SA/SD

March 18, 2016

Part I
System Analysis

1 Feasability Study

1.1 Understanding the problem

In this section, we make an effort to understand the purpose of the software. The LIS(Library Information Software) is intended to ensure a systematic manner of maintaining the database of a library and easing the job of a librarian.The LIS will help maintain a record of books in the library and members registered with the library.The software aims at hassle-free handling of basic library functions like issue/reserve/return books as well as giving the freedom to the librarian to add and delete books and members to the database. Extending further the software can be used by the librarian to decide on disposing of books which have not been issued for a long time. This software also helps to keep track of statistics of issue of a particular book ver a period of time.Since everything is done on the computer, it is easy to record all data and there are minimal chances of inconsistencies or ambiguities.

1.2 Scope of the Problem

In this section,we list the various functions performed by the software:

- Add books to the database
- Modify existing records of books
- Delete unused books from database
- Add users to the database
- Modify existing details of users
- Remove user from the database

- Query regarding books on ISBN number or name

- Query regarding user on user id or name

- Record issue,return and reservation of books

- Calculation of penalty on overdue books

- Maintains statistics of all books

1.3 Analysing stakeholders

The various stakeholders are:

- Librarian

- Member

These stakeholders have been clearly described with their attributes and methods in the SRS.

Part II

System Design

6   Refined System Parameters

6.1   Global System Architecture

The overall system architecture is a 2-tier architecture which includes client

in the software being designed.

6.2   Platform

6.2.1   Hardware

6.2.2   Software

6.2.3   Networking

6.3   Software Architecture

Object-oriented architecture forms the basis of the LIS. In this style data representations and their associated primitive operations are encapsulated in an abstract data type or object. The components of this style are the objects—or instances of the abstract data types. Objects interact through function and procedure invocations. Two important aspects of this style are A. that an object is responsible for preserving the integrity of its representation (usually by maintaining some invariant over it), and B. that the representation is hidden from other objects. Thus the aspects of OOA mentioned justify our choice.

6.3.1   Details

6.3.2   Justification

7   Database Design

8   Design Details

8.1   Refinement of UML diagrams

8.2   Prototype Design

8.3   Design I/O procedures and user interfaces

8.4   Design of classes in target language

8.5   Exception Design

9   Adoptable Practices

10   Any Other Information Of previous Stages