Project Design Report

[Library management database system]

[Amigo]

[11/21/2022]

Peiyu Li

Preethi Reddy Tera

Yagnashree Velanki

Introduction

Our project is to create a library management database system. The library is a vital part of each university, it provides students with various sources, for example, students can borrow books from the library, and there are study rooms for students to schedule group meetings for their final projects, etc. For better management of library resources, record students' borrowing and returning books, and better allocation of resources like library study rooms, we create an easy access and management database system. This database can be used to record the students' details, borrow and return books, the reservation of study rooms and lecture halls, and so on. We have 18 entities in total, and the details of each of them are described in the next section.

Some of the features or aspects that we have added in our design are:

We have created a separate entity called WIFI in which each member that comes into a library should enter that network using the password to log in to access library resources.

There is a settlement section, if the borrowed books or gadgets get damaged then a penalty will be issued to the user and the amount is reflected in the penalty section.

Membership is created for the people who are not in the organization and are willing to use library resources. For the people in the organization i.e., students/ staff organization id will be the membership for the library

All the employees working in the library cannot modify the details of books or students. Only a few employees working in admin positions have the authority to modify the details.

Requirements Analysis

Currently we are focusing on the college/university library management system. Our requirements are

1. *The availability of books or articles should be shown to students before they apply for a borrowing. The information (location) of books should be available to students so that they can find the books they need to borrow.*
2. *Students/staff having student/staff IDs should be members of the library and do not need any other memberships to use the library resources. But people from outside of the university can access the library if they have a membership card from the library.*
3. *The availability of study rooms, lockers, and lecture halls should be updated for each reservation. In addition, to avoid conflicts, resources like study rooms, lockers, and lecture halls need to be booked before use.*
4. *If the resources of the library like books or electronic gadgets are damaged when they are borrowed from the library, then there will be minimal charges on students/members as a penalty.*

other constraints:

- A person cannot borrow more than 5 books at a time. If he/she wants to borrow other books, then they need to return the borrowed items first.
- A person cannot borrow a book for more than 10 days. He/She has to reissue the book from the library.
- Penalty will be issued if the borrowed book is damaged or not returned on time
- Students can use the study rooms only if they make a reservation before using them, the start time and the end time should be stored, and students should leave after the end time.
- Same IDs for 2 or more books shall not be allowed.
- Users must enter the borrow and return date.

Data Requirements and Conceptual Design

Entities and Attributes:

Entity 1: Employee

This attribute defines the employees that are working in the library which can be both a student from the organization or a staff member of the organization

Attributes

- emp_Id: ID of each employee working in the library, type: varchar(10)
- ssn: Social security number of an individual, type: varchar(10)
- first_name: first name of the employee, type: char(20)
- Last_name: Last name of the employee, type: char(20)
- position: Post in which they are working, type: varchar(20)

Relationships

- *Relationship: A library employee can be a student or a staff member of the university/organization.

Primary Key

- Employees are identified by the emp_Id attribute because this attribute is uniquely defined so that each employee can be identified by emp_Id.

Entity 2: Student

This entity is used to store the details of the students who are also the users of the library.

Attributes

- stu_id: ID of each student that is in the organization/university, type: varchar(10)
- ssn: Social security number of an individual student, type: varchar(10)
- first_name: first name of the student, type: char(20)
- Last_name: Last name of the student, type: char(20)
- dept: Name of the department in which the student is studying, type: varchar(10)
- major: Major course in which the student is studying, type: varchar(10)
- DOB: date of birth of the student, type: Date.

Relationships

- *Relationship: A student/user can use the resources like Borrow, purchase, and reserve from the library

Primary Key

- students are identified by stu_id attribute because this attribute is uniquely defined for each student

Entity 3: Staff

This entity is used for storing the details of the staff present in the organization. They may be the persons working in the library or people working in other departments of the organization/University.

Attributes

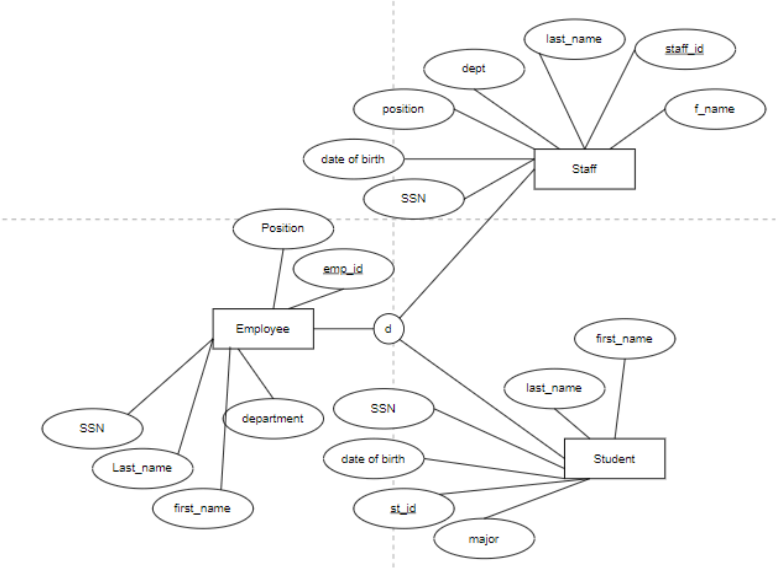
- staff_id: ID of each staff that is present in the organization/university is stored here, type: varchar(10)
- ssn: Social security number of each person working as staff, type: varchar(10)
- first_name: first name of the staff, type: char(20)
- Last_name: Last name of the staff, type: char(20)
- dept: Name of the department in which they are working, type: varchar(10)
- position: name of the position in which they are like a professor, manager, director, type: varchar(30)
- DOB: date of birth of the staff, type: Date.

Relationships

- *Relationship: A person from the staff can be a user of the library or can be the person working in the library.

Primary Key

- staff are identified by the staff_id attribute because this attribute is uniquely defined for each person. An employee in the library can also be considered staff.



Entity 4: Books

This entity is used to store the data of the books that are present in the library.

