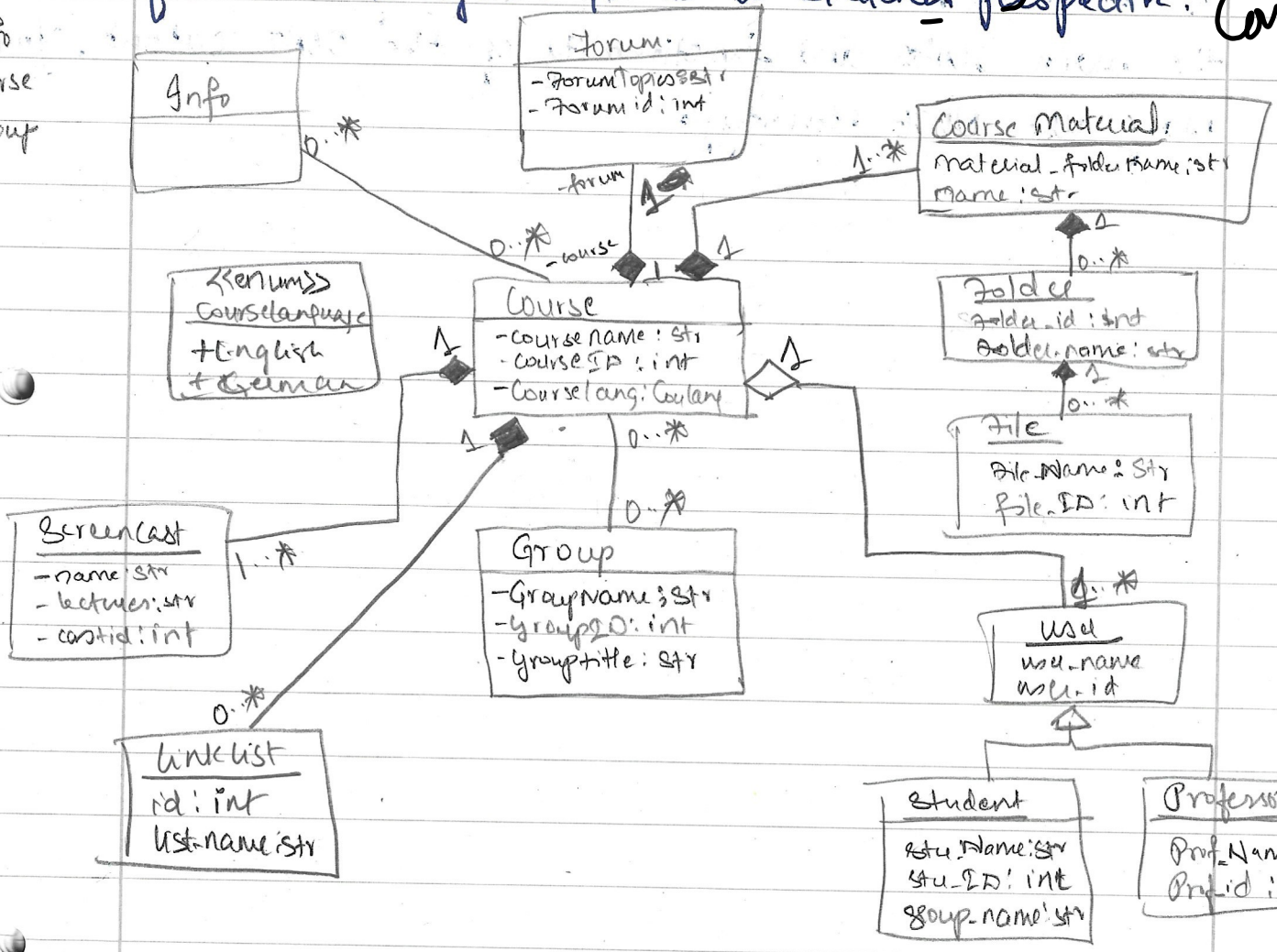
Assignment 01

1. Domain Model

Task: Create a UML class diagram as a domain model for the OLAT system from a students perspective:



→ ASS04
→ XML
→ Cypher

1. Use Cases, Navigation Model

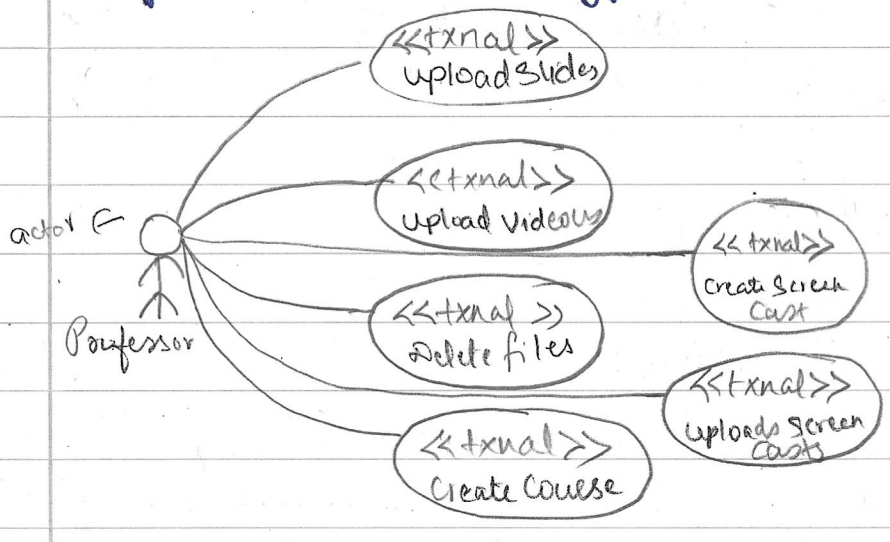
UML

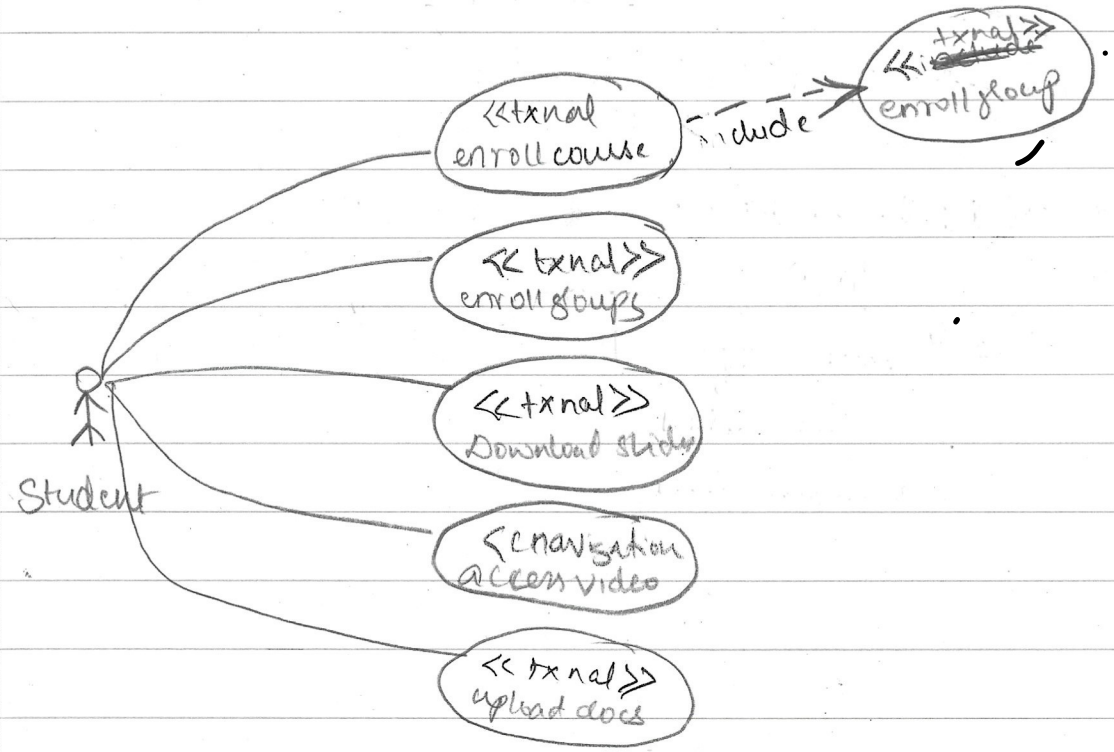
Task:

a) Usecase: Create a UML use case diagram that shows the main roles and functionality of the OLAT cutout. You may use UWE stereotypes

transactional
navigational
Personalised.

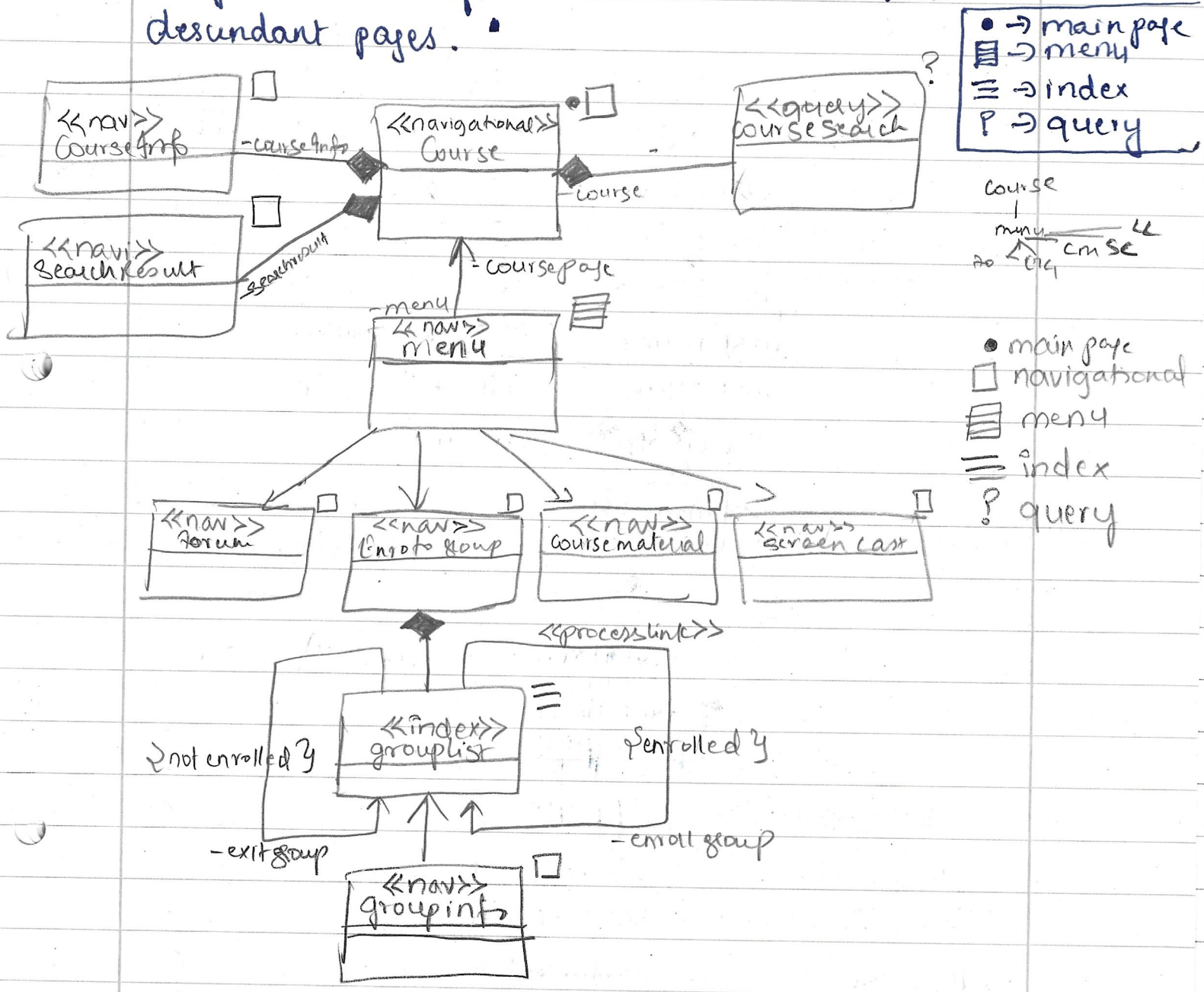
extends: optional functionali
include: mandatory functionali





(b) UWE Navigation Model:

Use the UWE extensions to UML class diagrams for a navigation model of the OLAT course view and its descendant pages.



→ XML DTD and document from Domain Model.

