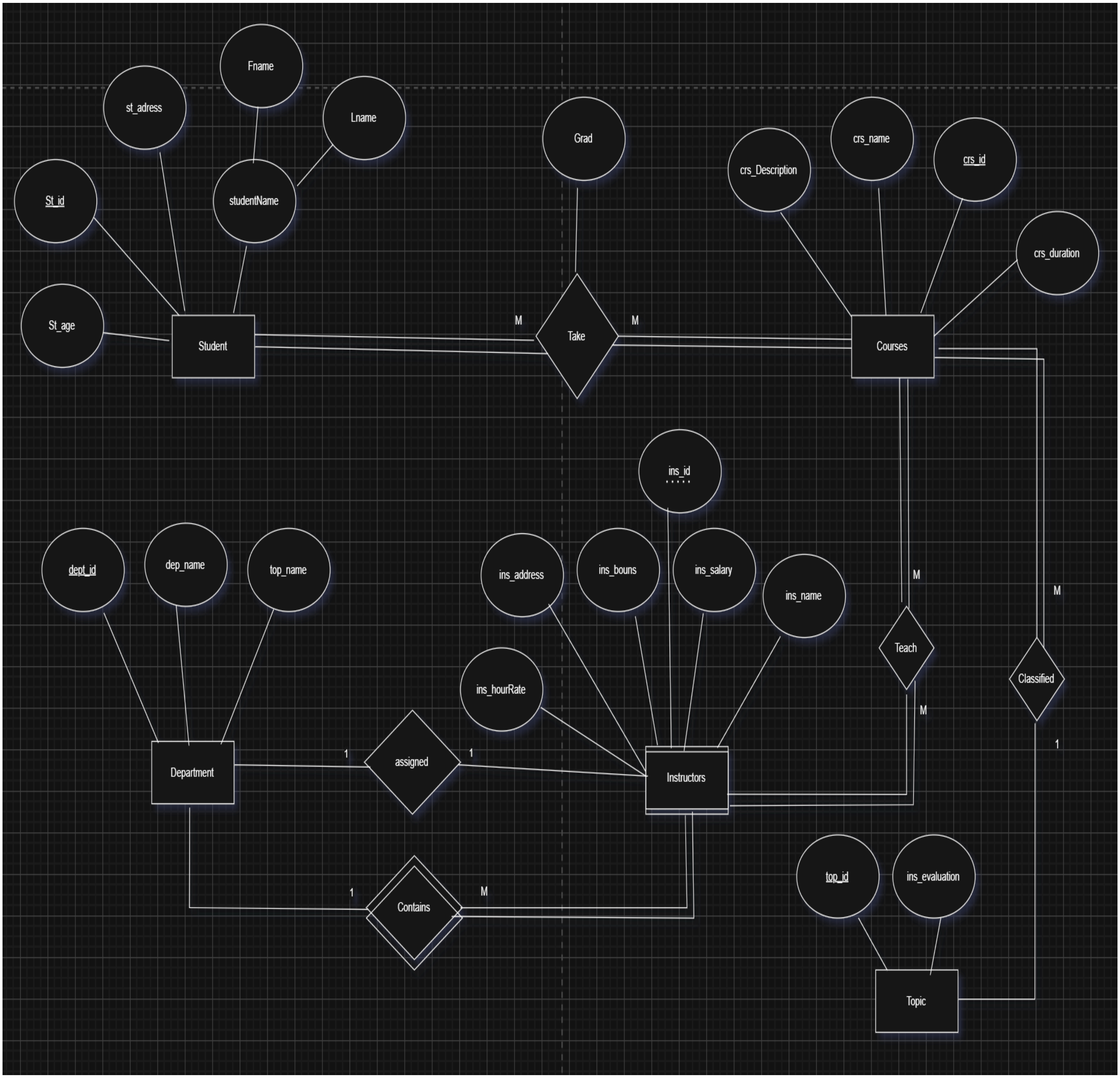