Attributes

- book_id: The ID of each book is unique for easy identification. , type: varchar(10)
- Name: Name of the book, type: varchar(30)
- Author: Name of the person who wrote the book, type: char(20)
- gener: Gener is classified as which field the book belongs to i.e., computer science, history, arts, etc. type: char(20)
- Published year: Year in which the book is published, type: Date
- edition: Edition of the book, type: varchar(5).
- publication name: Name of the publication which published the book, type: varchar(30)

Relationships

- *Relationship: This Entity is related to everyone i.e., a student/user or employee/staff in finding the required books.

Primary Key

- Books are identified by the book_id attribute because this attribute is uniquely defined for each book.

Entity 5: Journals/Articles/Magazines(JAM)

This entity is used to store the data of the Journals/ Articles/ Magazines that are present in the library.

Attributes

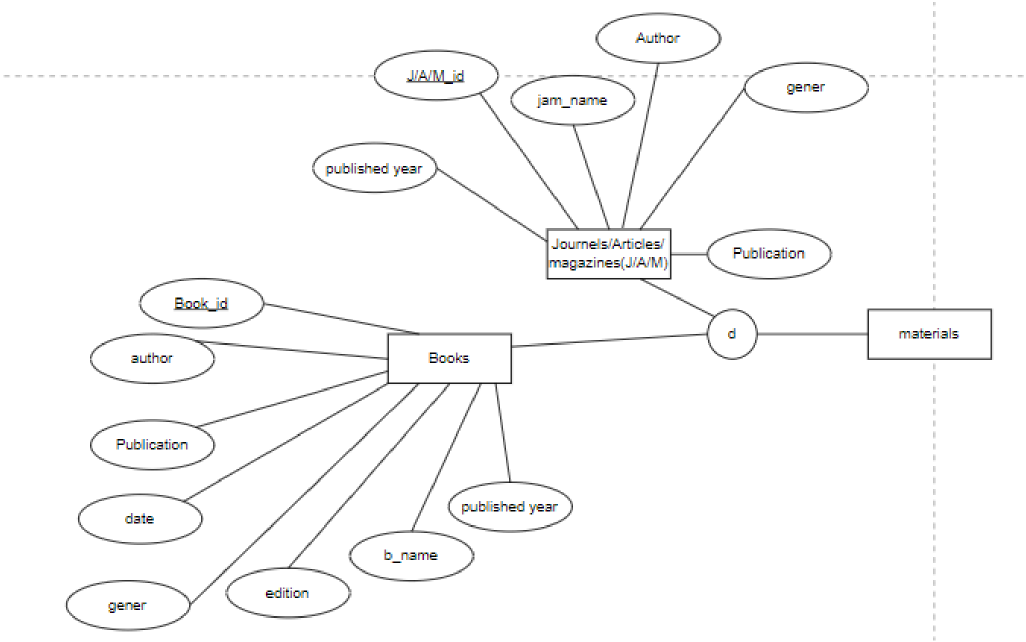
- jam_id: The ID of each article/ journal/ magazine is unique for easy identification. , type: varchar(10)
- Name: Name of the J/A/M, type: varchar(30)
- Author: Name of the person who wrote particularly J/A/M, type: char(20)
- gener: Gener is classified as which field the J/A/M belongs to i.e., computer science, history, arts, etc. type: char(20)
- Published year: Year in which the J/A/M is published, type: Date
- edition: Edition of the J/A/M, type: varchar(5).
- publication name: Name of the publication which published the J/A/M, type: varchar(30)

Relationships

- *Relationship: This Entity is related to everyone i.e., a student/user or employee/staff in finding the required Journal/Article/Magazines.

Primary Key

- J/A/M are identified by the jam_id attribute because this attribute is uniquely defined for each J/A/M.



Entity 6: StudyRoom

The study room is designed for students' group meetings, students can make a reservation before two days.

Attributes

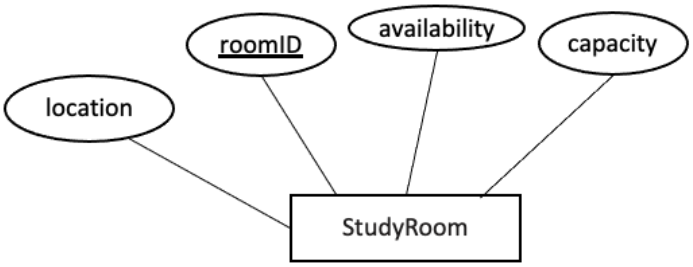
- roomID: the ID of each study room, type: char(6)
- location: where the study room is located, type: varchar(15)
- availability: whether the specific StudyRoom is being occupied or not, type: boolean
- capacity: the maximum number of users in each study room, type: Integer

Relationships

- *Relationship: the student entity can reserve and use the study room

Primary Key

- It is identified by roomID attribute because this attribute should be uniquely identified so that students can reserve study room by the roomID



Entity 7: Lab

There are several labs in the library, students can enter into the lab to use computers and printers in each lab.

Attributes

- labID: the ID of each lab, type: char(6)
- location: where the lab is located, type: varchar(15)

Relationships

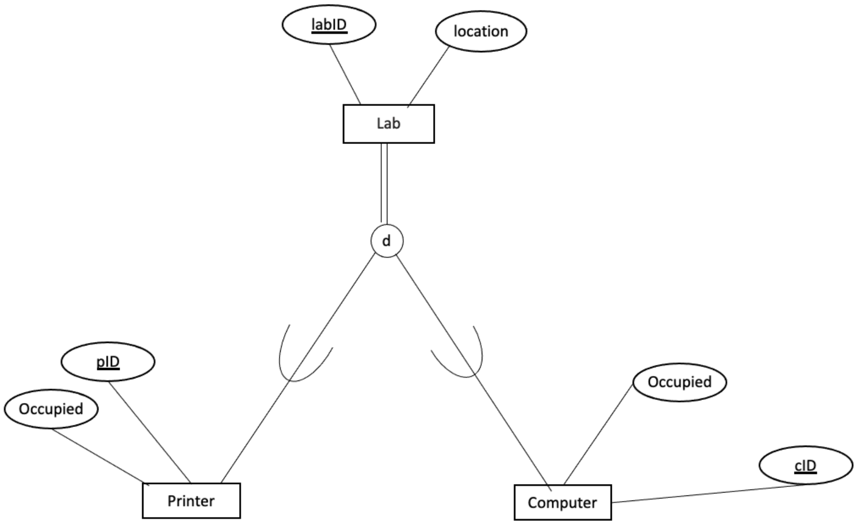
- *Relationship: the student entity can enter into the lab to use computers and printers in the lab

Primary Key

- It is identified by labID attribute because this attribute should be uniquely identified so that students can find the lab by the labID

There are computers and printers in each lab that can be used by students, each computer or printer has the id and occupied attributes: the id is the primary key that used to uniquely identify each computer or printer, and the occupied attribute shows whether the specific computer/printer is being occupied or not

Computer (cID, Occupied)
Printer (pID, Occupied)



Entity 8: Guests

This entity is for guests that are not one of the members of the university, to get access to the library sources, they should buy a membership.

