

ADVANCED SOFTWARE DESIGN

LECTURE 2

DOMAIN MODELLING

Kiko Fernandez-Reyes

OVERVIEW

- What is a domain model?
- Why is it important?
- Representation: Classes, relations and attributes
- In-class example

WHAT IS A DOMAIN MODEL?

Representation that **illustrates meaningful concepts from a specific domain**, e.g.:

- **Stock prediction app**

Stock, ISIN, index, currency, exchange, ...



- **Wine tracker app**

Wine, provenance, vintage, type of grape, ...

- **Scrabble app**

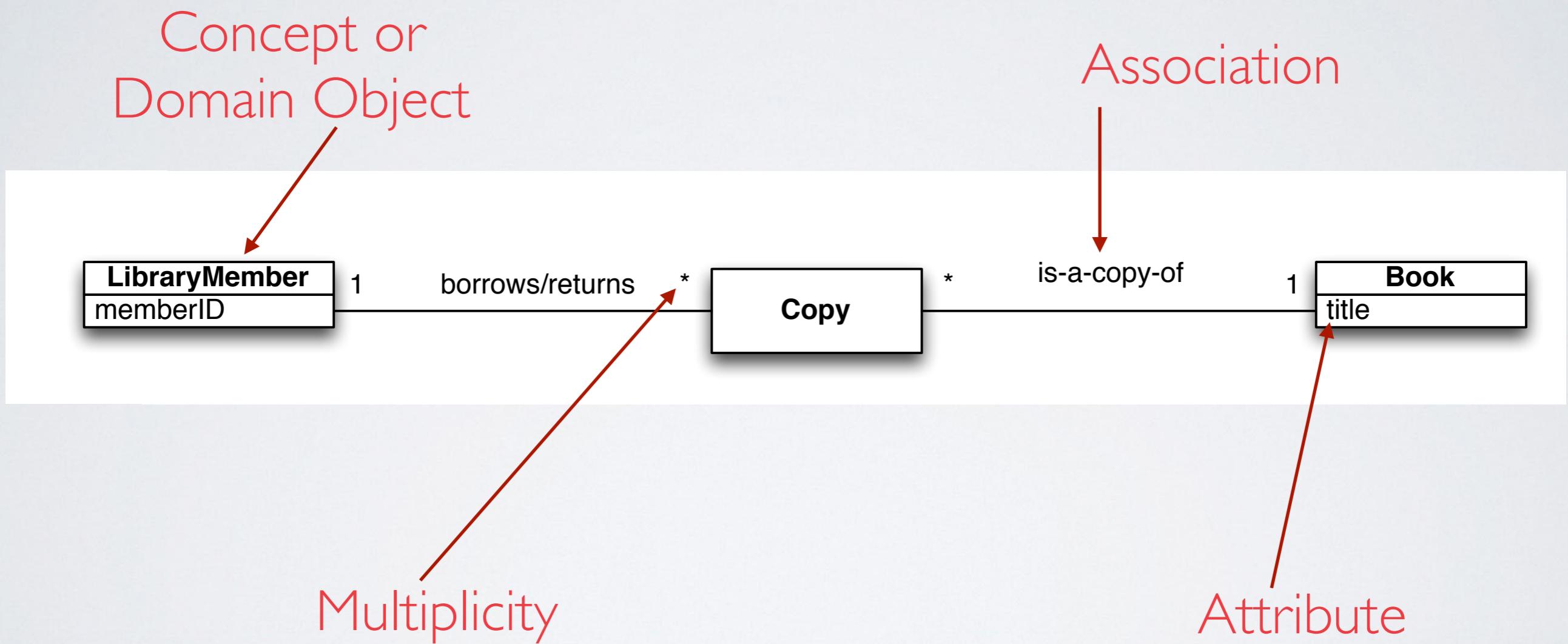
Words, tiles, players, points, ...



LIBRARY EXAMPLE

- You are member of a library
- You would like to take a copy of a book

DESIGN NOTATION



WHY A DOMAIN MODEL?

- To **communicate** with other people
- Understand **key concepts** of certain domain
- Capture **common abstractions**
- Express high-level **relations** of the domain
- **Visual representation of concepts**



GENERAL GUIDELINES

- **Model just domain artefacts, not software artefacts**
- Model a part of the domain at a time
- **Getting it perfect isn't a requirement** – not cost effective
- When classes and their behaviour are modelled in more detail, resulting **class model will often deviate from domain model.**

LEARN BY DOING:

How would you model Spotify?