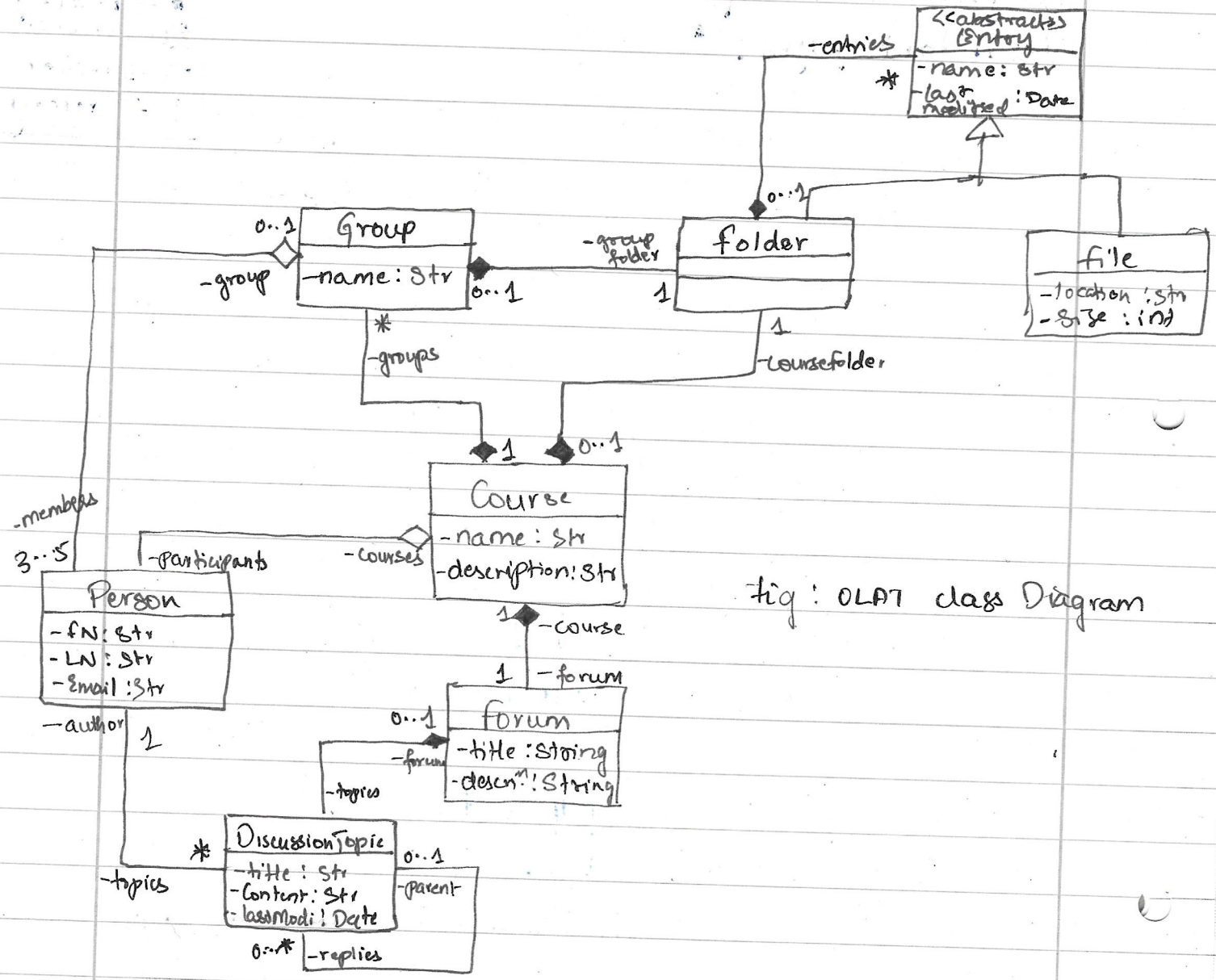


fig: OLAT class Diagram

(a) Create an XML DTD according to the mapping rules given ~~for~~ here

(b) Write an XML document that represents some of the facts from our current course. The document shall confirm to our DTD

<!Element course (folder, group*, forum)
<! Attlist course
id ID #REQUIRED
name CDATA #required
description CDATA #Required
Participant IDREF #Implied
<!Element folder (folder*, file)
<! Attlist folder
id ID #REQUIRED
name CDATA #required
last modified CDATA #required.
<!Element file EMPTY >
<! ATTLIST file
id ID #Required
location CDATA #required
size CDATA #required
name CDATA #required
last modified CDATA #required
folder IDREF #implied
↳ max 1 ↳ min 0
<! Element group (folder ?)
<! ATTlist group
#required : min 1
id ID #required
name CDATA #required
member IDREFS #implied.
<! Element forum (Discussion topics*)
↳ max *
<! Attlist forum
id ID #required
title CDATA #required
description CDATA #required

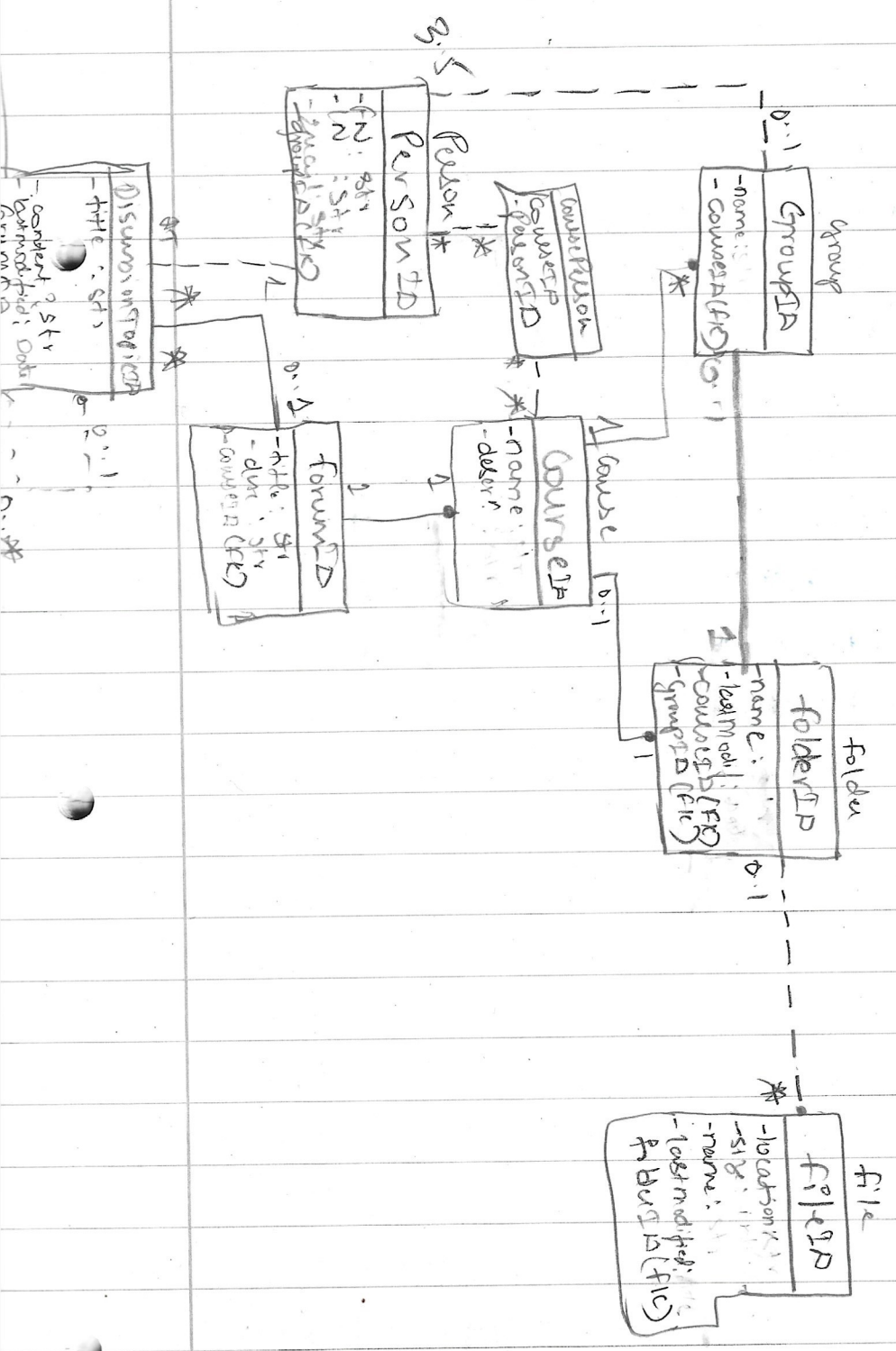
```
<! Element discussionTopic Empty >
<! Attlist discussionTopic
        id ID #required
        title CDATA #required
        Content CDATA #required
        last modif? date #required
<!Element Person Empty >
<! Attlist Person
        id ID #required
        first name CDATA #required
        last name CDATA #required
        email CDATA #required
```

(b) XML document

```
<?xml version="1.0"?>
<! DOCTYPE OLAT SYSTEM "C:\Users\Dell\Desktop\final DTD.dtd">
<OLAT>
<course participants="p1 p2 p3 p4 p5 p6 p7 p8 p9 p10"
        description="EWADIS" name="EWADIS" id="C1")
<group name="echo" id="g1" members="p1 p2 p3 p4">
<folder name="EWADIS-ECHO" id="f1" lastModified="Dec"
<file name="assig03" id="file1" lastmodified="Jan" size="20KB
    location="OLAT"/>
</folder>
</group>
<group name="alpha" id="g2" members="p5 p6 p7">
<folder name="EWADIS_ALPHA" id="f2" lastmodified="Dec">
<file name="assig02" id="file02" lastModified="Dec"
  size="20KB" location="OLAT"/>
</folder>
</group>
```

Assignment 04

1. Mapping OO schema to relationship DB. (O/R mapping)
   a) Create a relational schema.
   b) Explain decisions
   c) Test data
   d) Write SQL queries



Assignment 08

1. Mapping to Graph Database

(a) What are the major properties and the main differences between relational database and graph DB? How do data queries work in both paradigms? Please state some applications that benefit from one or the other technology. Provide a rationale why you think that the selected DB type fits better.

Ans Properties of Relational Database Vs graph DB

Relational DB	Graph DB
• Scalable	
• Includes Query & manipulation languages like SQL	NoSQL means non-relational
• tables (mathematical relations)	Other flavors of NoSQL DB are
• relation schema (attributes)	• Obj. oriented DB
• Description of structured by relational schema: Tables, attributes (columns), types, keys, foreign keys, index	• Key value stores • Doc. stores.
• Data is kept in relations:	Straight-forward representation of objects and their relations.
• Computation and change of data via SQL SELECT, INSERT, UPDATE, DELETE	• Based on graphs.
• Navigation via sets: Projection, selection, joins	• Mathematical foundation in GT.
• Results of computations are new relations	• Various graph types
	• Efficient graph algorithms.
	• Graph features are
	• directed Vs undirected edges
	• typed vertices & or edges
	• type system capabilities
	• labeled vertices & or edges.
	• Ordered vertex/edge/incidence sets
	• binary edges Vs. hyper edges
	• heirarchical graphs. (Ex: nodes can contain graphs)

Difference b/w Relational & Graph DB

• An RDBMS uses the relational model which uses tables	• More flexible than relational model.
• Any relationship b/w tables is defined when the tables are created.	• Graph DB uses vertices and edges
• Has a rigid, predefined structure.	• Does not have a fixed schema.
• Uses primary keys & foreign keys to maintain referential integrity.	• Schema-less
• Not very flexible as compared to Graph DB	• Graph DB are best suited to connected data, such as social media, product recommendations, organisational charts etc.
	• Does a better job when faced with large sets of associative data.