Attributes

- name: the name of each guest, type: varchar(20)

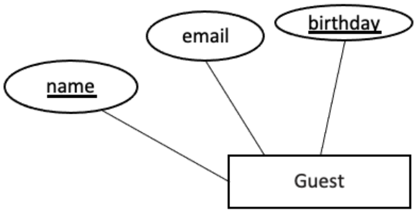
- email: each guest has an email address so that they can receive notification from the library to remind them of the expiration date of their membership, type: varchar(20)
- birthdate: the birthdate of each guest, type: Date

Relationships

- *Relationship: guests can buy membership

Primary Key

- It is identified by name and birthdate to avoid that people have the same name.



Entity 9: Memberships

This entity is used for guests that are not a member of the University, guests could buy a membership to get access to the library sources.

Attributes

- mID: the ID of each membership, type: char(10)
- username: each membership is held by a user, the user has a username, type: varchar(20)
- startDate: valid start date of the membership card, type: Date
- endDate: expire date, type: Date

Relationships

- *Relationship: guests can buy membership

Primary Key

- It is identified by mID attribute because this attribute should be uniquely identified so that guests can use their unique mID to enter the library.

Entity 10: Professors

This entity includes the information of professors, professors can get access to any source in the library.

Attributes

- name: the name of each guest, type: varchar(20)

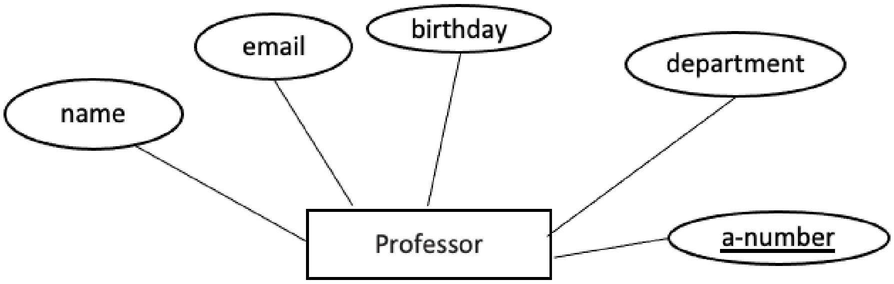
- A-number: a number that is used to identify each professor
- email: each guest has an email address so that they can receive notification from the library to remind them of the expiration date of their membership, type: varchar(20)
- birthdate: the birthdate of each guest, type: Date
- department: the department of each professor

Relationships

- *Relationship: professors can borrow books or reserve lecture halls

Primary Key

- It is identified by A-number.



Entity 11: LectureHall

This lecture hall is used for professors to provide some valuable talks to the public. Professors can make a reservation before two days.

Attributes

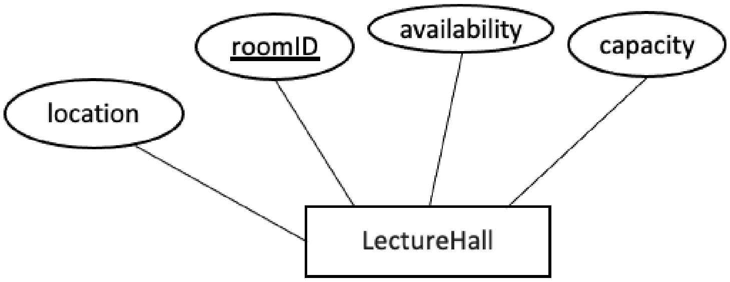
- roomID: the roomID of each lecture, type: char(10)
- location: where each lecture hall is located, type: varchar(15)
- availability: whether the specific lecture hall is being occupied or not, type: boolean
- capacity: the maximum number of users in each lecture hall, type: Integer

Relationships

- *Relationship: professors can reserve lecture hall

Primary Key

- It is identified by roomID attribute because this attribute should be uniquely identified so that professors can reserve their own lecture hall by roomID



Entity 12: WIFI

We have a separate network for library users only, users should log in the network before they use the WIFI.

Attributes

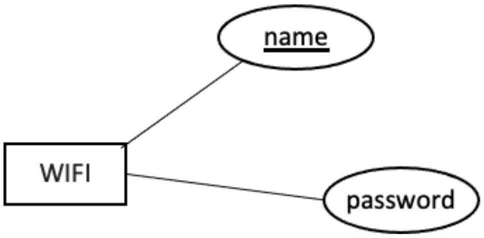
- name: the name of network, type: varchar(20)
- password: the password for users to log into the network

Relationships

- *Relationship: users can log into the network

Primary Key

- It is identified by name and password attribute .



Entity 13: Borrows (borrow_id, student_id, book_id, date)

The borrows entity has four attributes, the ID is the primary key which should be uniquely identified; the student_id is the foreign key which shows the Id of student who has borrowed the book; book_id is the foreign key which shows the Id of the book which has been borrowed; date shows the time stamp at which the book is borrowed.

Attributes

- borrow_ID: the borrow_ID of each book which has been borrowed, type: int
- student_id: the ID of the students who has borrowed the books, type: varchar(15)

- staff_id: the ID of the staff who has borrowed the books, type: varchar(15)
- mID: the ID of the guests who has borrowed the books, type: varchar(15)
- book_id: the ID of the book which has been borrowed, type: varchar(15)
- date: the timestamp at which the book has been borrowed, type: date

Relationships

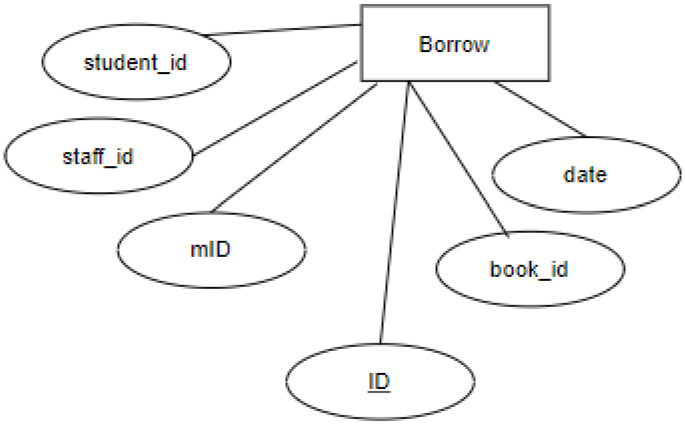
- *Relationship: person can borrow the books.

