuple that uses a compatible format with the existing borrower IDs. Borrowers are allowed to possess exactly one library card. If a new borrower is attempted withe same SSN, the system rejects and return a useful error message.

- Fines

fine_amt attribute is a dollar amount that is having two decimal places. paid attribute is a boolean value (or integer 0/1) that idicates whether a fine has been paid. Fines are assessed at a rate of $0.25/day (twenty-five cents per day). There are two scenarios for late books

1. Late books that have been returned — the fine is [(the difference in days between the due_date and date_in) * $0.25].

2. Late book that are still out — the estimated fine is [(the difference between the due_date and TODAY) * $0.25].If a row already exists in FINES for a particular late BOOK_LOANS record, then - If paid == FALSE, we do not create a new row, but just update the fine_amt if different than current value. - If paid == TRUE, we do nothing. In this mechanism for librarians to enter payment of fines (i.e. to update a FINES record where paid == TRUE)

– we do not allow payment of a fine for books that are not yet returned.

– The fines are grouped by card_no. i.e. SUM the fine_amt for each Borrower.

ASSUMPTIONS:

1. Used size of 13 because of ISBN upgradable(future scaling) to 13 and varchar as it may contain characters.
2. Used size of 100 as book names can be varying and taken maximum 100 as assumption. Title will always be present
3. Used as size will normally be 50 with 10 characters as buffer.
4. Standard size of 50 as name but max to be considered 50.
5. Pages will never exceed 9999.
6. Assumption max 99 branches.
7. Assumption name to be max of 50 characters.
8. Assumption address to be max of 100 characters.
9. Added an extra field of author_name in book_author table.