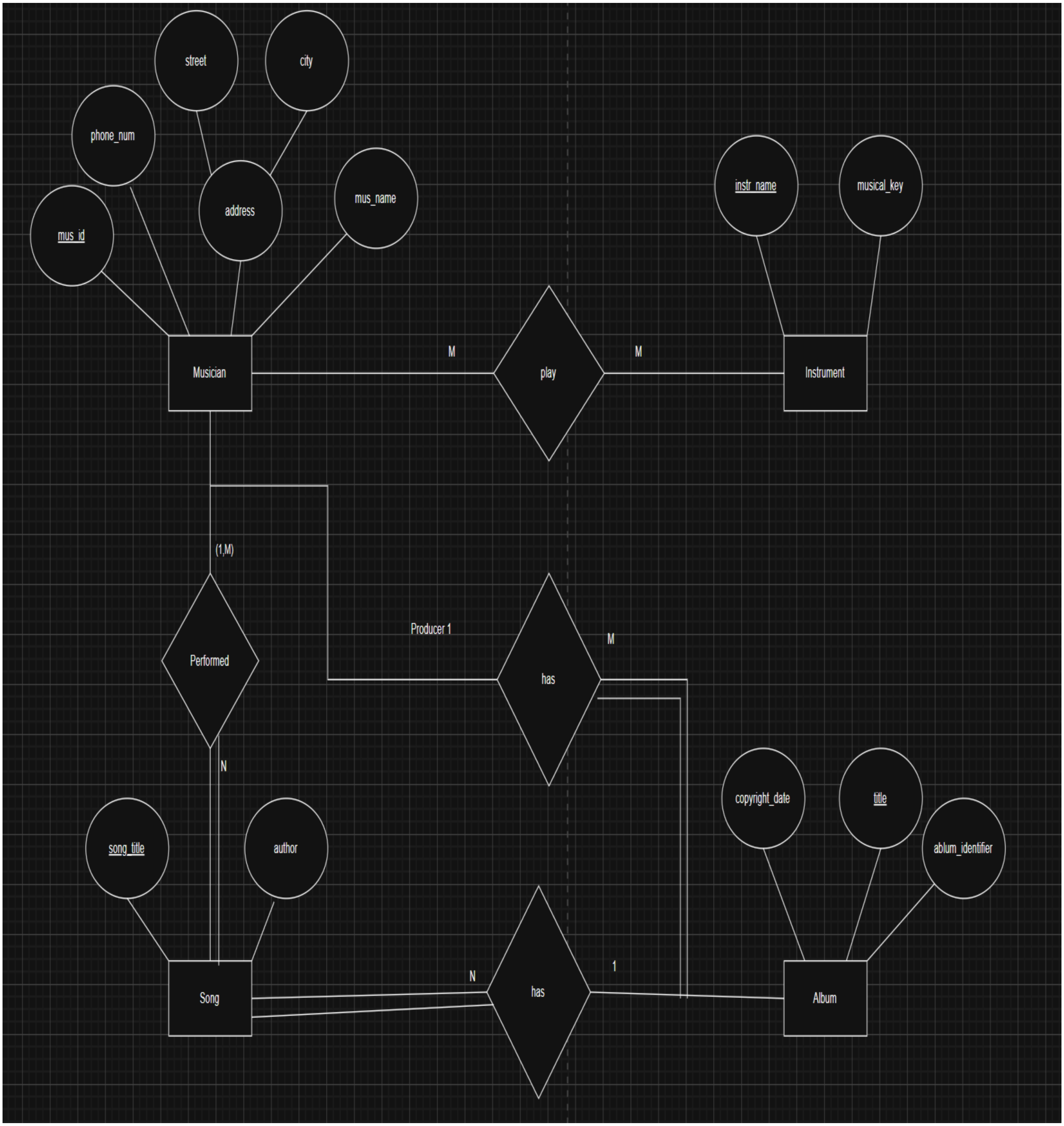