Primary Key

- It is identified by the borrow_ID attribute because this attribute should be uniquely identified.

Foreign Key

- book_id is the foreign key which is referred from the Books entity.
- student_id is the foreign key which is referred from the Student entity.
- staff_id is the foreign key which is referred from the Staff entity.
- mID is the foreign key which is referred from the membership entity.



Entity 14: Locker (locker_id, availability)
The locker entity has two attributes: locker_id, availability. The locker_id is the primary key which should be uniquely identified; the availability shows whether the specific locker is being occupied or not constantly.

Attributes

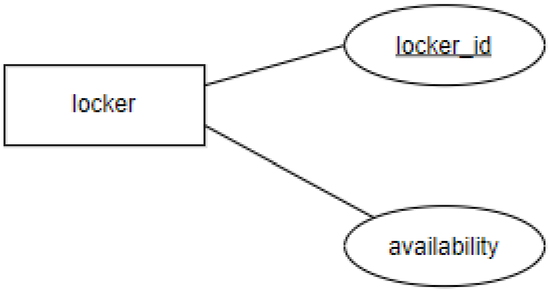
- locker_ID: the ID of each locker, type: int
- availability: whether the specific lecture hall is being occupied or not, type: boolean

Relationships

- *Relationship: person reserves the locker.

Primary Key

- It is identified by the locker_ID attribute because this attribute should be uniquely identified.



Entity 15: Return (return_id,student_id, book_id, date)

The return entity has four attributes, the ID is the primary key which should be uniquely identified; the student_id is the foreign key which shows the Id of student who has returned the book; book_id is the foreign key which shows the Id of the book which has been returned ; date shows the time stamp at which the book is returned.

Attributes

- return_ID: the return_ID of each book which has been returned, type: int
- student_id: the ID of the students who has returned the books, type: varchar(15)
- book_id: the ID of the book which has been returned, type: varchar(15)
- staff_id: the ID of the staff who has returned the books, type: varchar(15)
- mID: the ID of the guests who has returned the books, type: varchar(15)
- date: the timestamp at which the book has been returned, type: date

Relationships

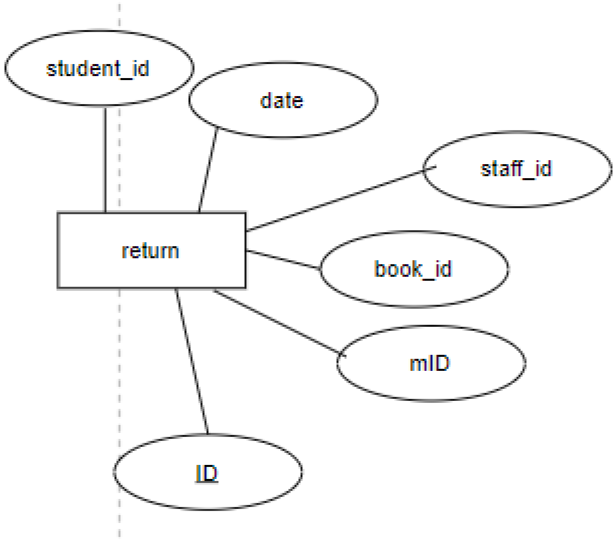
- *Relationship: person can return the books.

Primary Key

- It is identified by the return_ID attribute because this attribute should be uniquely identified.

Foreign Key

- book_id is the foreign key which is referred from the Books entity.
- student_id is the foreign key which is referred from the Student entity.
- staff_id is the foreign key which is referred from the Staff entity.
- mID is the foreign key which is referred from the membership entity.



Entity 16: Settlements (settelment_id,book_id, mID, student_id, staff_id, amount, resources_used)

The settlements entity has seven attributes, the settelment_id is the primary key which should be uniquely identified; the student_id is the foreign key which shows the Id of student ; book_id is the foreign key which shows the Id of the book;the mID is the foreign key which shows the Id of membership;the staff_id is the foreign key which shows the Id of staff;amount shows the total amount that must be paid by a person; resource_used shows the type of resource they have used.

Attributes

- settelment_id : the ID of the settlement, type: int
- student_id: the ID of the students who have to settle the bill, type: varchar(15)
- book_id: the ID of the book which has been damaged, type: varchar(15)
- staff_id: the ID of the staff who have to settle the bill type: varchar(15)
- mID: the ID of the guests who have to settle the bill, type: varchar(15)
- amount: the total amount the person is required to settle, type: Integer
- resource_used: type of the resource the person has utilized, type: varchar(15)

Relationships

- *Relationship: person has to settle the amount if resources are damaged or utilized .

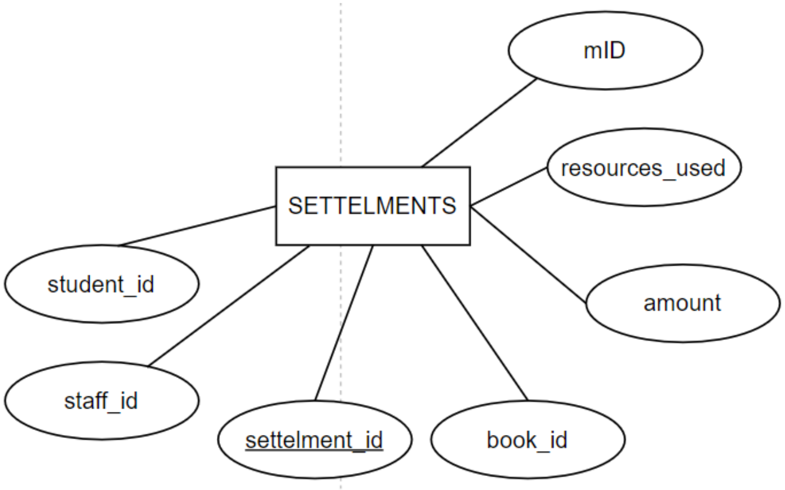
Primary Key

- It is identified by the settelment_id attribute because this attribute should be uniquely identified.