- How do data query work in Relational DB?
  - A typical SELECT query syntax may be written as follows: The square bracket [], represents optional parameters and the lowercase notation depicts user-defined variables.

```
SELECT [DISTINCT] columns        → removes duplicate records
FROM tables                      → forms a projection on the references
[WHERE expression]               → applies expression on ref. table
[GROUP BY columns]               → applies filter on the group.
[HAVING Expression]              → sorts result
[ORDER BY columns]
```

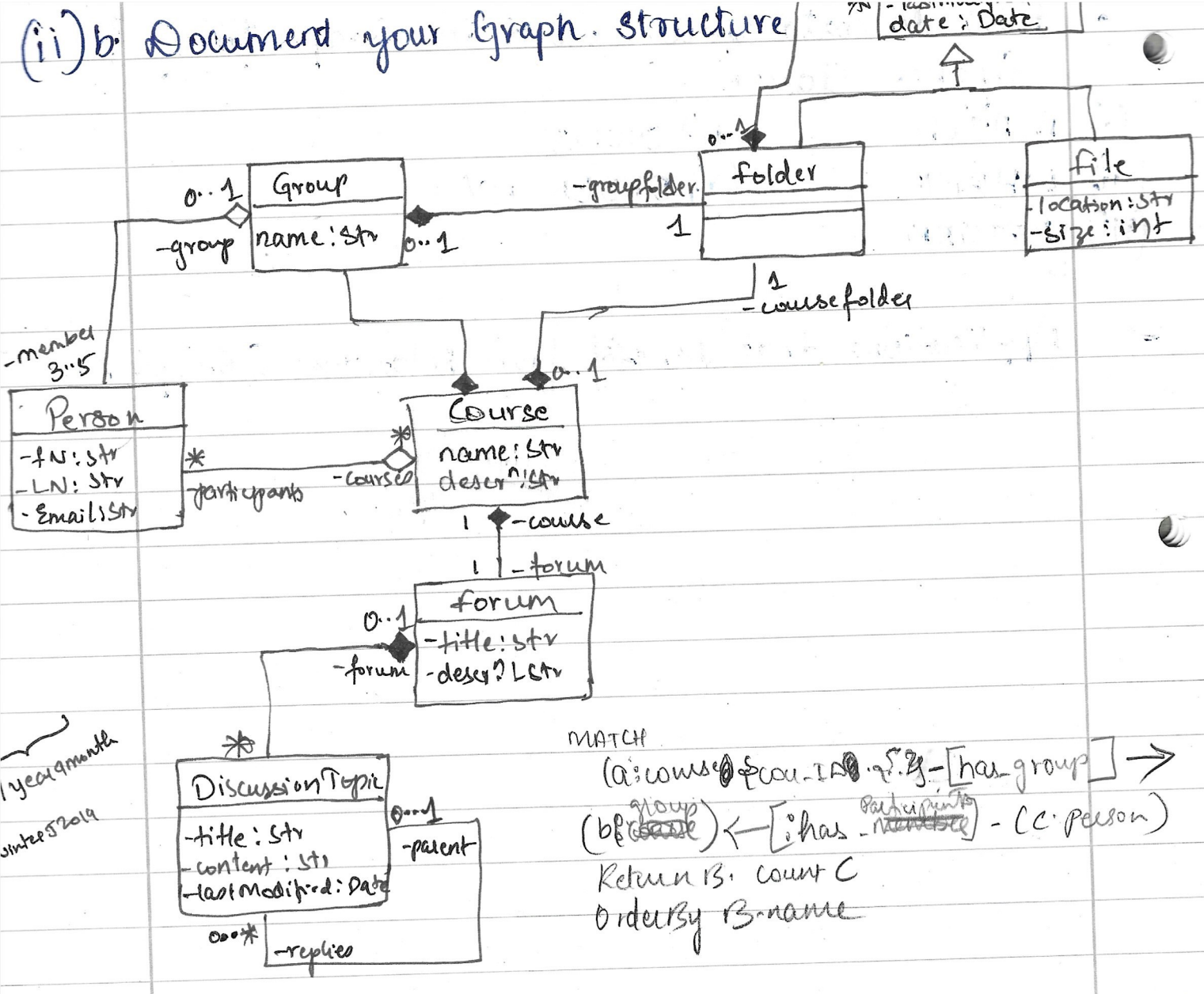
- How do data query work in Graph Query ?

Cypher Clause

(i.) MATCH
(ii.) WHERE
(iii.) RETURN
(iv.) COUNT
(v.) ORDER BY

→ Applications that benefit from Relational & Graph DB.

(ii) b. Document your Graph structure



1 year 9 months

Winter 2014

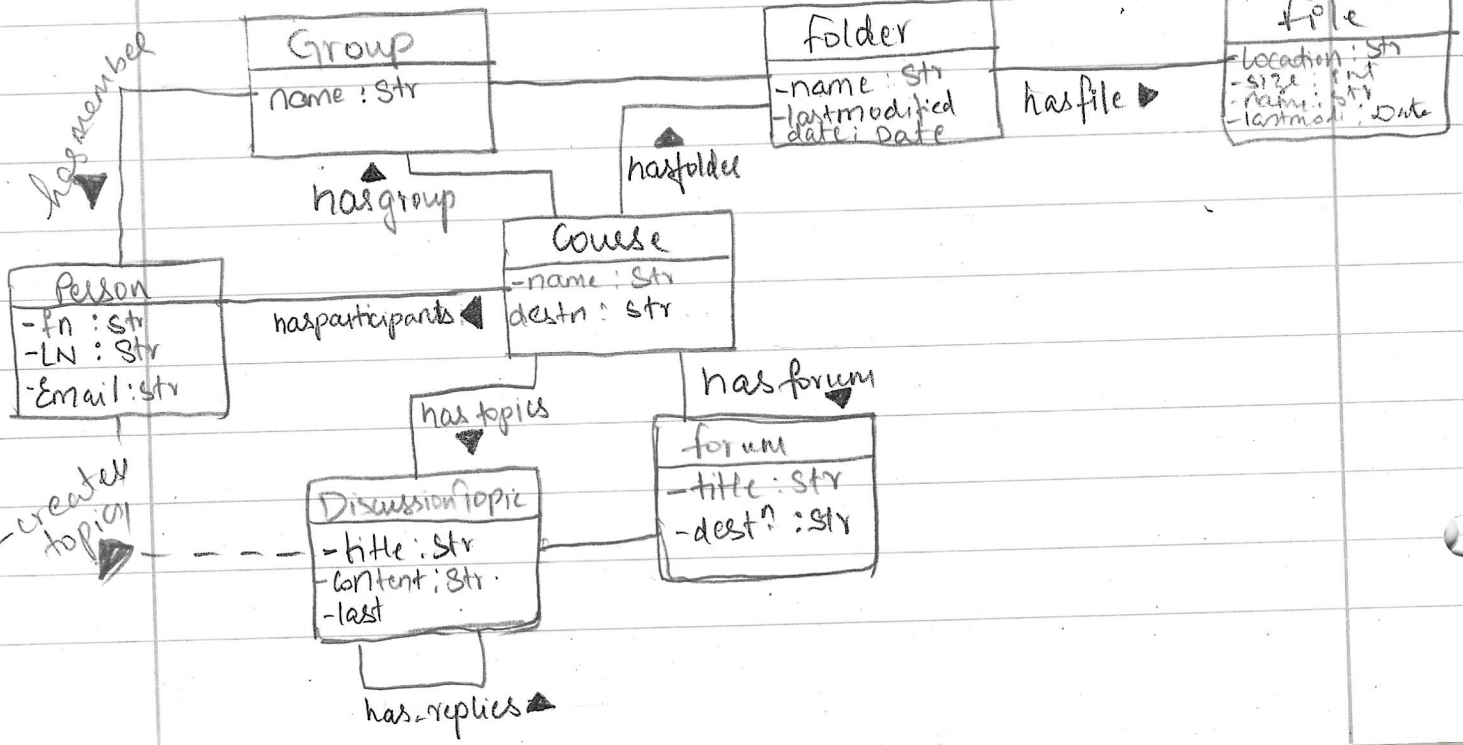
MATCH

(a:course {cou_ID: $cou_ID}) -[:has_group]->

(b:group) <-[:has_participants]- (c:person)

Return b, count c

Order by b.name



(c) Write Cypher Queries:

Try to provide two Cypher MATCH... queries to answer the following questions. If you think that there can be no solution, state why!

- Given a course (referenced by its ID), please provide a sorted list of its groups with the number of members in each group

Match
Return
Order By
Count

```
MATCH
  (a:course {cou_ID:{...}} _ [:enrolled to] →
  (b:group) ← [:part of] - (C:members)
RETURN C
ORDER BY C.name
```

- For each top-level discussion topic, compute the set of person who contributed to the topic and its replies.

Match

```
MATCH
   (a:DiscussionTopic) - [:has_replies] → (b:DiscussionTopic)
   ← [:creates_topic] - (c:person)
RETURN C
COUNT (C)
```

MATCH
RETURN
ORDER (BY)
Count

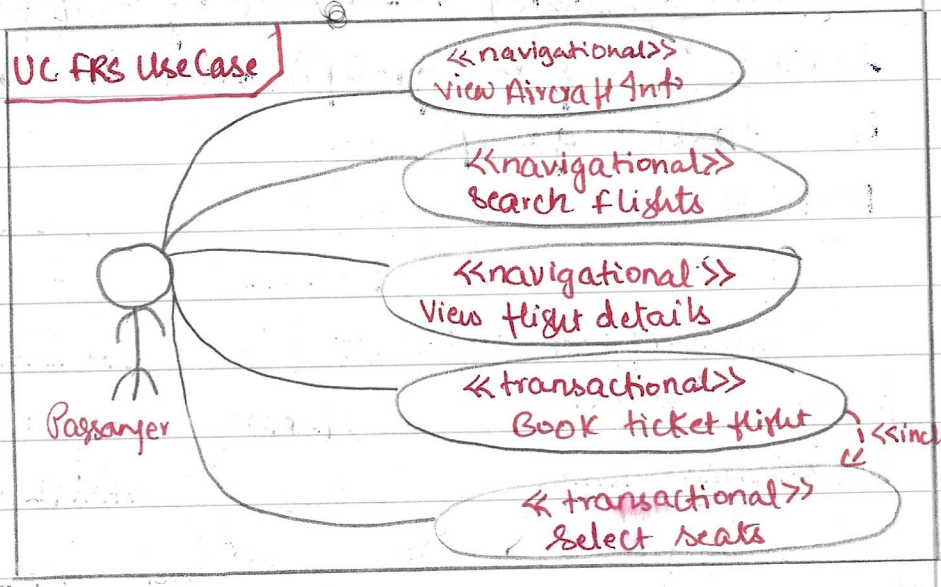
```
MATCH
  (a:course {cou_ID:{...}} - [:has [group] →
  (b:group) ← [:has Member] - (C:person)
RETURN B, count C
Order BY B.name.
```

```
MATCH
(a:DT) - [:has_replies] → (b:DT) ←
 [:creates_topic] - (C: person)
RETURN C
Count (C)
```

Assignment 06

1 UNIT : Navigation and Content Models

In a flight reservation system (FRS), the passangers can search for flights. The following dig. shows the main use cases.



Use Case: It is a model that represents how the user interacts with the system

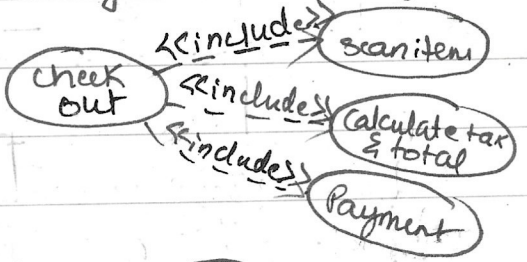
Three types of Use Cases

(i) Navigational : View, browse, read (no modification)

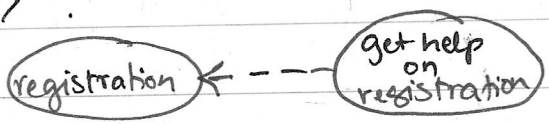
(ii) Transactional : add, modify, create, delete (make changes)

(iii) Personalised : Similar to transactional but related to one / single user only.