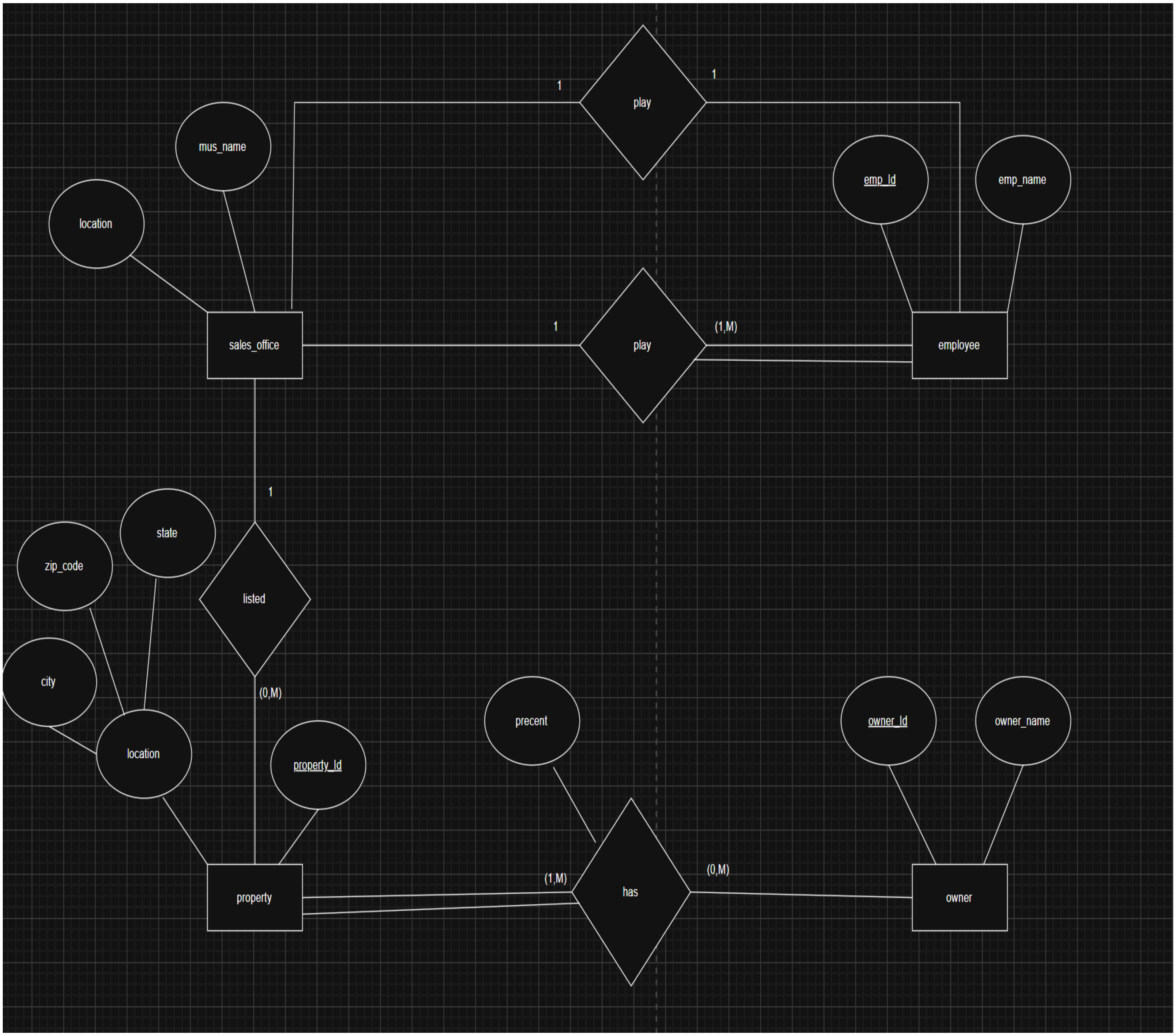
Problem 1:


Fname
st_adress
Lname
Grad
crs_name
crs_Description
crs_id
St_id
studentName
crs_duration
St_age
Student
M
Take
M
Courses
ins_id
dept_id
dep_name
top_name
ins_address
ins_bouns
ins_salary
ins_name
M
M
Teach
Classified
ins_hourRate
M
1
1
1
assigned
Department
Instructors
top_id
ins_evaluation
1
Contains
M
Topic

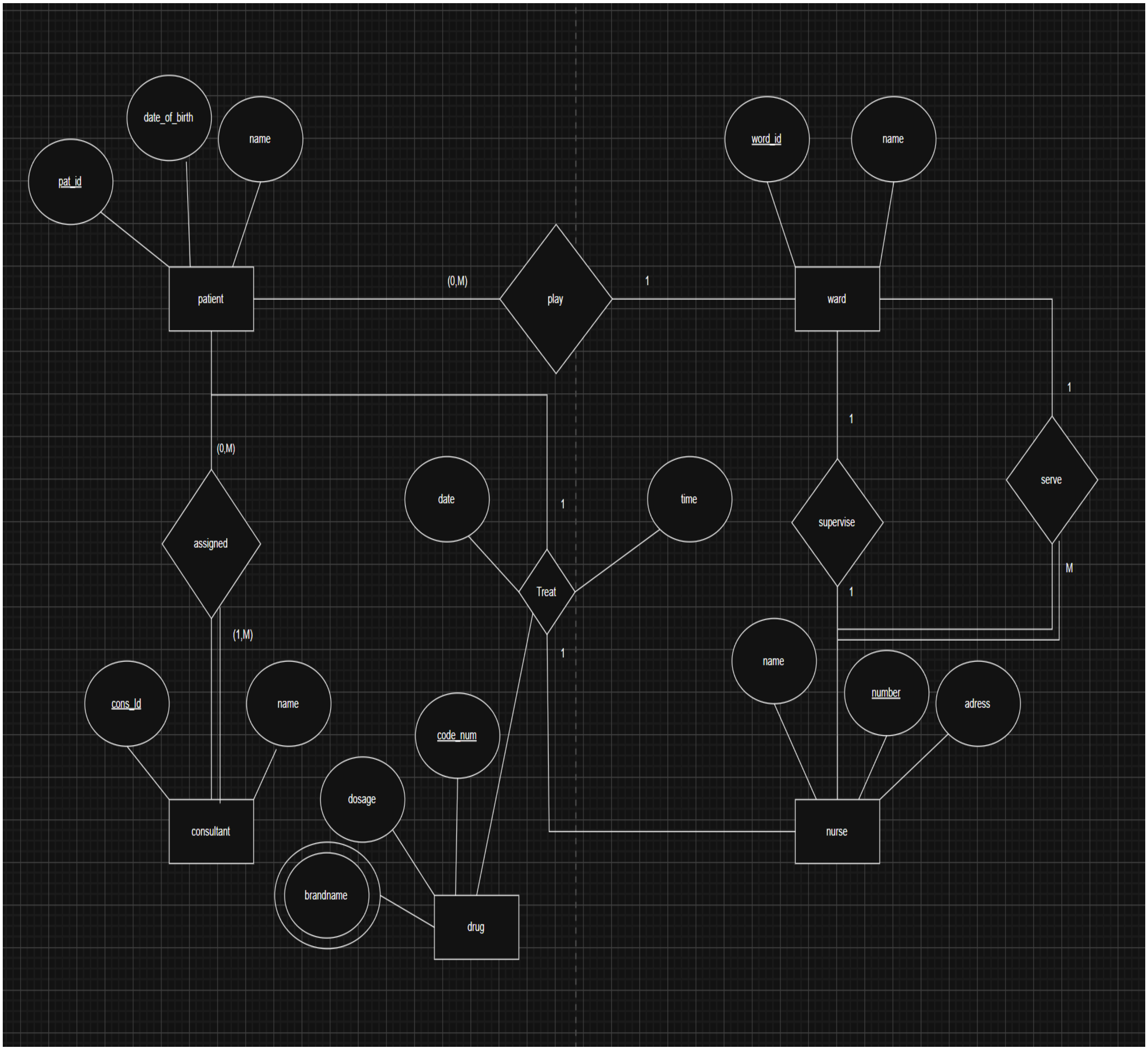
Problem 2:


street
city
phone_num
address
mus_name
mus_id
instr_name
musical_key
Musician
M
play
M
Instrument
(1,M)
Producer 1
M
has
Performed
N
copyright_date
title
ablum_identifier
song_title
author
N
has
1
Song
Album

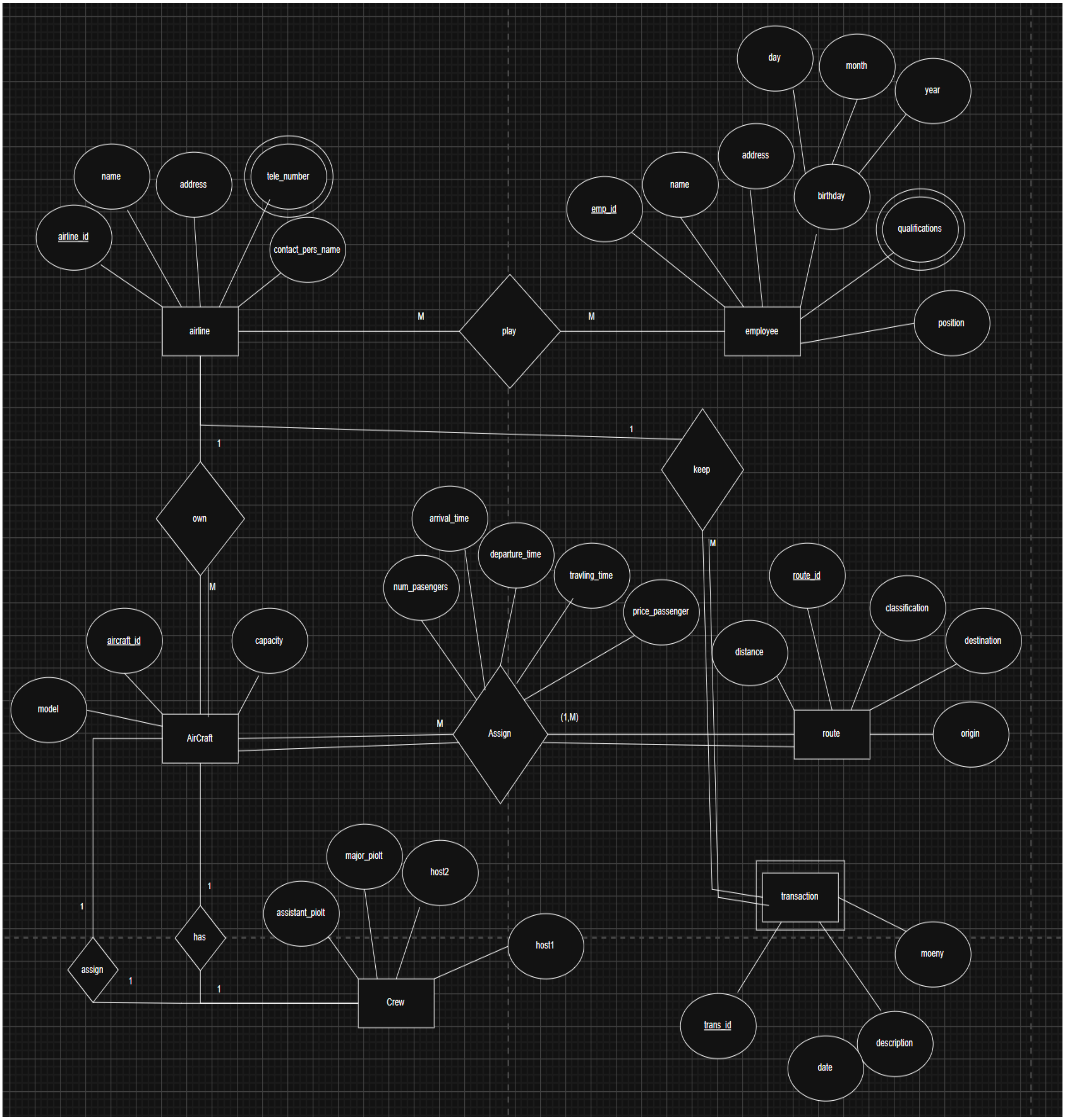
Problem 3 :


1
play
1
mus_name
emp_Id
emp_name
location
1
play
(1,M)
sales_office
employee
1
state
zip_code
listed
city
(0,M)
precent
owner_Id
owner_name
location
property_Id
(1,M)
has
(0,M)
property
owner