MODELLING SPOTIFY

Soon, we are going to:

- **Identify domain elements:** Song, Artist, etc
- **Identify relationships:** Artist **is a member of** a Group
- **Add multiplicity**



ACTIVITY I: ELICIT CORE DOMAIN ELEMENTS

IDENTIFYING DOMAIN ELEMENTS

Select key domain abstractions.

nouns and noun phrases – gives candidate classes

tangible or ‘**real world**’ things – book, copy, course

roles – library member, student, director of studies

events – arrival, leaving, request

interactions – meetings, intersection

DISCARDING CANDIDATES

redundant – encompassed by some other compass, e.g.:

loan vs *short term loan*. Select most general.

vague – e.g., *item*

an event or operation – something done by system. If it has no state, discard it.

meta-language – language used to define things. *requirements*, *system*, ...

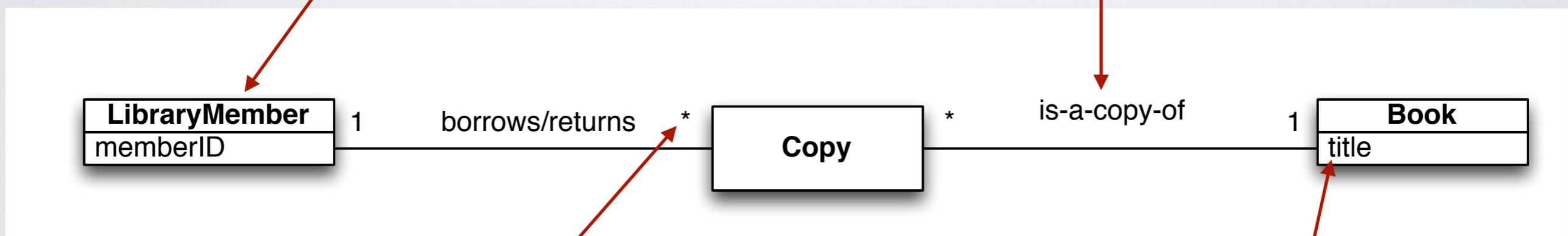
outside the scope of the system – e.g., *library*, *week*, ...

an attribute – something simple, an attribute of another class.



DESIGN NOTATION: DOMAIN OBJECT

**Concept or
Domain Object**



Group work

FIND DOMAIN ELEMENTS



Group work

Domain Elements



(missing items: concert, lyrics, etc)



ACTIVITY 2: ELICIT CORE ASSOCIATIONS

ASSOCIATIONS

An association is a **relationship between concepts** that indicates **some meaningful and interesting connection**.

Clarify our understanding of the domain by describing objects in terms of how they work together

FINDING ASSOCIATIONS

Concepts A and B are associated if:

- A **is a physical/logical part of** B, e.g.
Wing-Airplane, Bottle-Tag
- A **is physically/logically contained in** B, e.g. ItemDescription-Catalogue
- A **is a line-item of a transaction or report** B, e.g. SaleLineItem-Sale
- A **is a member of** B, e.g. Cashier-Store
- A **uses or manages** B, e.g. Cashier-Register



FINDING ASSOCIATIONS

Concepts A and B are associated if:

- A **is a description of** B, e.g. *ItemDescription-Item*
- A **is logged/reported/captured in** B, e.g. *Sale-Register*
- A **is an organisational subunit of** B, e.g. *Department-Store*
- A **communicates with** B, e.g. *Customer-Cashier*
- A **is next to** B, e.g. *City-City*

FINDING ASSOCIATIONS

Concepts A and B are associated if:

- A **is related to a transaction** B, e.g. *Customer-Payment*
- A **is a transaction related to another transaction** B, e.g. *Payment-Sale*
- A **is owned by** B, e.g. *Register-Store*
- A **an event related to** B, e.g. *Sale-Customer*