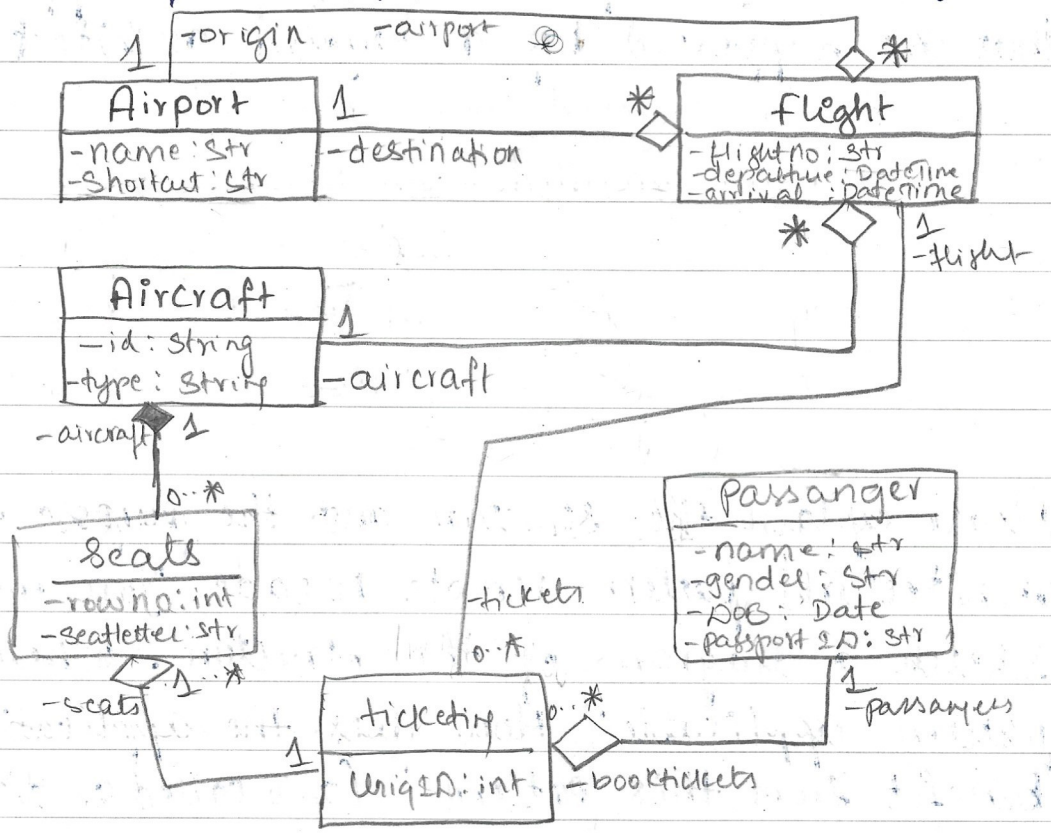
(iv) <<include>> :



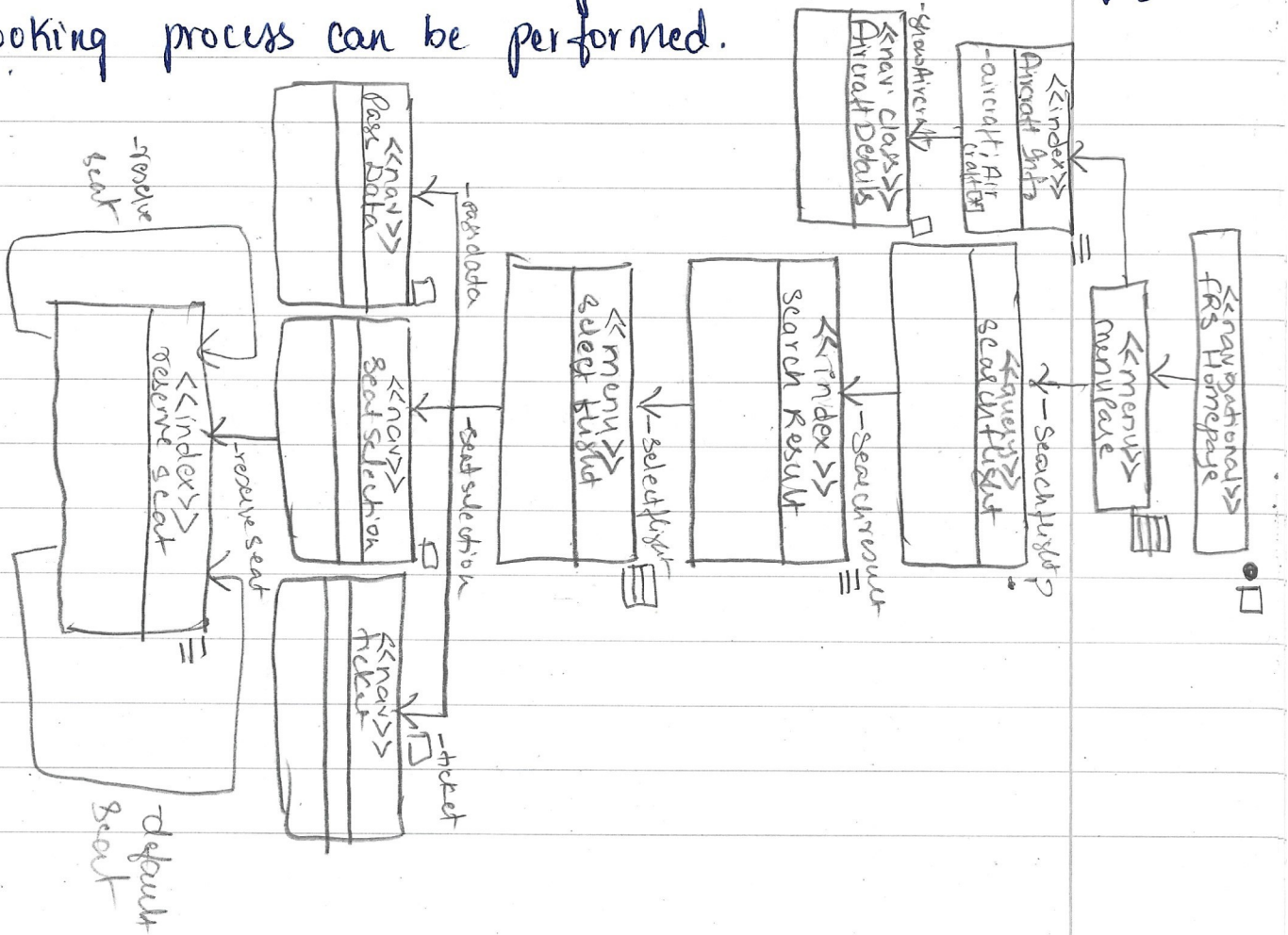
(v) <<Extend>> :



a.) Add the missing info to the domain model : Use classes association, and attributes to describe the relevant data.



(b) Extend the initial UWE navigation model so that the flight booking process can be performed.

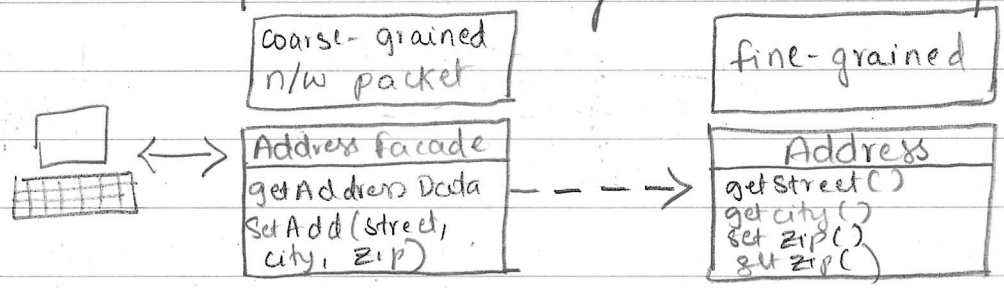


2a) Name two quality measures for web applications that are supported by a layered architecture!

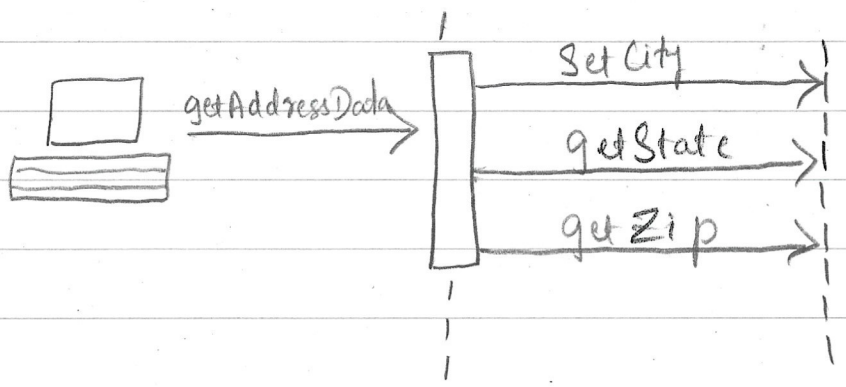
- In a layered architecture, the services provided by a layer are provided to an upper layer and realised using the services of subordinate layers.
- Interface definitions at the boundaries between layers are especially important

2b) Please explain the structure and the purpose of the architectural pattern remote facade. You may also sketch a diagram of the structure. Where is this pattern applicable? How does the application design benefit from this pattern? (also called as distribution pattern)

- Remote facade provides a coarse-grained facade on fine-grained objects to improve efficiency over a network.
- Remote facade contains no domain Logic.
- Remote facade work along with Data Transfer Object (DTO)

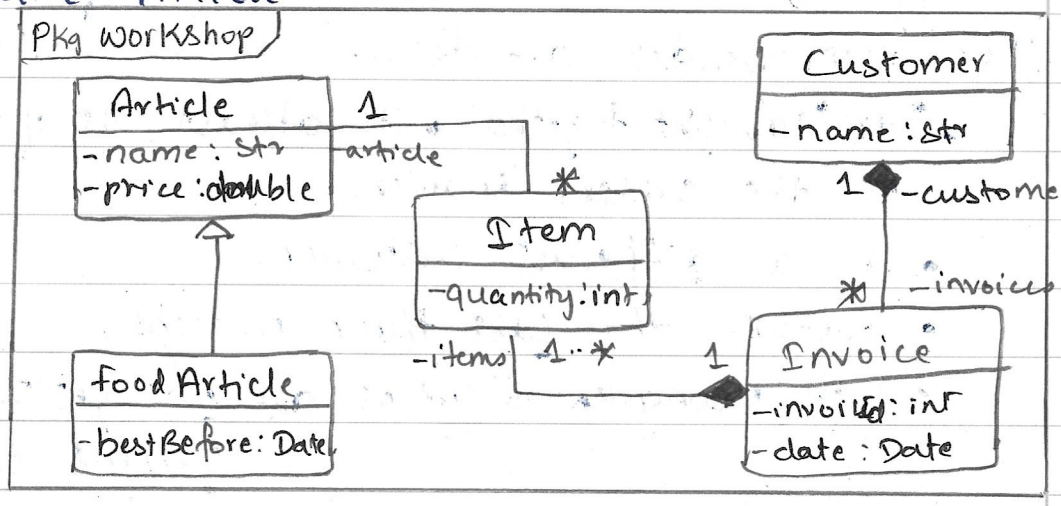


Ex: One call to a facade cause several call from the facade to the domain object.

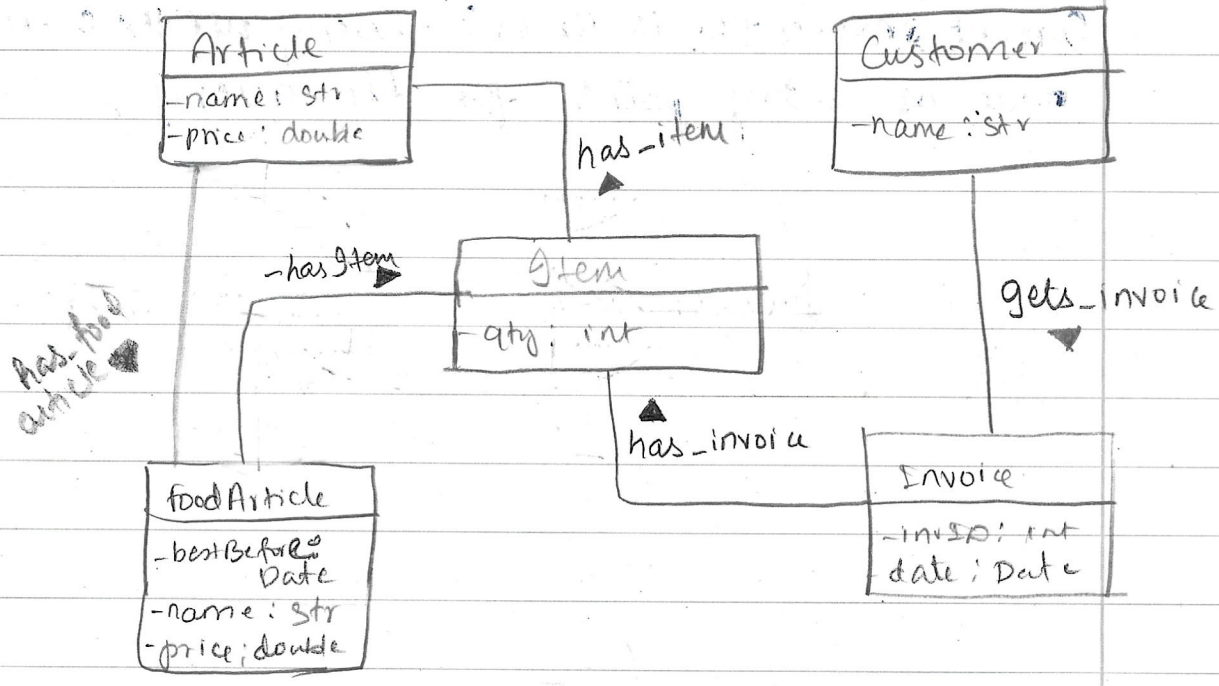


03) Graph Database: Schema and Query

a) The class dig. shows the obj model of a simple web shop application. Customers can buy Articles, all purchases are recorded in invoices. The invoice contains a list of items which contain the quantity for the respective Article.