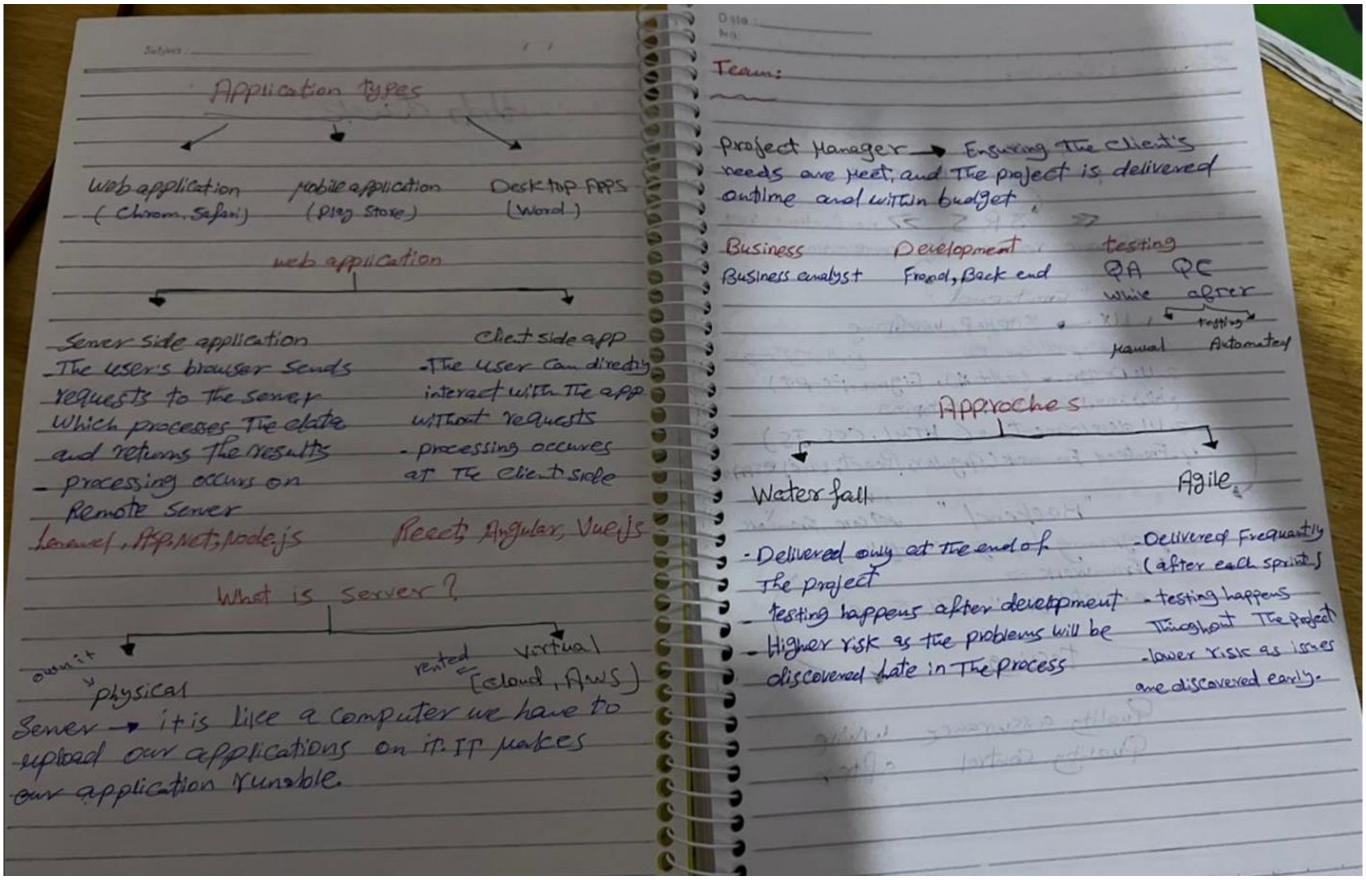
Problem 4 :


date_of_birth
name
pat_id
word_id
name
patient
(0,M)
play
1
ward
1
1
(0,M)
date
time
serve
supervise
assigned
1
Treat
M
1
(1,M)
1
name
cons_Id
name
number
adress
code_num
dosage
consultant
nurse
brandname
drug

Problem 5:


day
month
year
address
birthday
name
emp_id
qualifications
position
employee
M
play
M
name
address
tele_number
airline_id
contact_pers_name
airline
1
1
keep
M
own
M
arrival_time
departure_time
num_pasengers
travling_time
price_passenger
route_id
classification
destination
distance
origin
route
(1,M)
Assign
M
aircraft_id
capacity
model
AirCraft
major_piolt
host2
assistant_piolt
host1
1
1
has
1
assign
1
1
Crew
transaction
moeny
trans_id
description
date

Notes:



Application types

Web application
(Chrom, Safari)

Mobile application
(Play Store)

Desk top APPS
(Word)

web application

Server side application
- The user's browser sends requests to the server which processes the data and returns the results
- processing occurs on Remote Server

Laravel, Asp.Net, Node.js

Client side app
- The user can directly interact with the app without requests
- processing occures at the client side

React, Angular, Vue.js

What is server?

own it
physical

rented
virtual
[cloud, AWS)

Server → it is like a computer we have to upload our applications on it. IT makes our application runable.

Team:

Project Manager → Ensuring the client's needs are meet, and the project is delivered ontime and within budget.