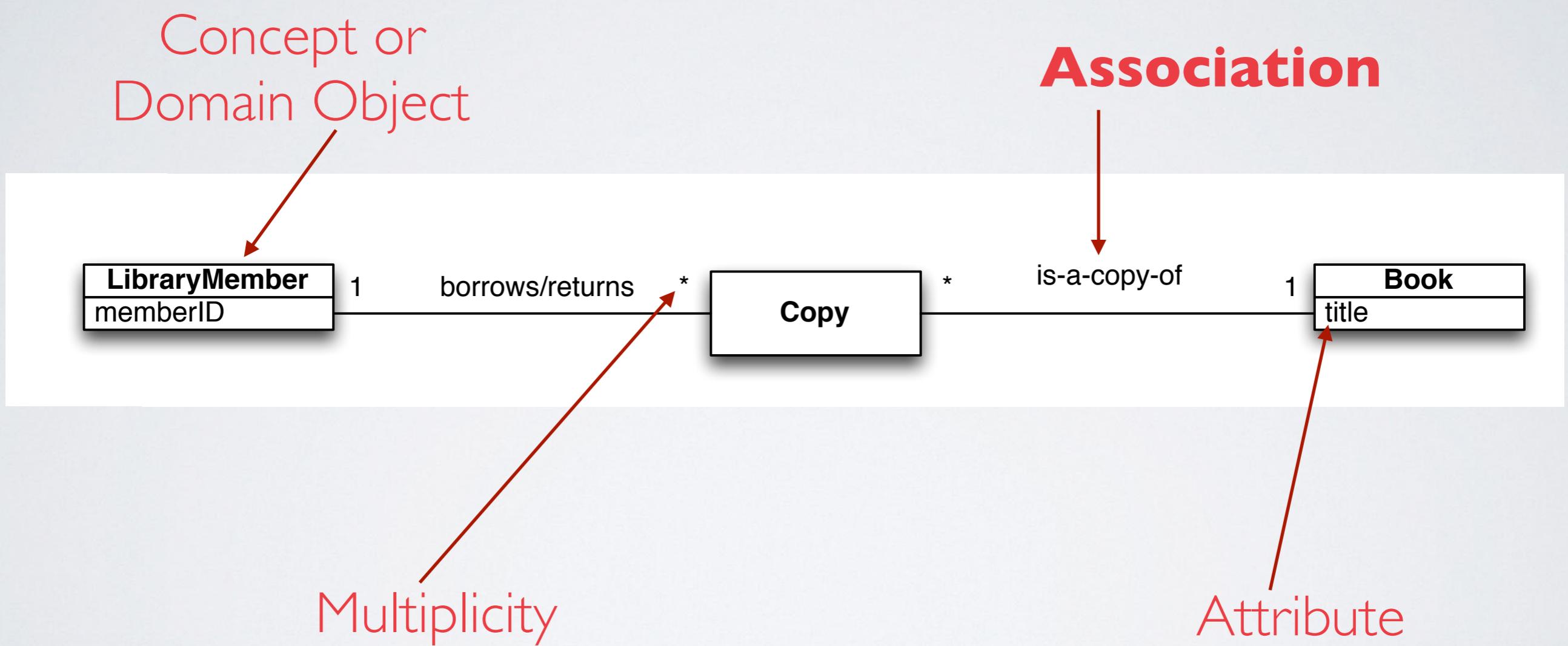
FINDING ASSOCIATIONS (FROM SOFTWARE SIDE)

Concepts A and B are associated if:

- an A **sends a message to** a B
- an A **creates** a B
- an A **has an attribute whose values are one or more** Bs
- an A **receive a message with** a B as an argument