Foreign Key

- book_id is the foreign key which is referred from the Books entity.
- student_id is the foreign key which is referred from the Student entity.

- staff_id is the foreign key which is referred from the Staff entity.
- mID is the foreign key which is referred from the membership entity.



Entity 17: QuickShop:

Order (order_id,date, total_amount)

Product (product_id, product_name, price)

Payment (payment_id, date, total_amount)

The QuickShop entity has 3 entities present in it namely: order, payment, product; The order entity has three attributes: order_id, total_amount, date; order_id is a primary key which has to be uniquely identified; total_amount shows the total cost of that order; date shows the timestamp at which the order took place. The payment entity has three attributes: payment_id, total_amount, date; payment_id is a primary key which has to be uniquely identified; total_amount shows the total price; date shows the timestamp at which the payment took place. The product entity has three attributes: product_id, product_name, price; product_id is a primary key which has to be uniquely identified; product_name shows the name of the product ;price shows the cost of that product.

Attributes

- order_id : the ID of the order, type: Varchar(10)

Entities
ORDER:
Attributes

- order_ID: the order_ID of each order, type: varchar(10)
- total_amount: the total cost of that order, type: Integer

- date: the timestamp at which the order took place, type: date

Primary Key

- It is identified by the order_ID attribute because this attribute should be uniquely identified.

PAY:
Attributes

- payment_ID: the payment_ID of each order, type: varchar(10)
- total_amount: the total cost of that payment, type: Integer
- date: the timestamp at which the payment took place, type: date

Primary Key

- It is identified by the payment_ID attribute because this attribute should be uniquely identified.

Entity 18: PRODUCT:
Attributes

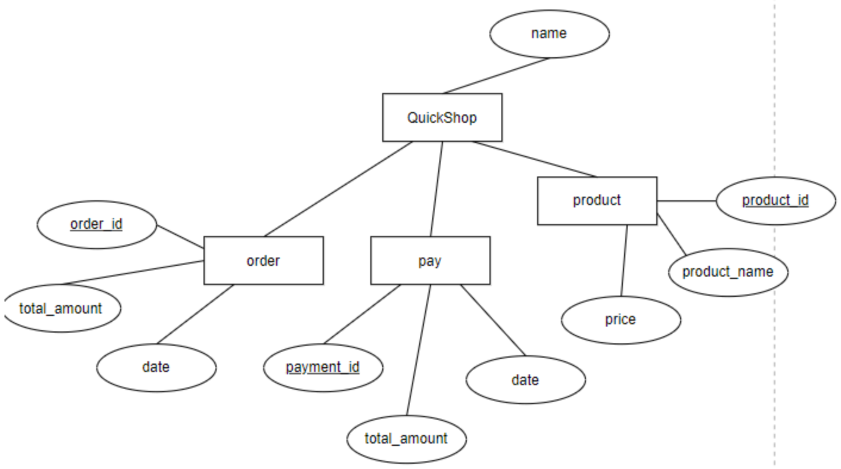
- product_ID: the product_ID of each order, type: varchar(10)
- price: the cost of that product, type: Integer
- product_name: the name of each product, type: char(10)

Primary Key

- It is identified by the product_ID attribute because this attribute should be uniquely identified.

Relationships

- *Relationship: person purchases from QuickShop.

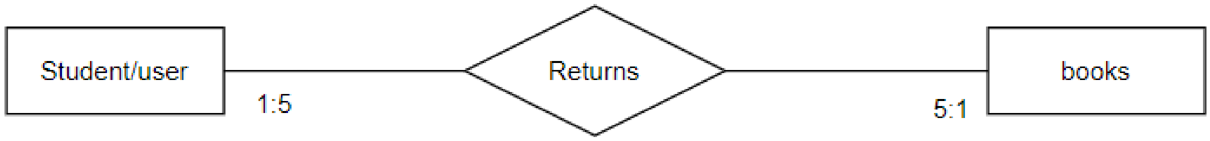


Functional Requirements

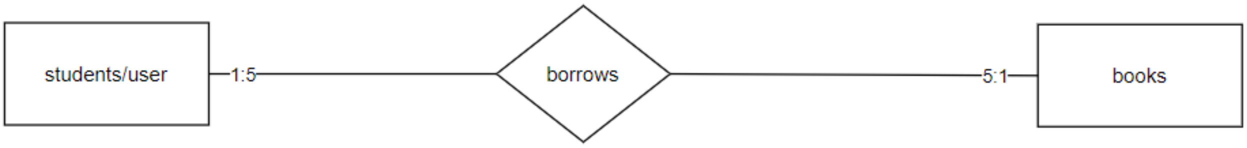
- Membership id given to people from outside the organization should be auto-generated.
- Access to modify book details can only be given to library Admin staff.
- Students/ users can only borrow up to 5 books at a time. If they try to borrow a new book after borrowing 5 books a trigger is issued saying the maximum limit reached.
- views are created for students/ users to check the availability of books/resources.
- students/professors/guests can check the book availability by the title/author of books
- Library staff can query the information of books that have been borrowed and the information of the person that borrowed the books
- The information of books should be updated, for example, if the books have been returned, the availability of the books for other students should be updated
- The students' borrowing and returning records should be updated on time
- Students can query the availability of study rooms
- After a reservation is made by a student, the availability of that study room that was reserved by the student should be set to 'False'.
- The availability of study rooms should be updated in time, if the students leave, the availability of that study room should be set to 'True'
- Professors can query the availability of lecture halls
- After a reservation is made by a professor, the availability of the lecture hall that is reserved by the professor should be set to 'False'.
- The availability of lecture halls should be updated in time, if the professor leaves, the availability of that lecture hall should be set to 'True'

Relationships: [Heading 2]

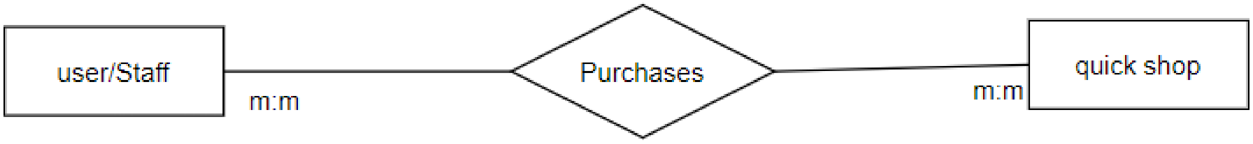
'Returns' is the relation between students and books where it shows us the details about the students returning the books they have borrowed from the library.