Business
Business analyst

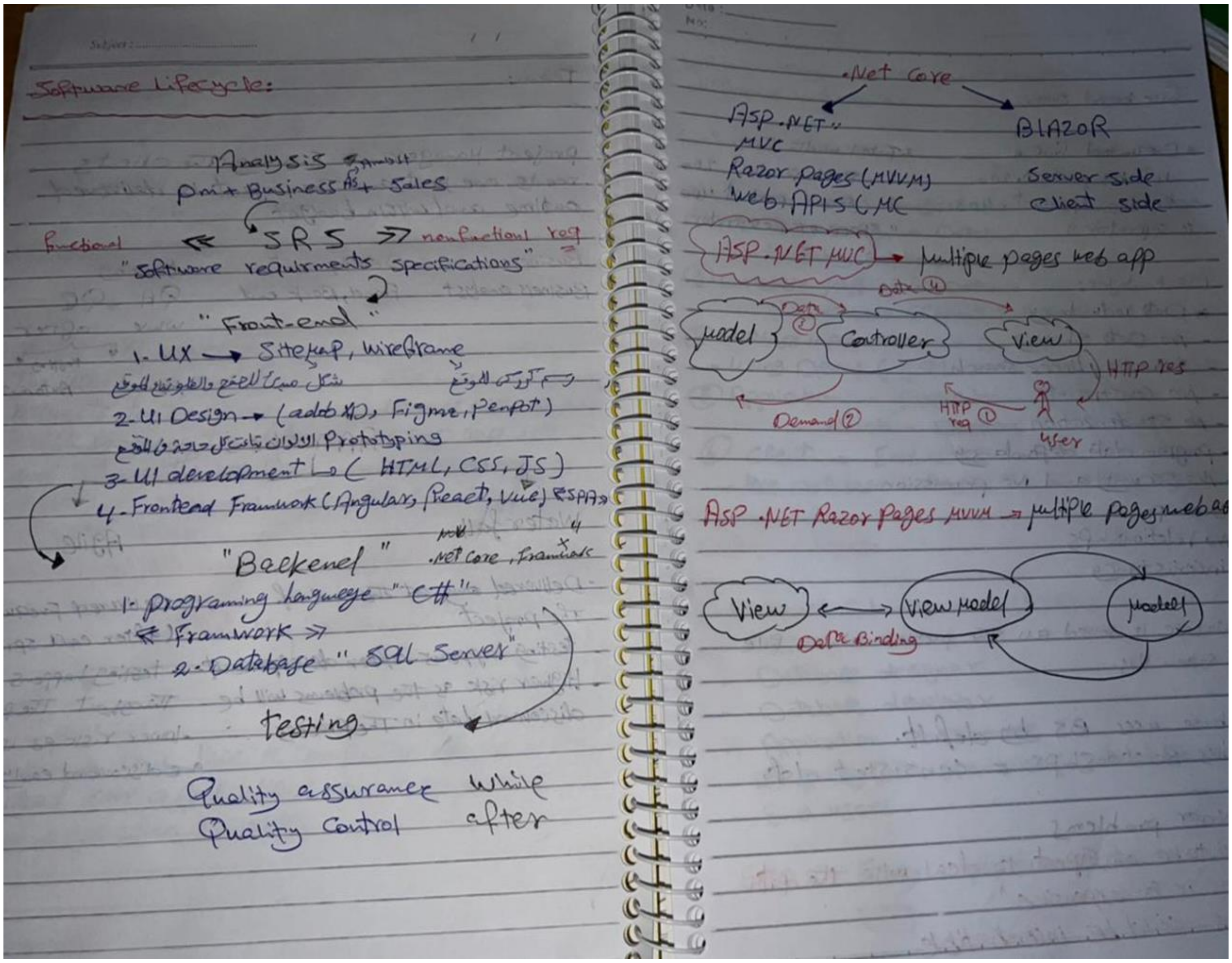
Development
Frond, Back end

testing
QA QC
while after
testing
Manual Automated

Approches

Water fall
- Delivered only at the end of the project
- testing happens after development
- Higher risk as the problems will be discovered late in the process

Agile
- Delivered Frequantly (after each sprint)
- testing happens Throghout the project
- lower risk as issues are discovered early.



Software Lifecycle:

Analysis Analyst

pm + Business As + Sales

Functional ⇇ SRS ⇉ nonfunctional req

"Software requirments specifications"

"Front-end"

1. UX → Sitemap, wireframe

شكل مبدئي للصفح والعلوم تباع الموقع رسم كروكي للموقع

2. UI Design → (adob XD, Figma, Penpot)

الالوان بتاعت كل حاجة في الموقع Prototyping

3. UI development → (HTML, CSS, JS)

4. Frontend Framwork (Angular, React, Vue) «SPA»

"Backend" .Net Core, Framwork

1. programing language "C#"

⇇ Framwork ⇉

2. Database "SQL Server"

Testing

Quality assurance while

Quality control after

.Net Core

ASP.NET

MVC

Razor pages (MVVM)

