The Project:

Library Automation System.

The Problem:

A city library in Kolkata planned to automate its’ library operations and member operations.

Library lends books, CDs and DVDs to its’ members. Library administrator should be able to add a book/CD/DVD with book/CD/DVD name, author name, category (Fiction, Entertainment, Technical, and Management etc.), number of copies.

Member should be registered with library before he is allowed to check out book/CD/DVD. Member needs to give name, date of birth, sex, address, contact number and gets a member id. Member id needs to be specified for any transactions to be done in library.

Search facility needs to be made available to members by book name or author name or category or combination of above to search the relevant resources.

Member can choose one or more book/CD/DVD to check out. Software needs to ensure the availability of the book before check out. Due date needs to be displayed once checked out. Due dates are 10 days, 5 days and 3 days for book, CD and DVD respectively. Member also should be able to see the list of books that she/he checked out. Also should be able to return the book that he/she checked out. When the member returns the book beyond the due date, he/she should be fined accordingly. Fines are Rs. 1/-, Rs. 2/- and Rs. 3/- for book, CD and DVD respectively.

Administrator should be able to see the list of books checked out or due or returned for a given date.

Objectives:

1. Library Administrator can add new resources to the library.
2. People can become new member of the library.
3. Only members can borrow resources from the library.
4. Members can search resources by book name or author name or category or combination of these.
5. Member can choose one or more book/CD/DVD to check out. Software needs to ensure the availability of the book before check out.
6. Due date needs to be displayed once checked out.
7. Member also should be able to see the list of books that she/he checked out.
8. Also should be able to return the book that he/she checked out. When the member returns the book beyond the due date, he/she should be fined accordingly.

Preliminary Ideas:

1. The library administration system is designed to cater to the needs of the library administrator and the members of the library.
2. This web based application can be used by the library administrator to add records of the resources and check the details of the resources that are borrowed and returned on a particular date.
3. This application is also built keeping in mind the needs of the members of the library. The library administrator on behalf of the members can search particular resources and provide them to the members. Apart from this he can also provide membership to the new customers visiting the library and provide them a member id. The administrator can inform the members of the number of resources borrowed by them and if any fine is due from them.

The Feasibility Study

As the initial investigation is completed, it leads to a more detailed investigation

of the system. The conclusions of the initial study become the input for the

detailed study. We can also refer to the feasibility study as the detailed study or

detailed investigation. Feasibility study is called like because as in the first phase,

we just check briefly about the problems related to the old system, & the need of

the new system. So, in this phase, that initial survey is further expanded to a more

detailed feasibility study.

The tasks performed during this phase were as follows:

(a) The user’s demonstrable needs are fulfilled.

(b) The availability of resources was being checked.

(c) The estimation required for the resources were achieved.

(d) The impact of the system on the organization has been check by placing a model in it.

Requirement Analysis and Specification

Objective To Be Fulfilled:

(a) Development of software in the given time.

(b) To create an effective and efficient application.

User Requirements:

(a) The application should provide a user friendly environment.

(b) The application should be easily understandable and reliable.

(c) The application should fulfill all essential facilities.

(d) The software being built must provide platform independent application.

Requirement Determination Techniques & System Analysis Method:

(a) User communication

(b) Team discussion

(c) Analysis of existing system

(d) Study of old projects and records.

# Design – High level and Low level

***Entity Relationship Diagram***

Conceptual modeling is a very important part in designing a proper database application. Here the term database application refers to a software product that has a central database containing data and some associated programs that operate on the data in the database to give desired results of our queries. In the Library Automation System problem we are representing conceptual model of the problem by the help of ER data modeling. An entity-relationship (ER) diagram is a specialized graphical representation that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

For our database application consider a LIBRARY AUTOMATION database that keeps track of all the books, CD’s and DVD’s in the library, along with that the LIBRARY AUTOMATION database also keeps details about the members of the library such as the name, date of birth, sex, address, contact number etc. The LIBRARY AUTOMATION database also keeps record of the resources of the library that has been lent to the various members of the library.