In short: if A **has to know something about some** B

DESIGN NOTATION: ASSOCIATION



Group work

ADD ASSOCIATIONS TO
DOMAIN MODEL DIAGRAM



Group work

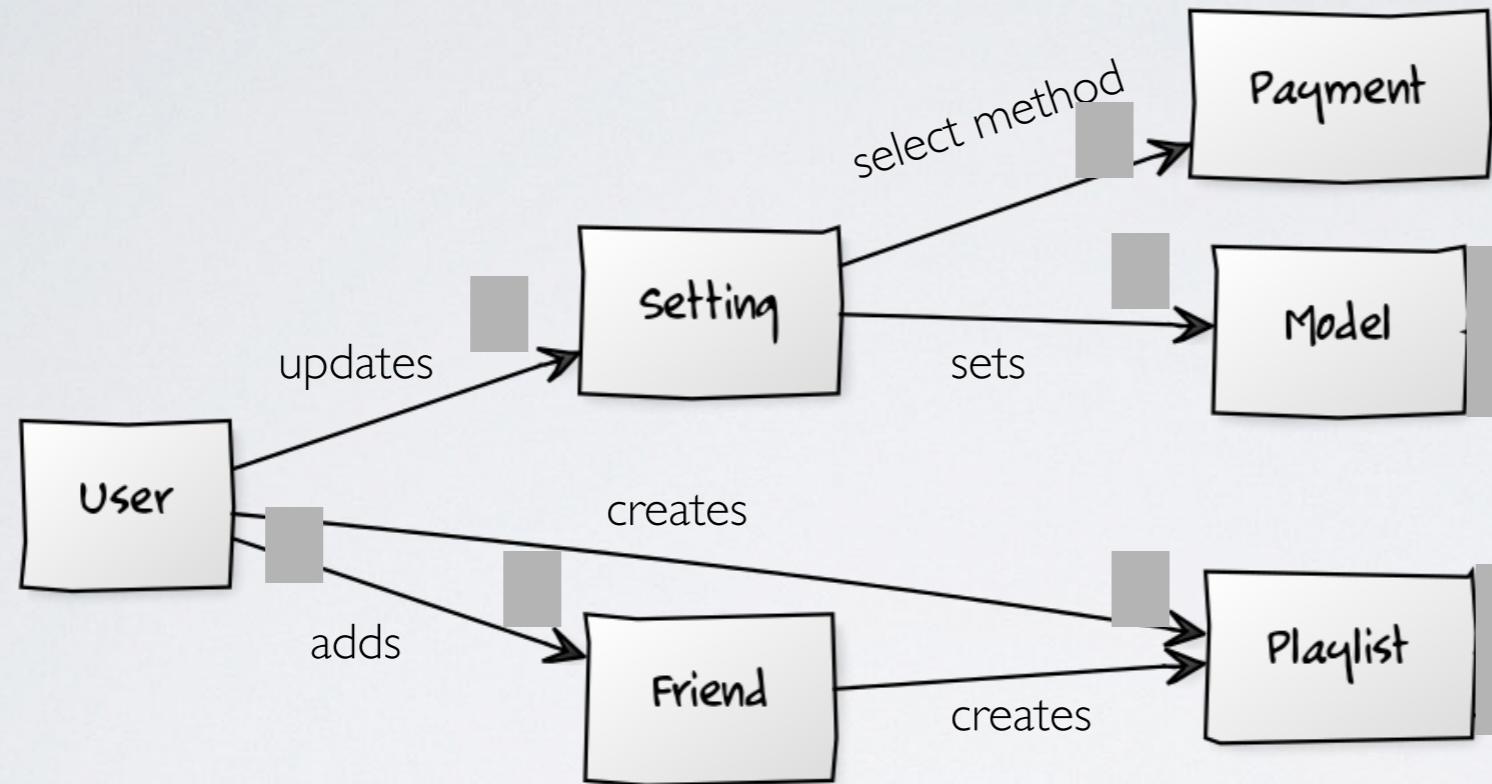
Associations

(missing items: add songs, bands, etc)