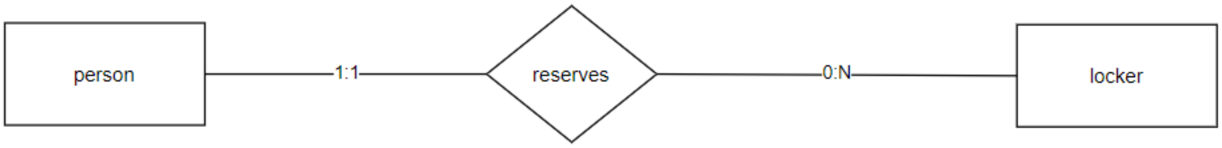
'Borrows' is the relation between students and books where it shows us the details about the books that have been borrowed by the students from the library.



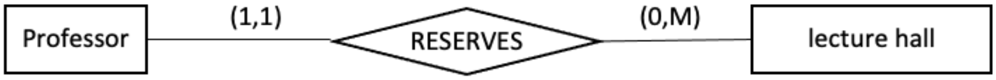
’Purchases‘ is the relation between the user/staff and quick shop where it shows us the details about the user/staff purchasing the orders from the quick shop.



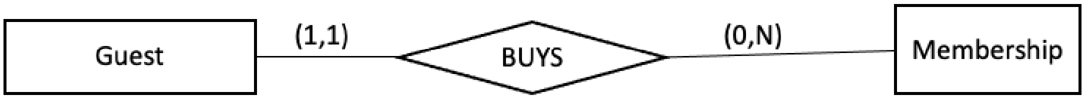
’Reserves‘ is the relation between the entity person and the entity locker where it shows us the details about the lockers reserved by a person in the library.



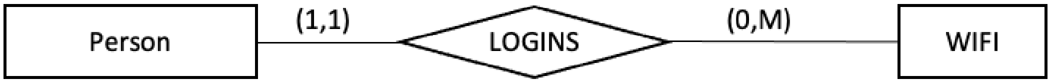
The ’reserves‘ shows the relation between the entity professor and the entity lecture hall, it shows the details of the reservations of the lecture hall made by professors.



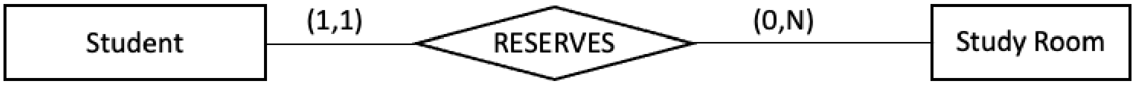
The ’buys‘ shows the relation between the entity guest and the entity membership, it shows the details of the memberships that are bought by guests.



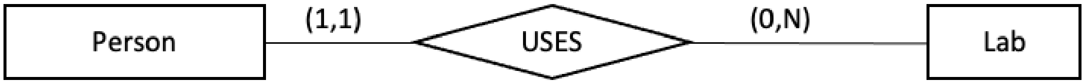
The ’logins‘ shows the relation between the entity person and the entity wifi.



The ‘reserves’ shows the relation between the entity student and the entity study room, it shows the details of the reservations of the study rooms made by students.



The ’uses‘ shows the relation between the entity person and the entity lab.



Mapping

Master table:

Person_id	Person_Type	Emp_id	staff_id	stu_id	name	Position

Employee

emp_id	name	department	Position

Staff

staff_id	name	date_of_birth	department	Position

Student

stu_id	name	date_of_birth	Dept	major

Material

Mat_id	mat_type	book_id	JAM_id	name

Books

Book_id	b_name	author	Category	published_year	edition	Publication

Journals/Articles/magazines(J/A/M)

J/A/M_id	jam_name	Author	Category	published_year	Edition	publication

Studyroom

roomId	occupied	capacity

Studyroom Reservation

stuId	roomId	Date	startTime	endTime

Faculty

facultyId	name	Department

Lecturehall Reservation

facultyId	hallId	Date	startTime	endTime

Lecturehall

hallId	occupied	capacity

Guest

memberId	name	phone

Membership

id	memberId	startDate	EndDate

Lab

labId	type	capacity

Printer

printerId	p_type	location

Locker

lockerId	availability

material_Borrow

ID	Person_id	Mat_id	date

material_Return

ID	Person_id	Mat_id	date

Settlements

settlement_id	Mat_id	Person_id	amount

order

order_id	order_name	order_amount	date

payment

payment_id	payment_amount	payment_method	date

product

product_id	product_name	price

QuickShop_list

order_id	product_id	product_quantity	amount

locker_Reserves

locker_id	Person_id	start_date	end_date

Data Dictionary (Phase 5):
In this section, show your data dictionary for each table (or relation you mapped). Demons trate the constraints (null, unique, or other check constraints such as [check age > 0]) on the data, show the data types and the descriptions of each attribute. Add necessary information about candidate, primary, and foreign keys.
`The date and time format used->Date: YYYY-MM-DD   Time: hh:mm:ss`

Person master table:

Table	Column	Data Type	Reference	Not null
Person	person_id	varchar(10)	Primary key	Y
Person	name	char(20)		Y

Person	DOB	Date		Y
Person	department	varchar(10)		Y
Person	Position	varchar(20)		Y
Person	Major	varchar(10)		N

Employee

Table	Column	Data Type	Reference	Not null
employee	emp_id	varchar(10)	person_id(person)	Y
employee	name	char(20)		Y
employee	department	varchar(10)		Y
employee	Position	varchar(20)		Y

Staff

Table	Column	Data Type	Reference	Not null
staff	staff_id	varchar(10)	person_id(person)	Y
staff	name	char(20)		Y
staff	date_of_birth	Date		Y
staff	department	varchar(10)		Y
staff	Position	varchar(20)		Y