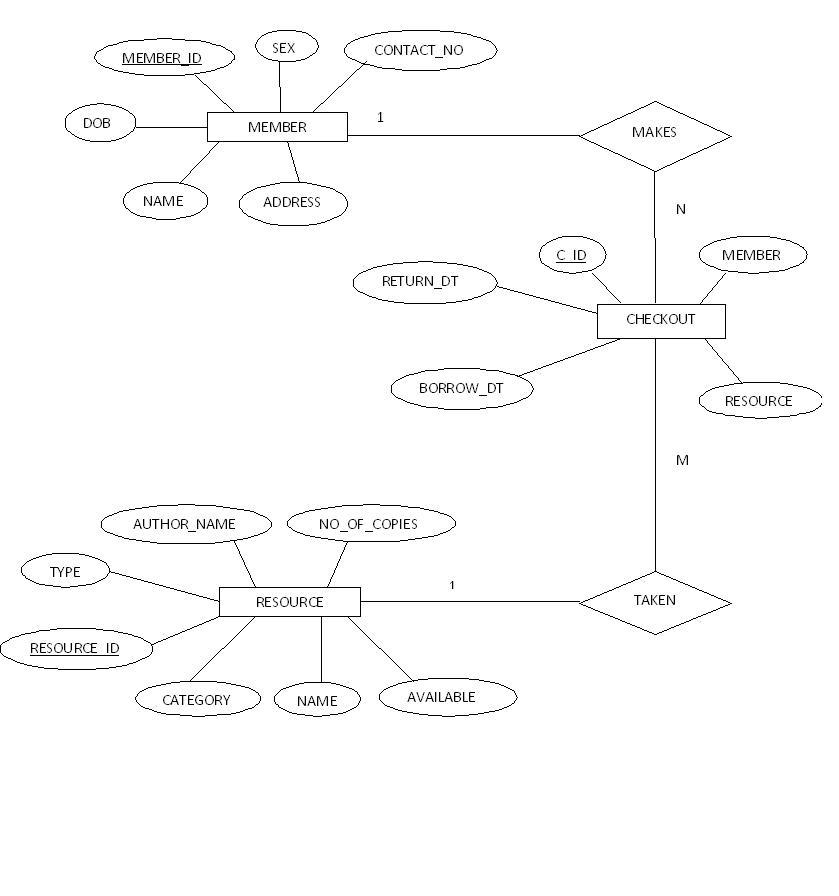
In our LIBRARY AUTOMATION database there are three entities namely: MEMBER, RESOURCE, and CHECKOUT.

1. MEMBER: This entity keeps record of all the members of the Library. Speaking in details this entity keeps the basic details about the members such as their name, gender, address, contact number, date of birth. The attributes of this entity are [NAME],[SEX],[DOB],[ADDRESS],[CONTACT\_NO],[MEMBER\_ID].
2. RESOURCE: This entity keeps information about the basic resources of the library which are books , CD’s and DVD’s. Each record in this entity is made up of all the information needed to categorize any resource in the library, starting with the name of the resource ,the resource id(a unique number to identify each instance of the resource separately), the category of the resource (i.e. Fiction, Entertainment, Technical, and Management etc.) ,the resource type(i.e. paperback, hard back, CD,DVD etc.), the name of the author or publisher or owner, the number of copies of that particular resource available in the library, whether it is still available or all the copies of that particular resource has been taken by members. The attributes of this entity are [RESOURCE\_ID],[NAME],[TYPE],[AUTHOR\_NAME],[CATAGORY],[NO\_OF\_COPIES],[AVAILABLE].
3. CHECKOUT: This entity keeps the vital information about the resources borrowed by the members of the library. It keeps information about the member who has borrowed a resource , the resource that has been borrowed, date it has been borrowed on ,date it must be returned. The attributes of this entity are [C\_ID],[MEMBER],[RESOURCE],[BORROW\_DT],[RETURN\_DT].