Graph Schema



(b.) Provide a graph query for a recommender system.
The recommer shall work as follows:
While the customer puts items into the shopping cart (rep. by the ID of the current Invoice), the system shall suggest other articles based on the foll. rule:
The recommended articles...

- Were "brought together" with atleast one of the articles in the current invoice.
- by a different customer.
- within the last three months,
- & the recommended articles are not yet contained on the current invoice

Given your schema, sketch a cypher query that could be used by the recommender system.
Current Invoice ID is know & thereby use the corrosponding node as entry point for the query.

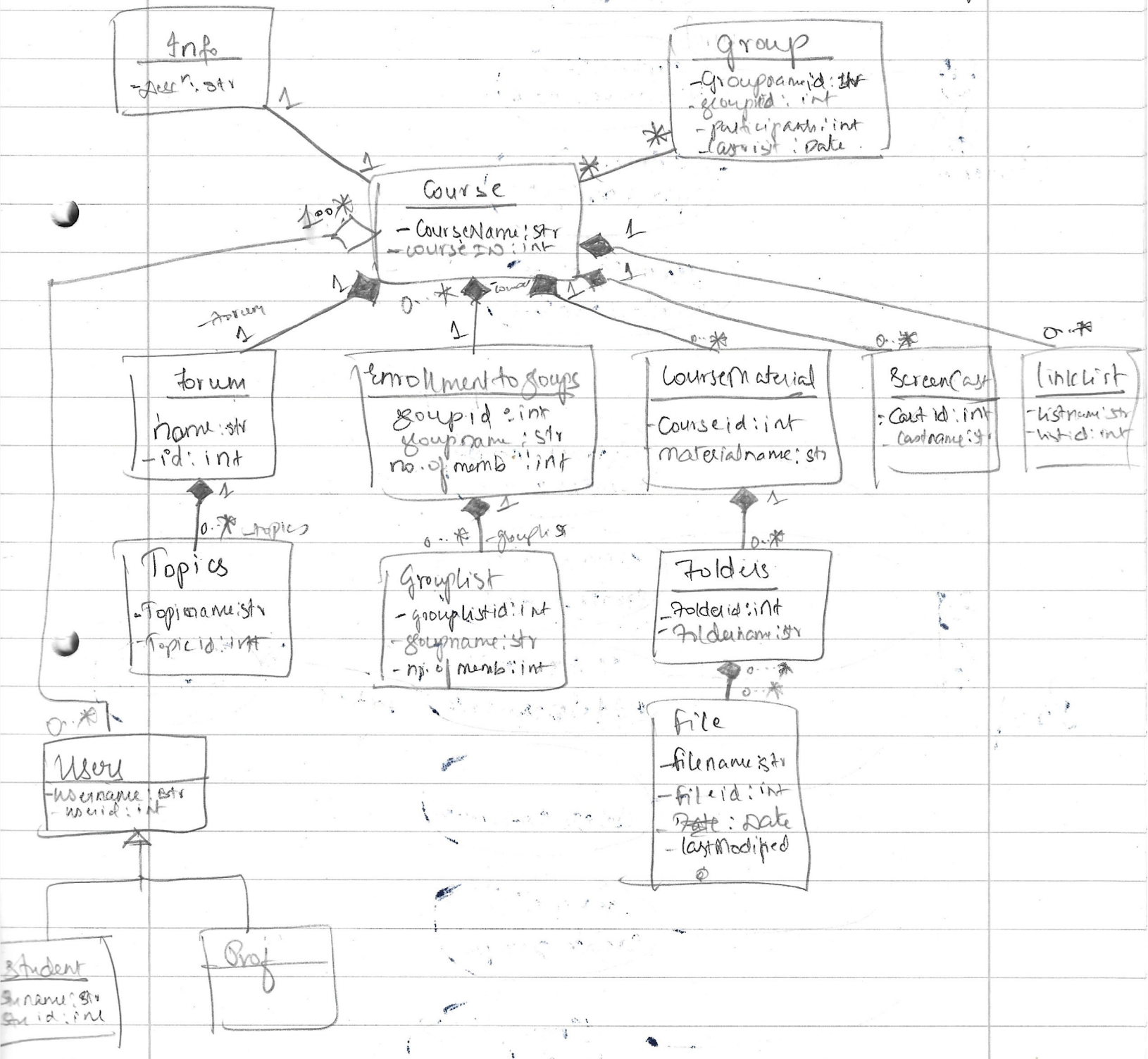
```
MATCH
(a:customer {cus_ID: $...}-[
```

Assignment 01

04/03/2021

1. **Domain Model**

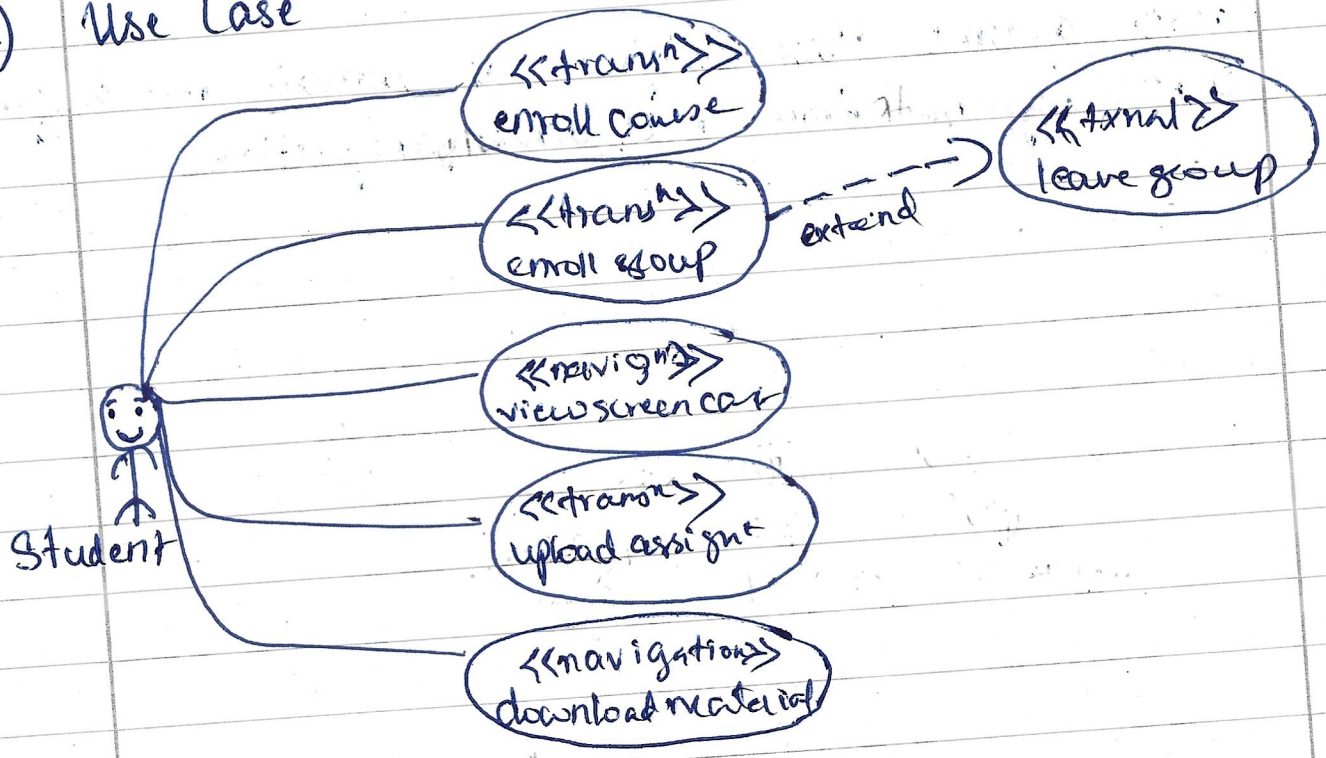
Create a UML class diagram as domain model for the OLAT system from a student's perspective.



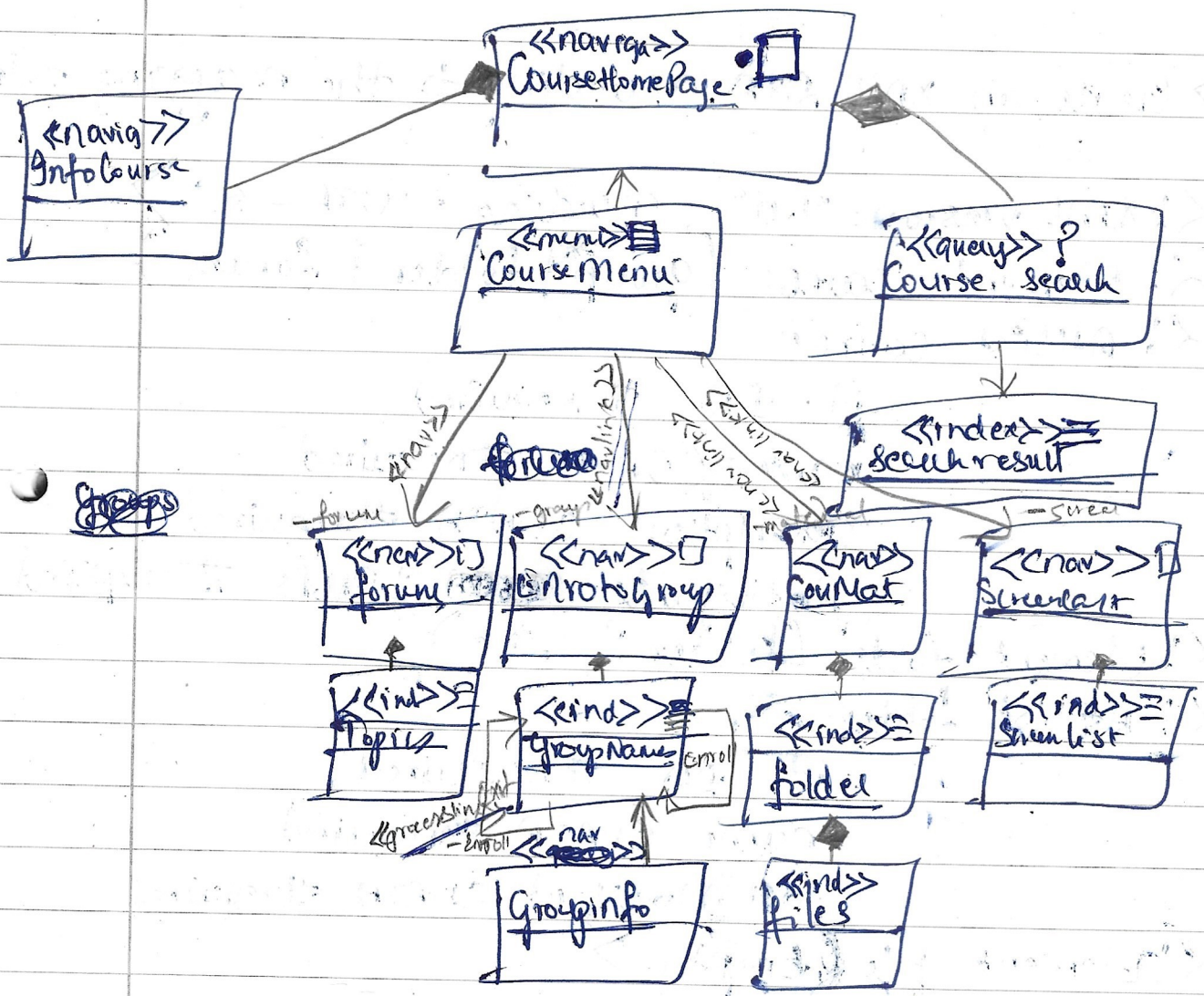
Assignment 02

04/05/2021

9.) Use Case



(b.) Navigational Model



Assignment 03

1. XML DTD and Document from Domain Model.

→ Create an XML DTD according to the mapping rules.

```
<! XML version: "1.0" , encoding = "UTF - 8" >
<! Element course (group* | folder | forum )
<! Attlist course
        id ID #Required
        name CDATA #Required
        description CDATA #required
        Participants IDREFs #implied
<! Element folder (folder?, file)
<! Attlist folder
        id ID #Required
        name CDATA #required
        last Modified CDATA #required
<! Element file (Empty >
        id ID #Required
        location CDATA #reqd
        Size CDATA #rqd
        name CDATA #rqd
        last modified CDATA #rqd.
        folder IDREF #implied.
<! Element group (folder)
<! Attlist group
        id ID #required
        name CDATA #rqd
        members #IDREFS #rqd.
```

<!Element forum (DiscussionTopic*) 1 — 0..*
<!ATTlist forum
id ID #Required
title CDATA #reqd.
description CDATA #reqd.
<!Element discussionTopic EMPTY >
<!ATTLIST discussionTopic
id ID #Required
title CDATA #rqd
Content CDATA #rqd
lastModified #rqd.
author IDREF #implied
Parent IDREF # implied
<!Element Person EMPTY >
0..1
<! ATTLIST Person
id ID #required
fN # CDATA #rqd
LN # CDATA #rqd.
email CDATA #rqd
-group IDREF #implied
topic IDREFS #implied
courses IDREFS # implied.