Group work

Associations

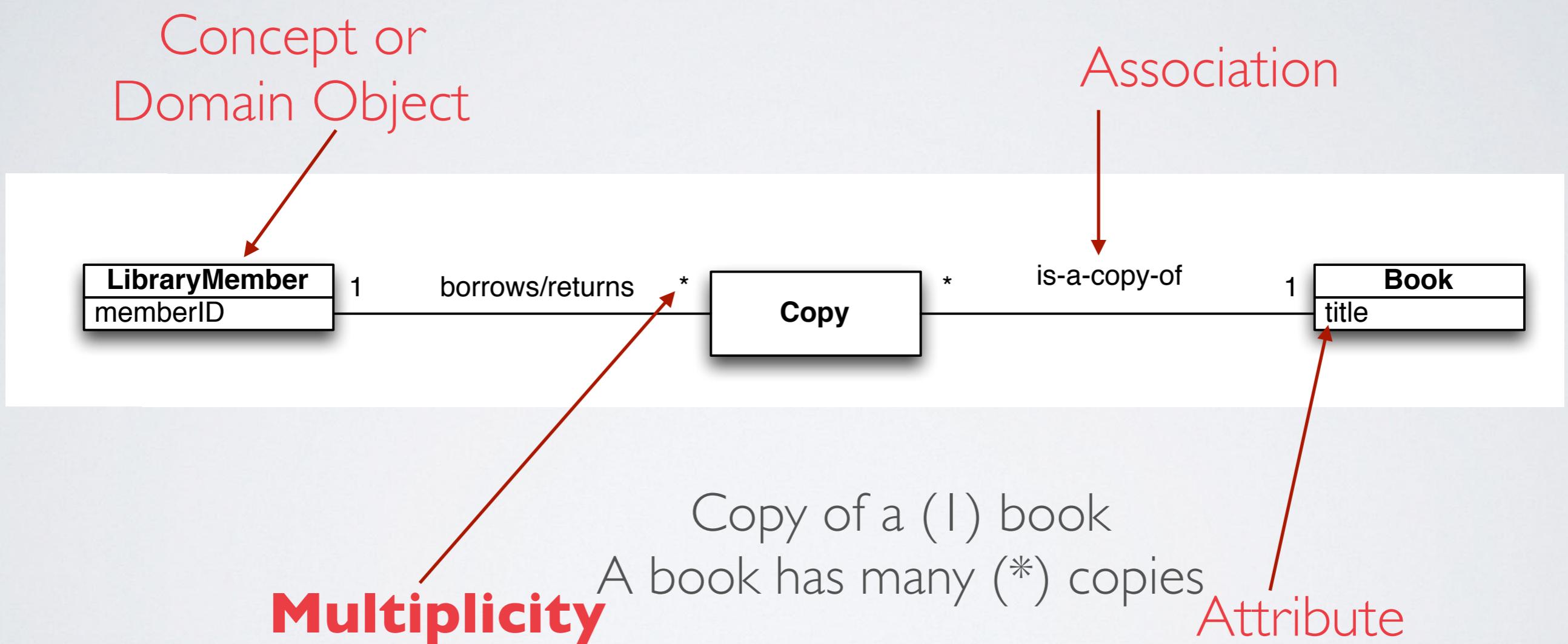


(missing items: add songs, bands, etc)