Web APIS (MC

BLAZOR

Server side

Client side

ASP.NET MVC → multiple pages web app

model

Controller

View

Data ②

Data ④

Demand ②

HTTP req ①

HTTP res

user

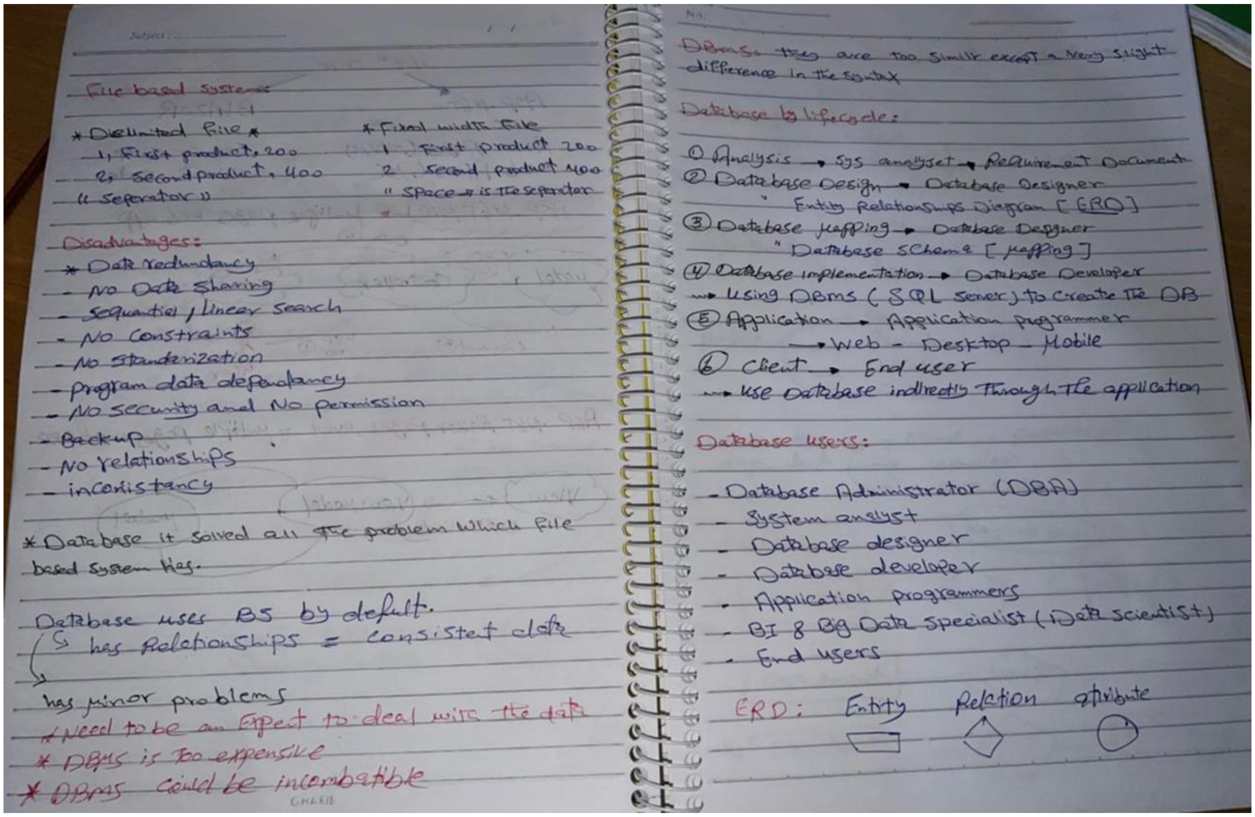
ASP.NET Razor Pages MVVM → multiple pages web ap

View

View model

model

Data Binding



File based systems:

* Dielimited File *
- 1, First product, 200
- 2, Second product, 400
- " seperator "

* Fixed width File
- 1 First product 200
- 2 Second product 400
- " space is the seperator

Disadvantages:

* Data redundancy
- No Data Sharing
- sequantiel / Linear search
- No Constraints
- No Standarization
- program data dependancy
- No security and No permission
- Backup
- No relationships
- inconistancy

* Database it solved all the problem which file based system has.

Database uses BS by defult.

↳ has Relationships = consistent data

↳ has minor problems

* Need to be an Expert to deal with the data
* DBMS is Too expensive
* DBMS could be incombatible

DBMSs they are too similar except a very slight difference in the syntax

Database by lifecycle:

① Analysis → sys analyset → Requirement Document
② Database Design → Database Designer
" Entity Relationships Diagram [ERD]
③ Database Mapping → Database Designer
" Database Scheme [Mapping]
④ Database implementation → Database Developer
→ Using DBMS (SQL Server) to Create The DB
⑤ Application → Application programmer
→ Web - Desktop - Mobile
⑥ Client → End user
→ Use Database indirectly Through the application

Database users:

- Database Administrator (DBA)
- System analyst
- Database designer
- Database developer
- Application programmers
- BI & Big Data specialist (Data scientist)
- End users

ERD: Entity ▱ Relation ◇ attribute ○