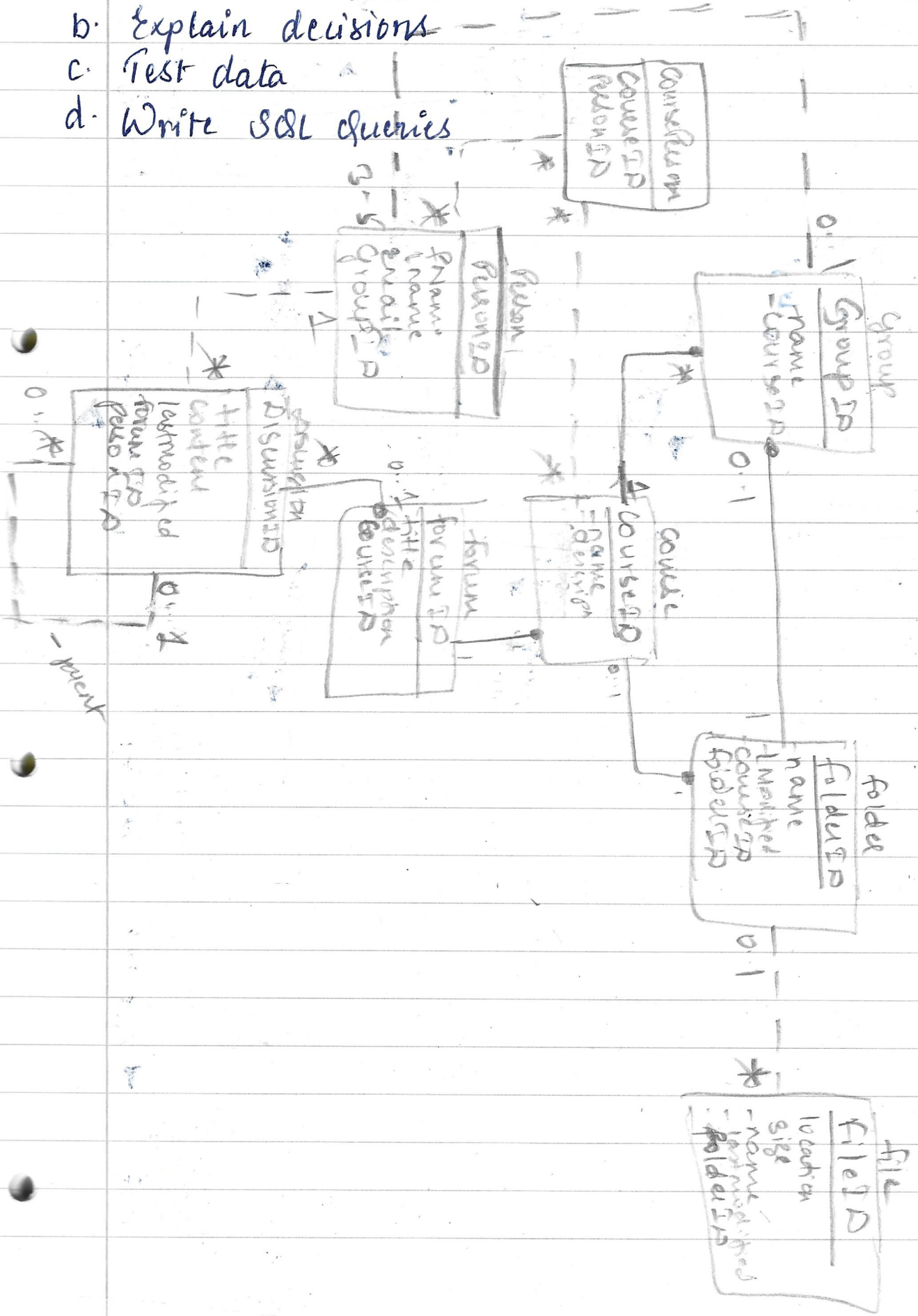
02. Write an XML documents that represents some of the facts from ~~the~~ our course. The document shall confirm to the DTD: Each non-abstract class and every association type shall occur atleast once.

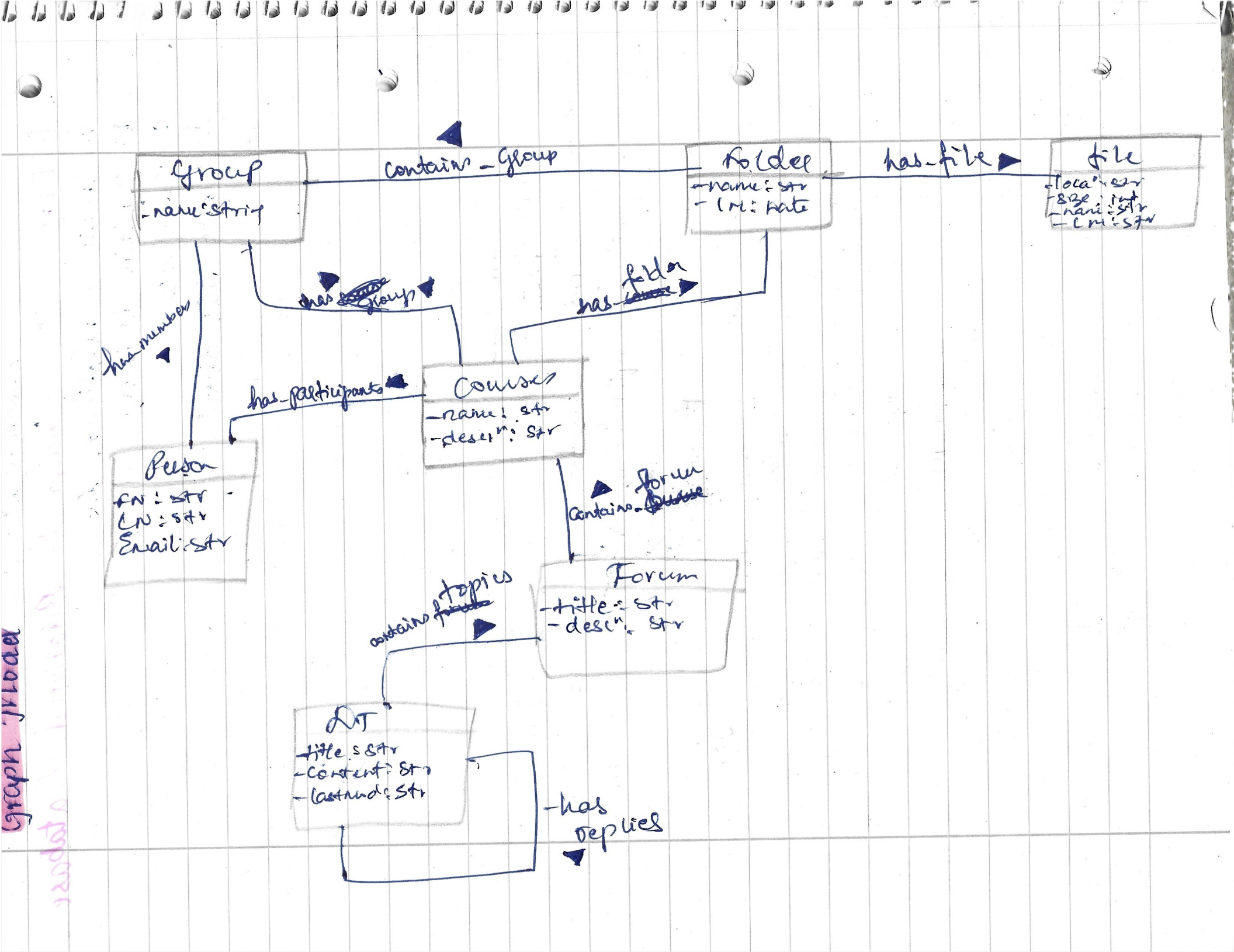
```
<?xml version="1.0" encoding="UTF-8">
<OLAT>
    <Course id="s" name="WE" Desc="..." parts="β, β2">
        <folder id="1" name="" LMD="" name="" >
            <file id="" "locat="" "size="" "name="" "
        </folder>
        <group id="" name="" member="" >
            <folder id="" name="" />
        </group>
        <forum "id"="" descrip="" >
            <OT id="" title="" Content="", LMD="" />
        </forum>
    </Course>
    <Person id="" fN="" LN="" Email="" group="" "
        course="" Bprcs="" />
</OLAT>
```

Assignment 04

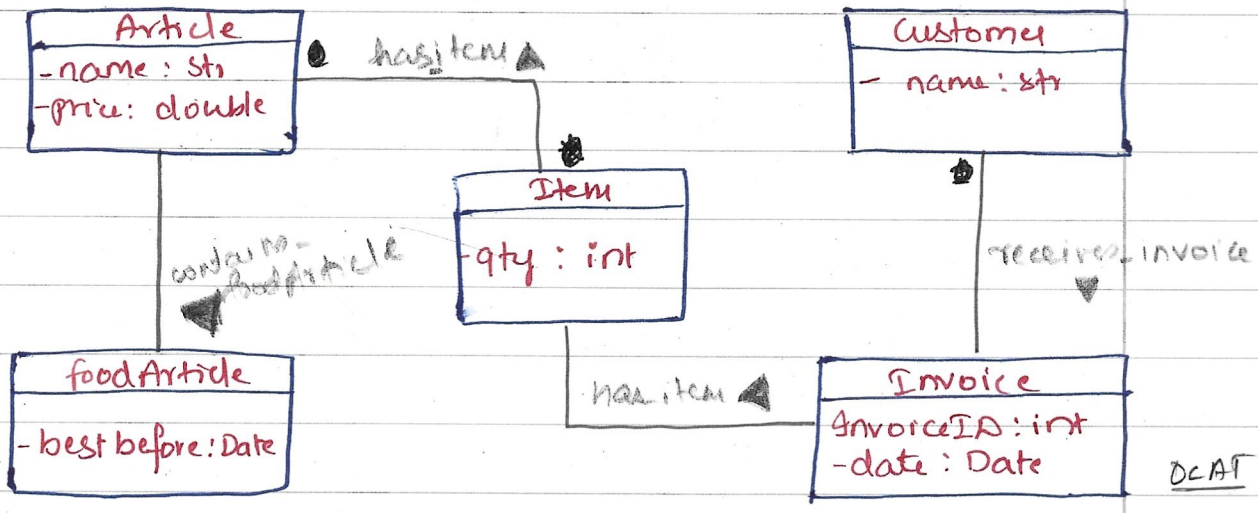
1. Mapping OO schema to Relational Database