ACTIVITY 3: REFINE DOMAIN MODEL, MULTIPLICITIES

DESIGN NOTATION: MULTIPLICITY



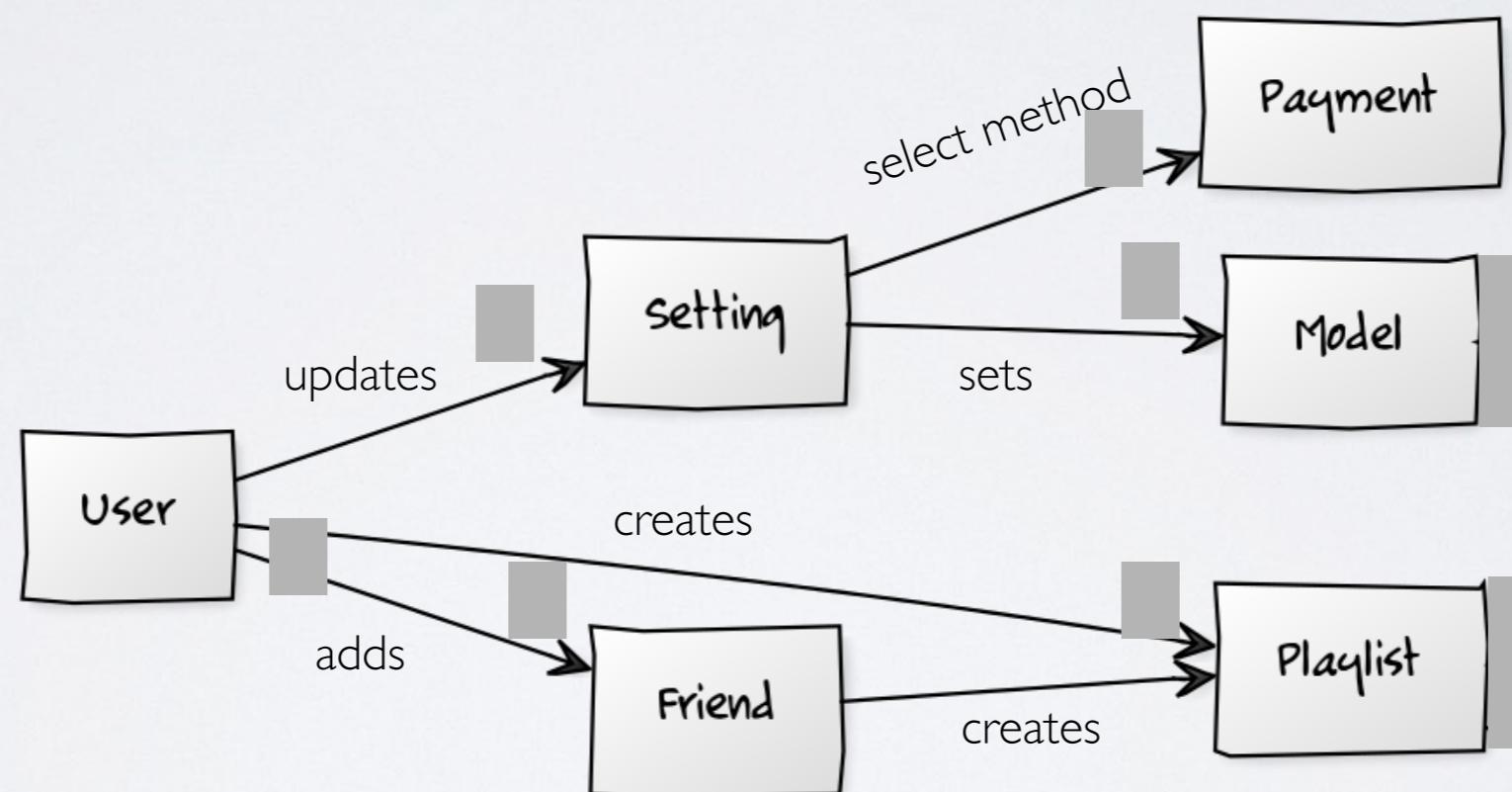
Group work

ADD MULTIPLICITY



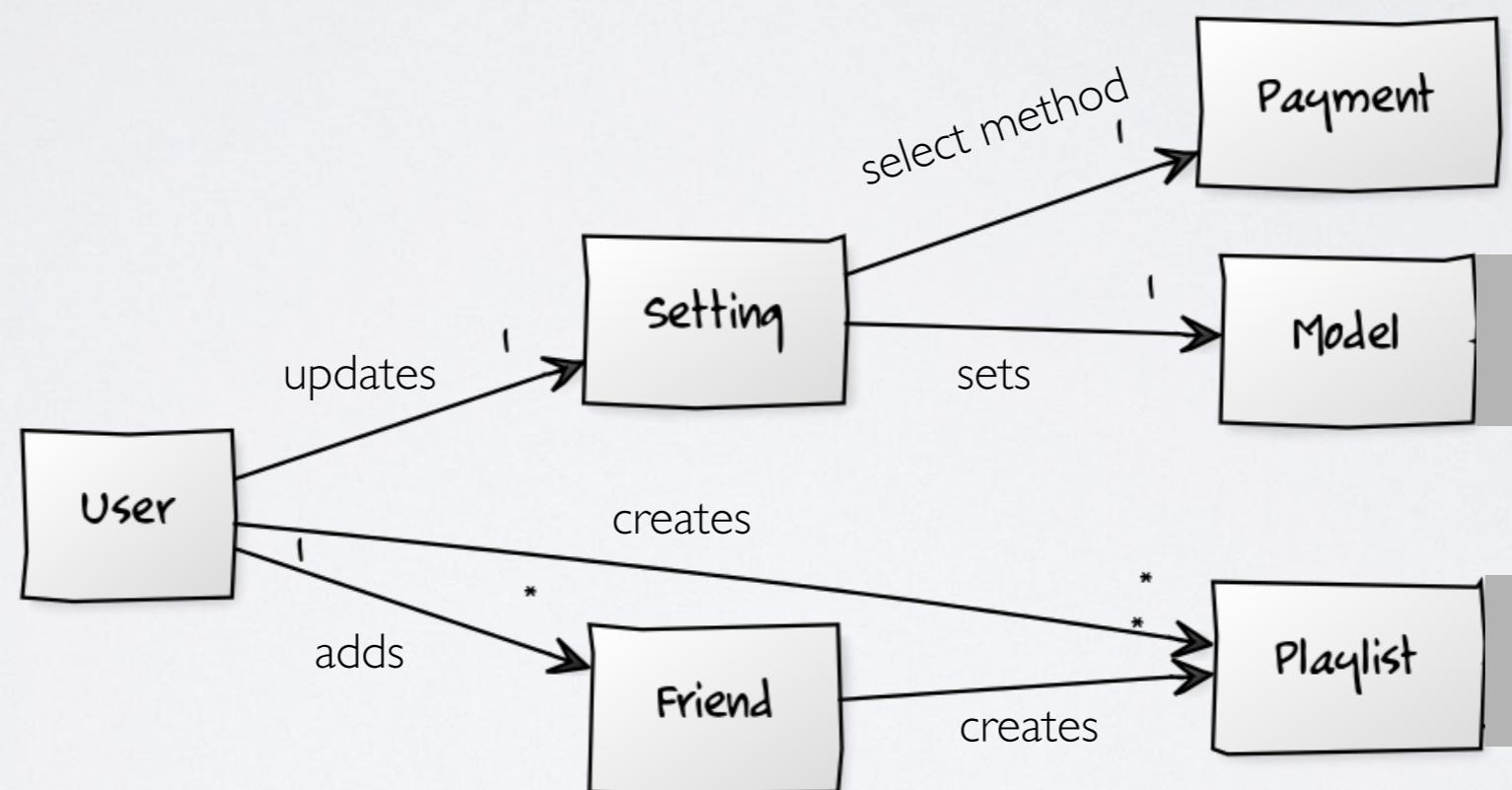
MODELLING SPOTIFY