Student

Table	Column	Data Type	Reference	Not null
student	stu_id	varchar(10)	person_id(person)	Y
student	name	char(20)		Y
student	date_of_birth	Date		Y
student	Dept	varchar(10)		Y
student	major	varchar(10)		Y

Material

Table	Column	Data Type	Reference	Not null
Material	mat_id	varchar(10)	Primary Key	Y
Material	name	varchar(30)		Y
Material	author	char(20)		Y
Material	Category	char(20)		Y
Material	published_year	Date		Y
Material	edition	varchar(5)		N
Material	Publication	varchar(30)		Y

Books

Table	Column	Data Type	Reference	Not null
Books	Book_id	varchar(10)	mat_id (Material)	Y
Books	b_name	varchar(30)		Y
Books	author	char(20)		Y
Books	Category	char(20)		Y
Books	published_year	Date		Y
Books	edition	varchar(5)		N
Books	Publication	varchar(30)		Y

JAM

Table	Column	Data Type	Reference	Not null
J/A/M	Book_id	varchar(10)	mat_id (Material)	Y
J/A/M	b_name	varchar(30)		Y
J/A/M	author	char(20)		Y
J/A/M	Category	char(20)		Y
J/A/M	published_year	Date		Y
J/A/M	edition	varchar(5)		N

J/A/M	Publication	varchar(30)		Y

Table	Column	Data Type	Reference	Not null
Studyroom	roomId	int		Y
Studyroom	occupied	bit		Y
Studyroom	capacity	int		Y

Table	Column	Data Type	Reference	Not null
Studyroom_re	stuId	int	student	Y
Studyroom_re	roomId	int	studyroom	Y
Studyroom_re	Date	Date		Y
Studyroom_re	startTime	Time		Y
Studyroom_re	endTime	Time		Y

Table	Column	Data Type	Reference	Not null
Faculty	id	int		Y
Faculty	name	varchar(30)		Y
Faculty	department	varchar(30)		Y

Table	Column	Data Type	Reference	Not null
Lecturehall	hallId	int		Y
Lecturehall	occupied	bit		Y
Lecturehall	capacity	int		Y

Table	Column	Data Type	Reference	Not null
Lecturehall_re	facultyId	int	faculty	Y
Lecturehall_re	hallId	int	lecturehall	Y
Lecturehall_re	Date	Date		Y
Lecturehall_re	startTime	Time		Y
Lecturehall_re	endTime	Time		Y

Table	Column	Data Type	Reference	Not null
Lab	labId	int		Y
Lab	l_type	varchar(15)		Y
Lab	capacity	int		Y

Table	Column	Data Type	Reference	Not null
Printer	printerId	int		Y
Printer	location	varchar(30)		Y
Printer	p_type	varchar(15)		Y

Table	Column	Data Type	Reference	Not null
Guest	memberId	int		Y
Guest	name	varchar(30)		Y
Guest	phone	varchar(16)		Y

Table	Column	Data Type	Reference	Not null

membership	Id	int		Y
membership	member_id	int	guest	Y
membership	startDate	Date		Y
membership	endDate	Date		Y

Table	Column	Data Type	Reference	Not null
locker	locketr_id	int		Y
locker	availability	boolean		Y

Table	Column	Data Type	Reference	Not null
material_Borrow	id	int		Y
material_Borrow	Person_id	varchar[10]	Person	Y
material_Borrow	Mat_id	varchar[20]	Material	Y
material_Borrow	date	Date		Y

Table	Column	Data Type	Reference	Not null
material_Return	id	int		Y
material_Return	Person_id	varchar[10]	Person	Y
material_Return	Material_id	varchar[10]	Material	Y

material_Return	date	Date		Y

Table	Column	Data Type	Reference	Not null
settlement	settlement_id	int		Y
settlement	Person_id	varchar[10]	Person	Y
settlement	Material_id	varchar[10]	Material	Y
settlement	amount	int		Y

Table	Column	Data Type	Reference	Not null
order	order_id	varchar[10]		Y
order	order_name	varchar[20]		Y
order	order_amount	int		Y
order	date	Date		Y

Table	Column	Data Type	Reference	Not null
payment	payment_id	varchar[10]		Y
payment	payment_amount	int		Y
payment	payment_method	varchar[10]		Y
payment	date	Date		Y

Table	Column	Data Type	Reference	Not null
product	product_id	varchar[10]		Y
product	product_name	char[10]		Y
product	total_amount	int		Y

Table	Column	Data Type	Reference	Not null
QuickShop_list	order_id	varchar[10]	order	Y
QuickShop_list	product_id	varchar[10]	product	Y
QuickShop_list	product_quantity	int		Y
QuickShop_list	amount	int		Y

Table	Column	Data Type	Reference	Not null
locker_reserves	locker_id	int	locker	Y
locker_reserves	Person_id	varchar[10]	Person	Y
locker_reserves	start_date	Date		Y
locker_reserves	start_date	Date		Y

Unique keys

Type	Table	Key	Columns
Primary	Person	pk_Person	person_id
Primary	Matreial	pk_material	mat_id
Primary	studyroom	pk_studyroom	roomId
Primary	lecturehall	pk_lecuturehall	hallId
Primary	lab	pk_lab	labId
Primary	printer	pk_printer	printerId
Primary	Guest	pk_guest	memberId
Primary	membership	pk_membership	Id
Primary	locker	pk_locker	locker_id
Primary	borrow	pk_borrow	borrow_id
Primary	return	pk_return	return_id

Primary	settlement	pk_settlement	settlement_id
Primary	order	pk_printer	printer_id
Primary	payment	pk_payment	member_id
Primary	product	pk_product	Member_id