a. Create a relational DB
b. Explain decisions
c. Test data
d. Write SQL queries



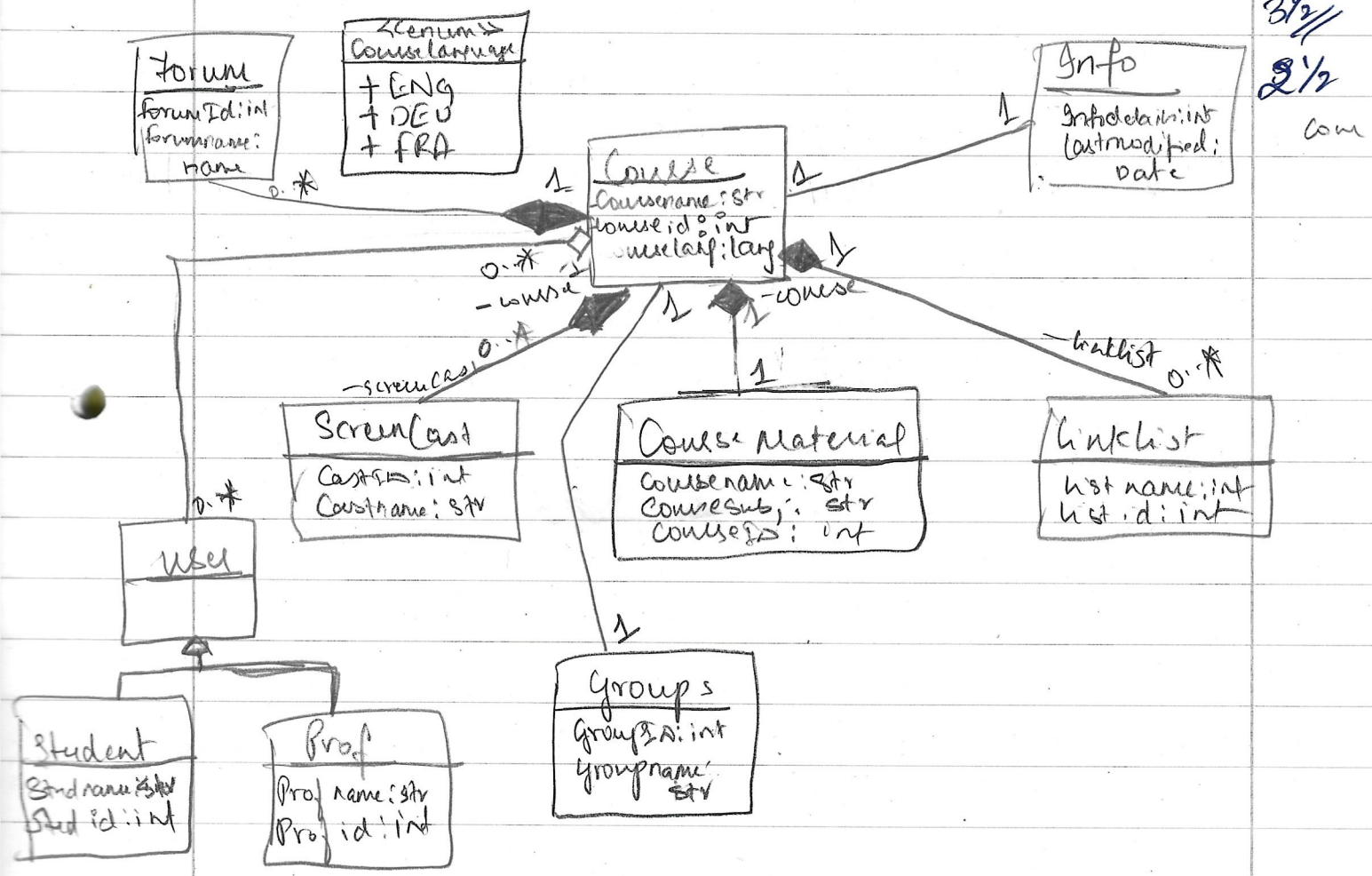

Group
-name: str
contains_Group
Folder
-name: str
-lm: date
has_file
file
-loca: str
-size: int
-name: str
-lm: str
has_member
has_group
has_folder
has_participants
Course
-name: str
-descrn: str
Person
FN: str
LN: str
Email: str
contains_forum
Forum
-title: str
-descrn: str
contains_topics
DT
-title: str
-content: str
-lastmod: str
has_replies

Assignment 06



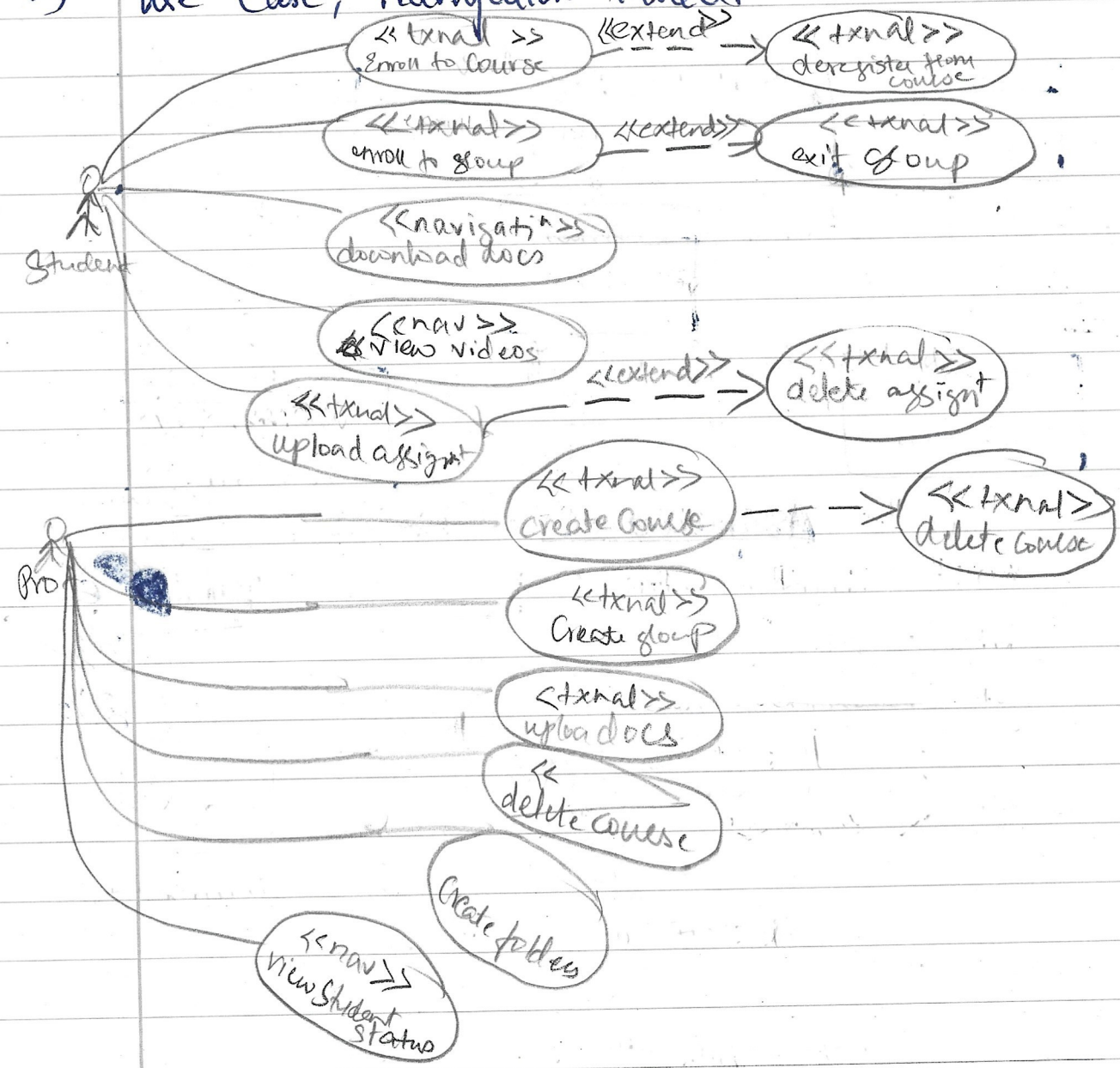
DCAT

Assignment 01

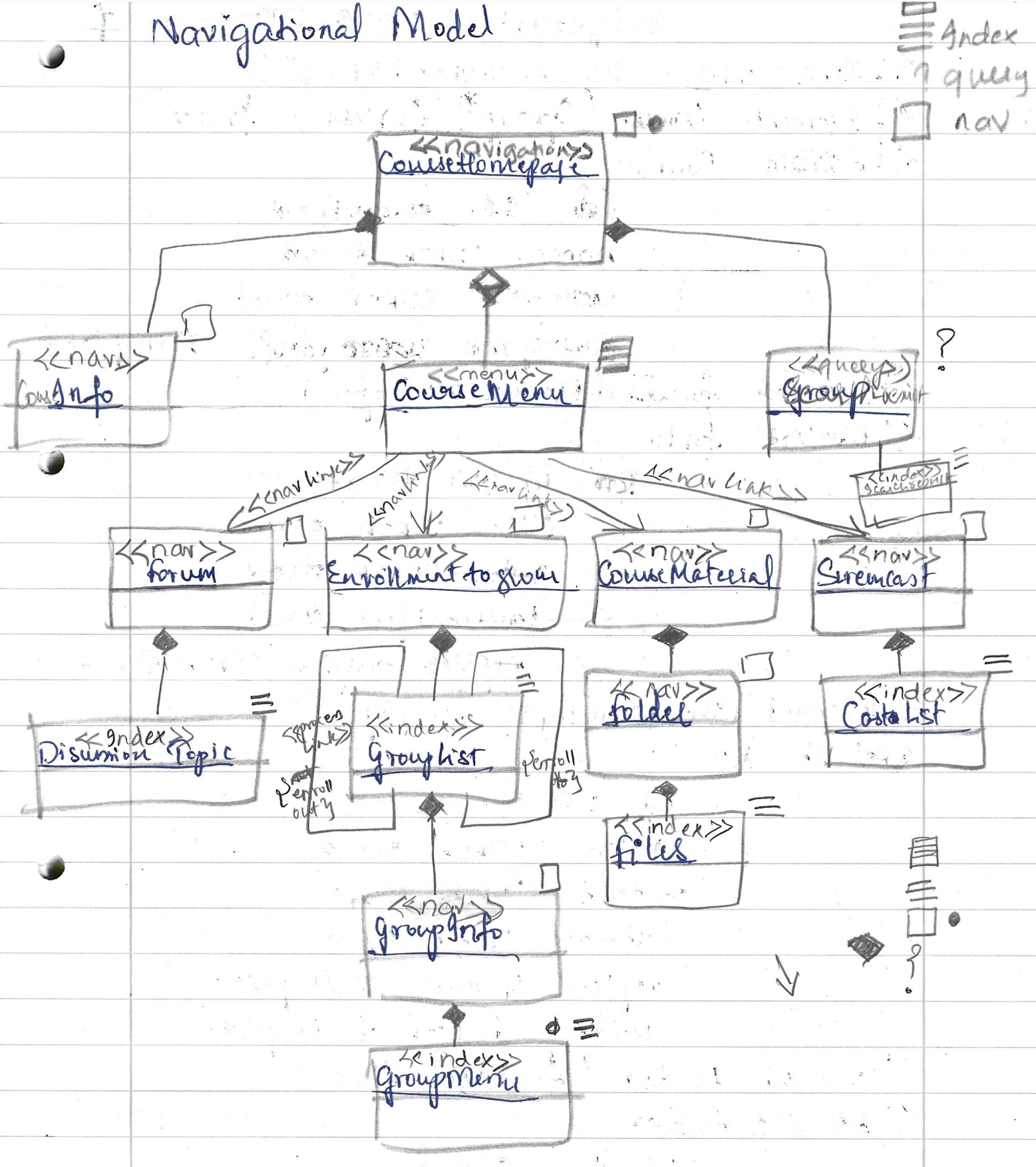


Assignment 02

→ Use Case, Navigation Model.



Navigational Model



Assignment 03

<!XML version="1.0" encoding="UTF-8" >
<!Element Course (Group*, forum, folder)
<!Attlist Course
id ID #required
name CDATA #rqd
description CDATA #rqd
Participants IDREFS #implied
<!Element forum (Discussion Topic)
<!Attlist forum
id ID #rqd.
title CDATA #rqd
~~Content CDATA #rqd~~
~~Lastmodified #rqd~~
~~replies~~ description CDATA #rqd.
<!Element DiscussionTopic Empty >
<Attlist DT
id ID #rqd.
title CDATA #rqd
content CDATA #rqd
Lastmodi CDATA #rqd
replies IDREFS #implied
author IDREF # implied
<!Element Folder (folder?, file)
<!Attlist folder
id ID #rqd.
name CDATA #rqd
Lastmodified CDATA #rqd.
<!Element file Empty >
<Attlist file id ID #rqd

location CDATA #rqd
size CDATA #rqd
name CDATA #rqd
lastModified CDATA #rqd.