Multiplicity for this example?



MODELLING SPOTIFY

Multiplicity for this example?

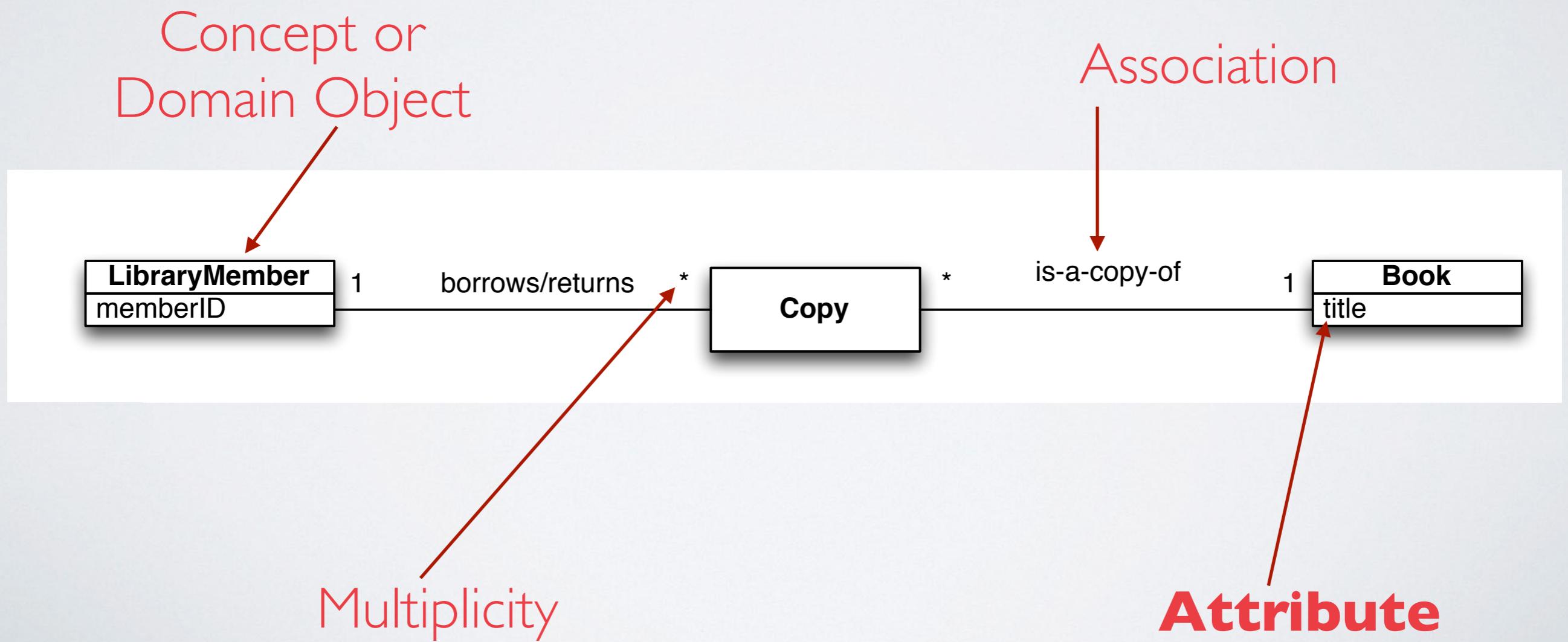


ACTIVITY 4: REFINE DOMAIN MODEL ATTRIBUTES

ATTRIBUTES

State of objects (excluding associations)