Foreign keys

Key	Table	Columns	ref-table	ref-column
fk_employee	employee	emp_id	Person	person_id
fk_staff	staff	staff_id	Person	person_id
fk_student	student	stu_id	Person	person_id
fk_books	books	book_id	Material	mat_id
fk_JAM	JAM	jam_id	Material	mat_id
fk_studyroom_re	studyroom_re	roomId	studyroom	roomId
fk_studyroom_re	studyroom_re	stuId	student	stuId
fk_lecturehall_re	lecturehall_re	hallId	lecturehall	hallId
fk_lecturehall_re	lecturehall_re	facultyId	faculty	facultyId
fk_membership	membership	memberId	guest	memberId
fk_locker_reserves	locker_reserves	Person_id	Person	Person_id
fk_locker_reserves	locker_reserves	locker_id	locker	locker_id
fk_borrow	borrow	Person_id	Person	Person_id
fk_borrow	borrow	Material_id	Material	Mat_id
fk_return	return	Person_id	Person	Person_id
fk_return	return	Material_id	Material	Mat_id
fk_settlement	settlement	Person_id	Person	Person_id
fk_settlement	settlement	Material_id	Material	Mat_id
fk_QuickShop_li	QuickShop_li	order_id	order	order_id

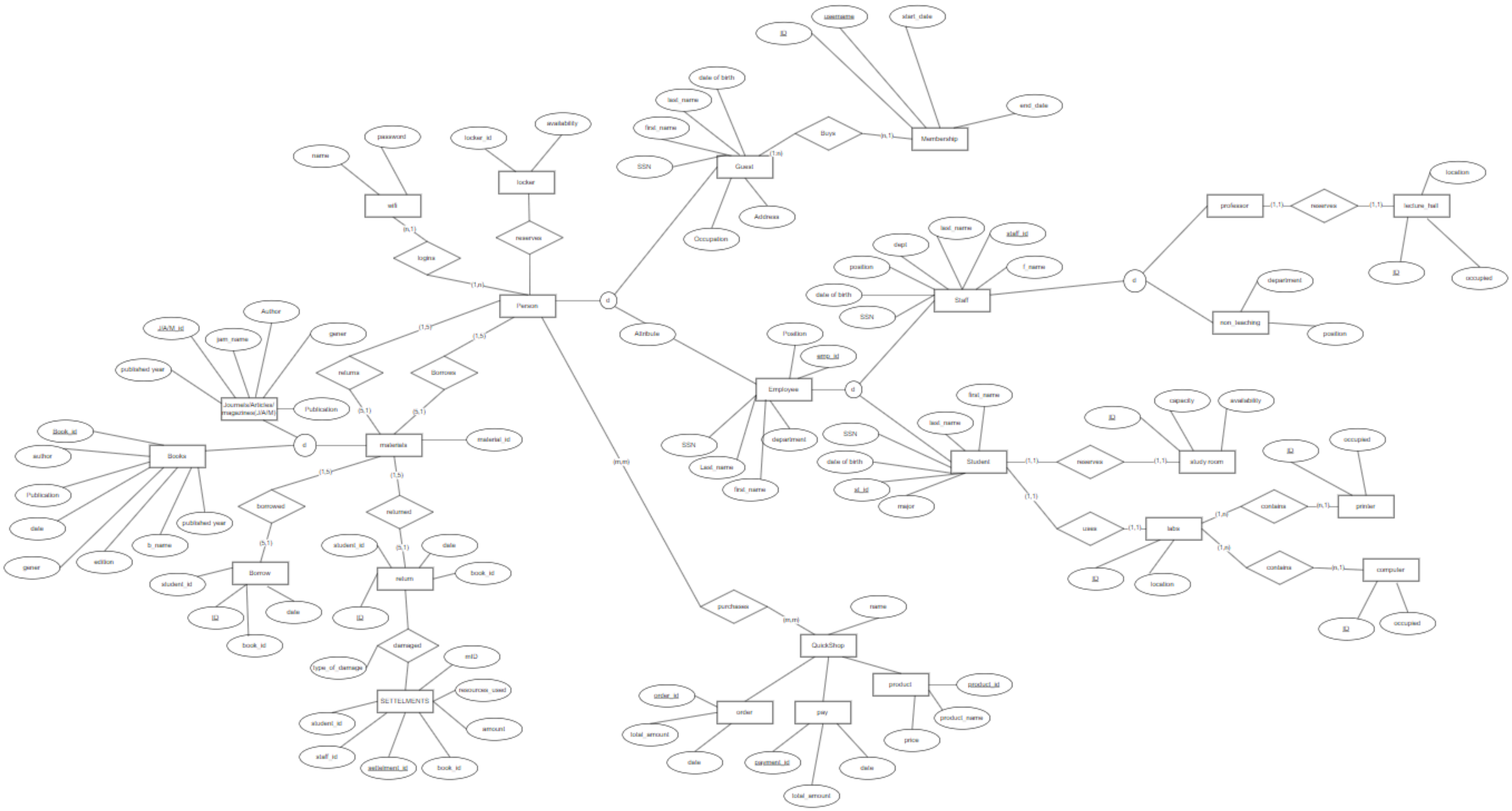
Key	Table	Columns	ref-table	ref-column
fk_employee	employee	emp_id	Person	person_id
fk_staff	staff	staff_id	Person	person_id
fk_student	student	stu_id	Person	person_id
fk_books	books	book_id	Material	mat_id
fk_JAM	JAM	jam_id	Material	mat_id
fk_studyroom_re	studyroom_re	roomId	studyroom	roomId
fk_studyroom_re	studyroom_re	stuId	student	stuId
fk_lecturehall_re	lecturehall_re	hallId	lecturehall	hallId
fk_lecturehall_re	lecturehall_re	facultyId	faculty	facultyId
st	st			
fk_QuickShop_list	QuickShop_list	product_id	product	product_id

Teamwork

We separated the work into three parts, and each of us designed several entities (including the attributes and the relationships, and the partial ER model), and wrote the requirements of the entities. Then we combined our work together, the TA might have noticed that there are some different formats in the relationships part. It is because we draw the figure separately.

s.No	Work	Done By	Description
1	Introduction, design Requirements, functional requirements.	Preethi, Yagnashree, Peiyu Li	All three of us discussed the details and came to a conclusion.
2	entities	Yagnashree works on entities 1-5 Peiyu Li works on entities 6-12 Preethi works on entities 13-18	Each one of us worked on several entities
3	ER diagram	Yagnashree	Entities 1-5
4	ER diagram	Peiyu Li	Entities 6-12
5	ER diagram	Preethi	Entities 13-18 combined all the entities.

Appendix A - ER diagram



The picture we show here is not clear enough to identify, so please find the image of ER diagram here and zoom in to get a clear image:

ER_Diagram.jpg