<!Element Person Empty>
<!Attlist Person
id ID #rqd.
fN CDATA #rqd
LN CDATA #rqd
Email CDATA #rqd.

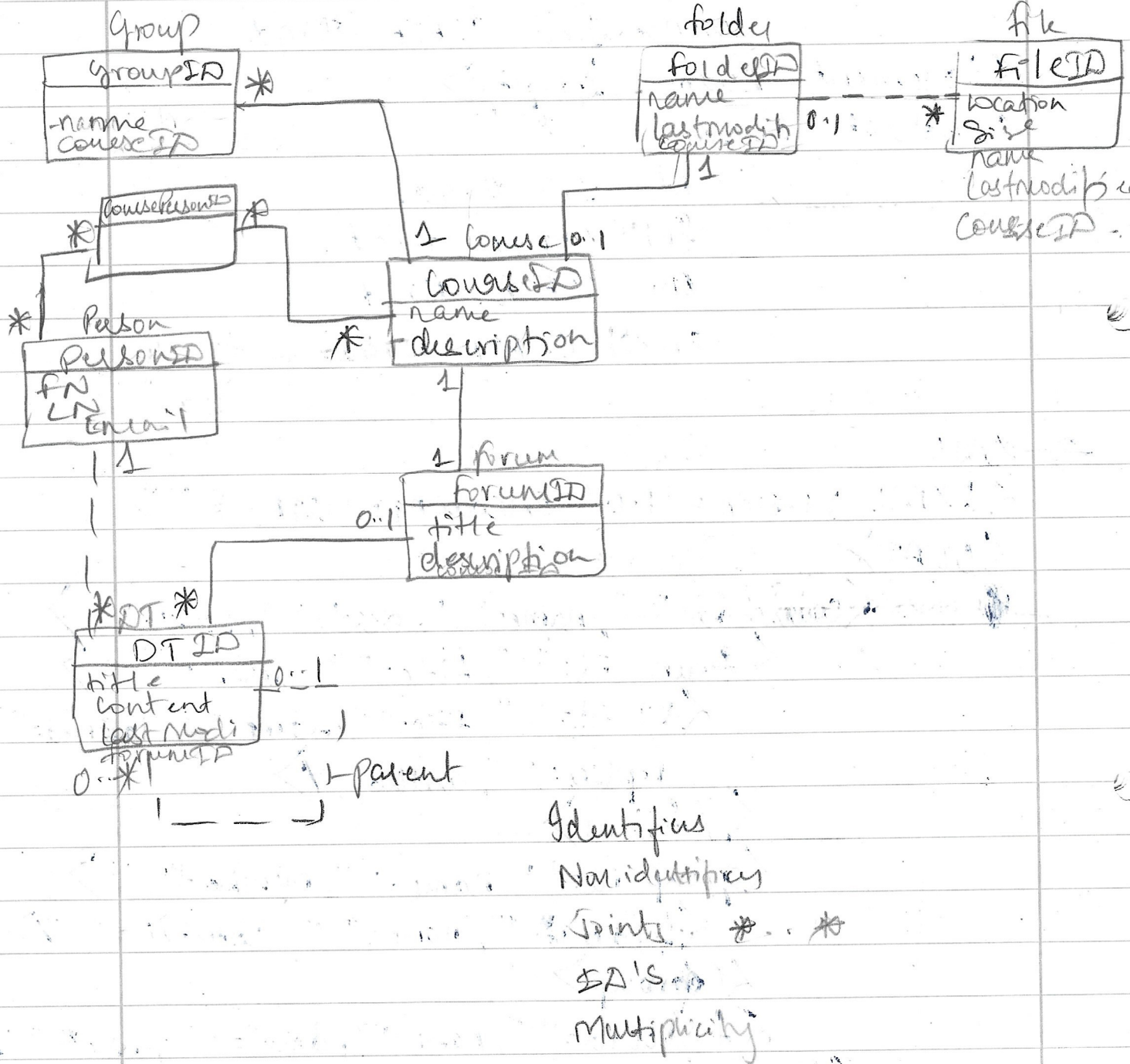
→ XML

<?XML Version="1.0" Encoding: "UTf-8" >
<OLAT>
<Course id:" ", name=" ", desc"=" ", Partic"=" " >
<forum id:" ", title=" ", desc"=" " >
<DT id=" ", title:" ", content:" ", lastmod:" ",
replies=" ", author "/>
</forum>
<group id:" ", name=" ", member:" " >
<folder id:"", name=" ", lastmodi=" "/>
</group>
<folder id:" ", name:"", desc"=" " partic"=" "/>
</Course>
<Person id =" " fN=" ", LN=" ", Email=" "/>
</olat>

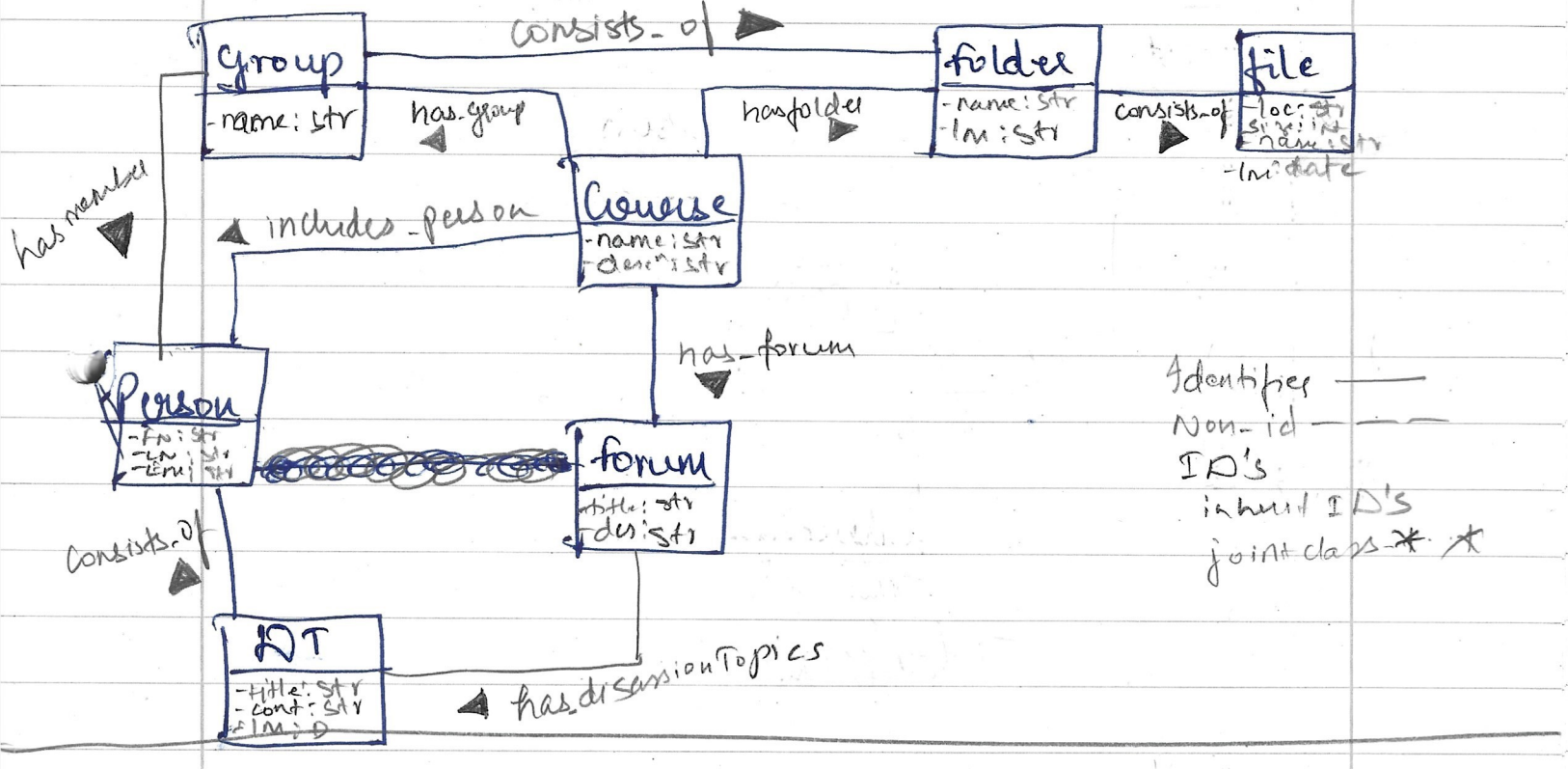
Assignment 04

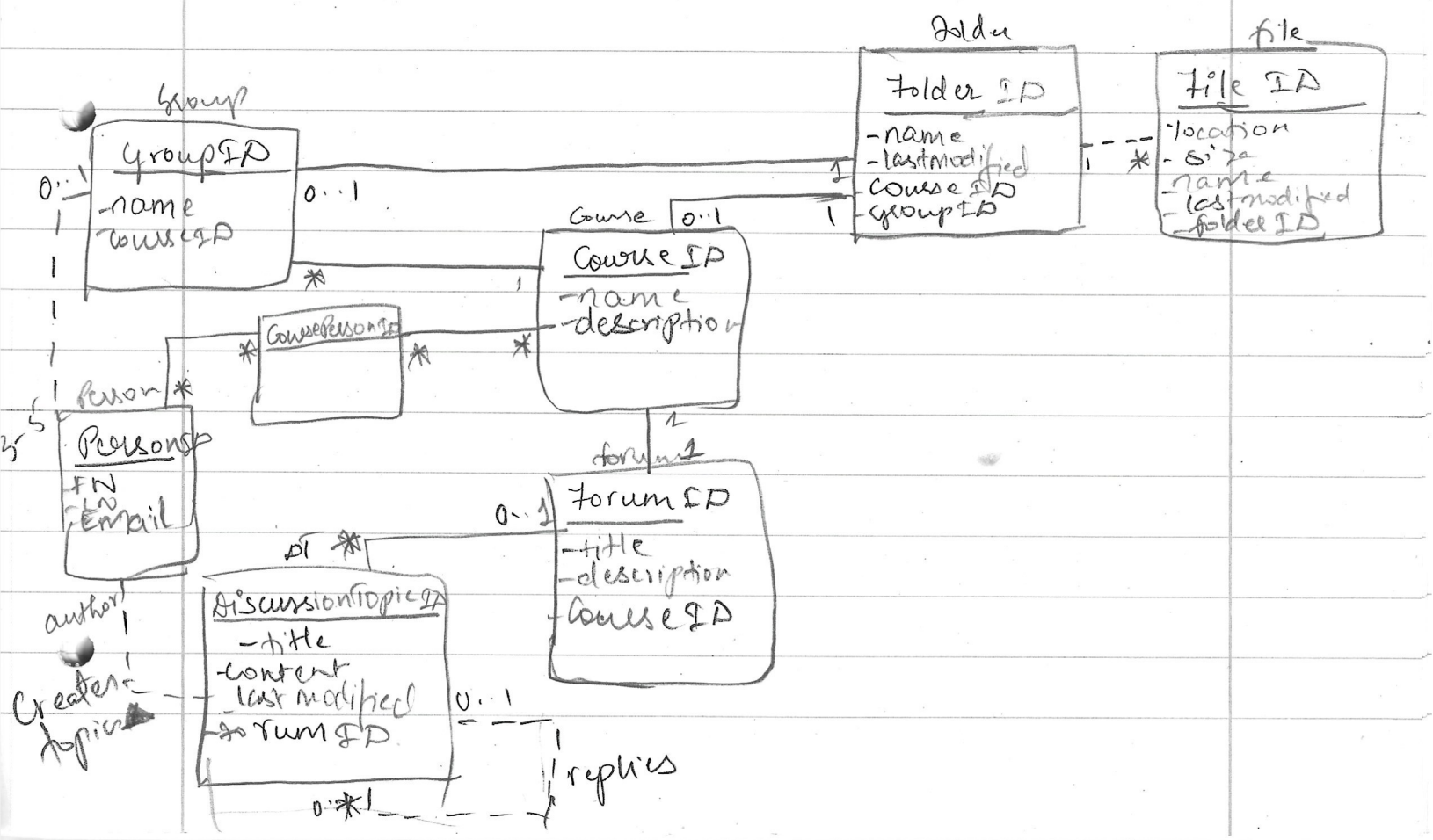
1. Mapping OO schema to RDB.

Non identifying ID's



Assignment 06





Assignment 06

Cypher Query