Now the above three entities are related to each other by the following relationships:

1. MAKES: The entities MEMBER and CHECHOUT are related by MAKES relationship. We know that when a member borrows a book, the relevant information is stored in CHECHOUT relation. As one member can borrow many books the relation MAKES is a 1: N relation.
2. TAKEN: The entities CHECKOUT and TAKEN are related by TAKEN relationship. The significance of TAKEN relation is that there are several copies of each resource in the library, thus different instance of the same resource can be borrowed by different members at the same time, thus the relation TAKEN is 1: M relation.

Entity Relationship Diagram:



**DATABASE DESIGN**

***Relational schemas and functional dependencies:***

1. **MEMBER(MEMBER\_ID, NAME, ADDRESS,DOB, SEX, CONTACT\_NO)**

This table contains the details about the members of the library system. The table consists of Name of the member [NAME], Gender [SEX], Date of Birth [DOB],Address of the member [ADDRESS],Contact number of the member [CONTACT\_NO],unique id for eacf member [MEMBER\_ID]. Here in this table the Primary key is [MEMBER\_ID].

This table has the following Functional Dependencies:

1. MEMBER\_ID -------> { NAME , SEX , DOB , ADDRESS , CONTACT\_NO }
2. **CHECKOUT(C\_ID, MEMBER, RESOURCE, BORROW\_DT, RETURN\_DT)**

This table contains the details of the resources that have been lent to the members. This table consists of checkout id [C\_ID], member id [MEMBER\_ID], resource id [RESOURCE\_ID], borrow date [BORROW\_DT], and return date [RETURN\_DT]. Here the primary key is [C\_ID].

This table has the following Functional Dependencies:

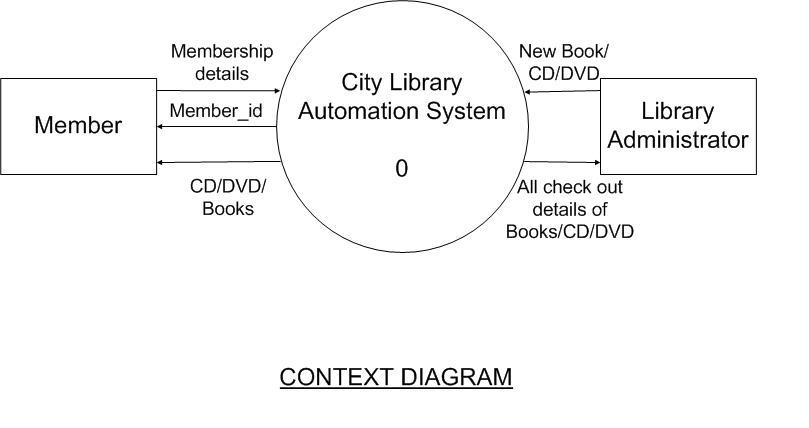
1. C\_ID 🡪 {BORROW\_DT, DUE\_DT, RETURN\_DT }
2. **RESOURCES(RESOURCE\_ID, NAME, TYPE, AUTHOR\_NAME, CATEGORY,NO\_OF\_COPIES,AVAILABLE)**

This table contains the details of the resources that form the collection of the library. This table consists of the resource id [RESOURCE\_ID], name of the resource [NAME], the resource type [TYPE], the name of the author or publisher or owner [AUTHOR\_NAME], the category of the resource [CATEGORY], availability info [AVAILABLE], number of copies of that particular resource [NO\_OF\_COPIES].

This table has the following Functional Dependencies:

1. RESOURCE\_ID 🡪 {NAME, TYPE, AUTHOR\_NAME, CATEGORY,NO\_OF\_COPIES,AVAILABLE }

**CONTEXT DIAGRAM**



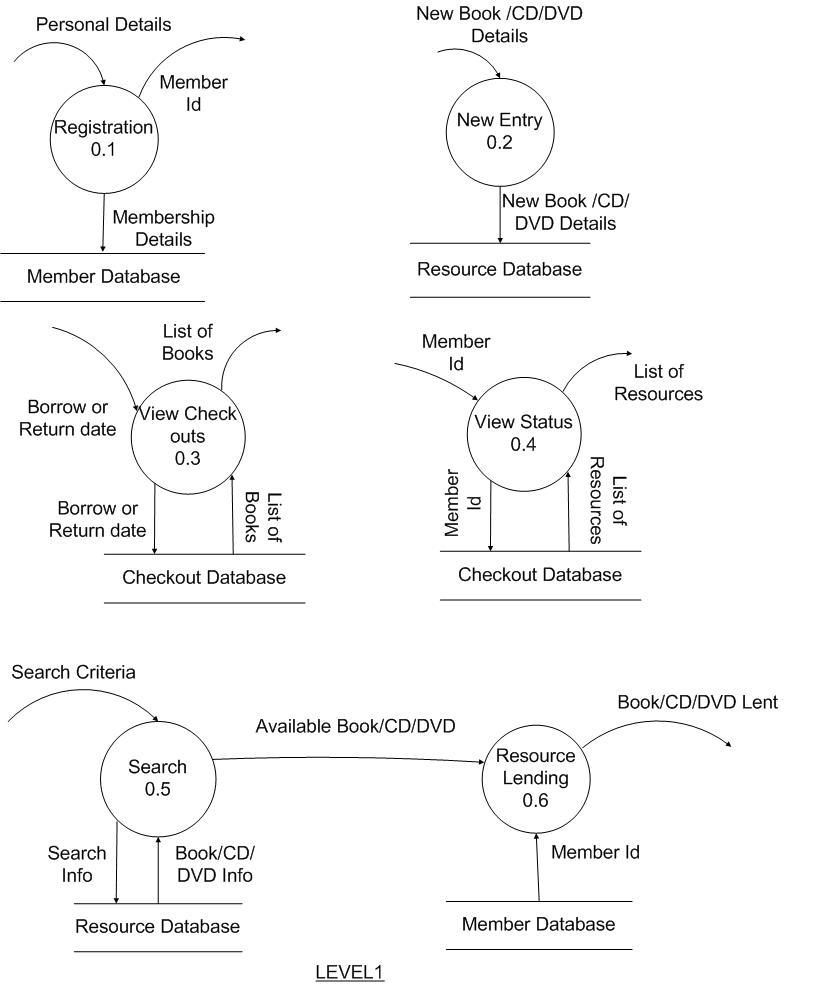
The Context diagram represents the entire system as a single module. The data values that are given to the system comprises of three types.

First of all the new resources (books, CD’s, DVD’s) may be added to the system which the library administrator does and also the library administrator can check out the details about all the resources in the system.

Secondly, a person may want to borrow resources from the library. For that he/she has to become a member of the library. Thus the person becomes a member by submitting his/her personal details to the system; in return the system assigns the person a unique Member\_id.

Thirdly, a member may want to borrow a particular resource from the library. The member submits the resource description and checks if that resource is available. If the resource is available then the member’s member id is taken and a instance of the resource is allotted in his name. If the resource is not available the member is informed of it. Apart from this the system also shows the details of all resources present in the database.

**DATA FLOW DIAGRAM**



The Level 1 of the Data Flow Diagram shows Context diagram broken down into six modules:

The *Registration* module takes the personal details about a person who wants to become a member of the library inserts the details in the member database and assigns the person a member\_id. The *New Entry* module takes in as input details of new resources that are added to the collection of the library by the library administrator, it inserts the details in the resource database. The *view checkouts* module takes in as input Borrow date or Due date or Return date, matches the inputs against the checkout database and shows the matching results, this module is mainly used to see the list of books that are currently with the members and various details like when and by whom it was borrowed or when it should be returned etc. The *View Status* module is used by the members to check out the resources borrowed by them and the total fine amount (if any) due from them for not returning resources on time. The *Search 0.5* module takes in as input the search criteria from the members and matches those criteria against the resource database, if any resource is found matching that criteria, then again with that information it is checked if it is available. If it is available then that information is passed to the *Resource Lending* module as input. The *Resource Lending* module takes the member\_id from the member database and does the needful to lend the resource to that member.