Associations are not modelled as attributes



ATTRIBUTES

Generally **attributes are only of primitive type**: ints, strings, enums, ... or library classes — NOT domain concepts

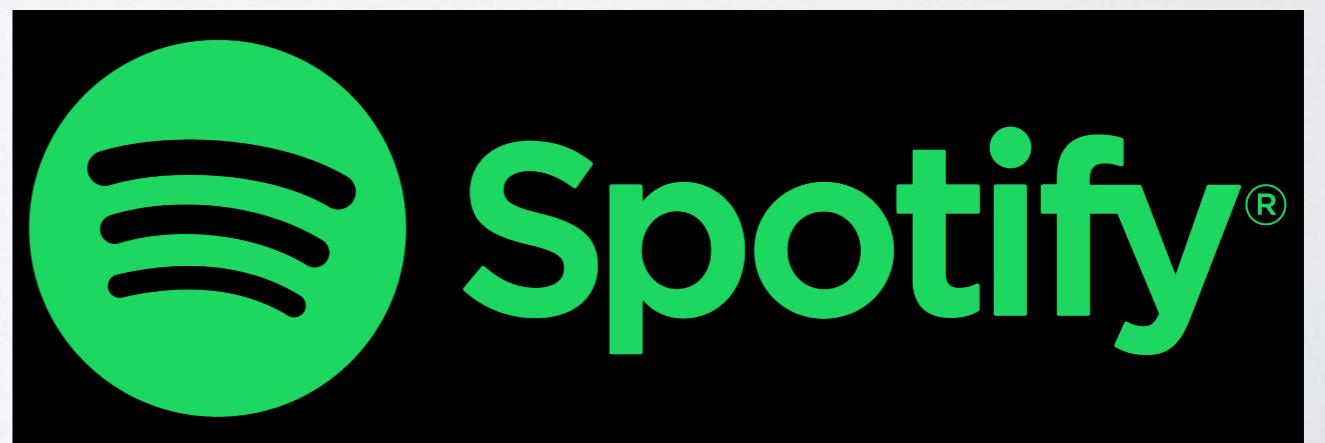
Primitive types may correspond to domain concepts:

- social security number (int) — has validation
- phone number (int) + persons name (string) — together represent concept
- promotional price (float) — has other attributes: start and end date
- payment (float) — has a unit, the currency

Group work

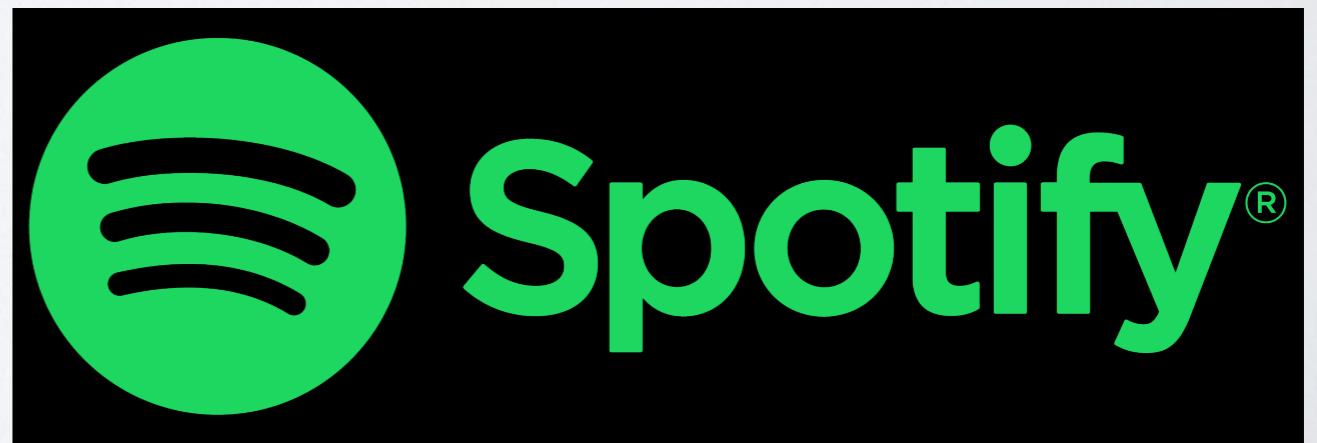
Refactoring

LET'S DO IT



Group work

HAND-IN DESIGN & BLACKBOARD REVIEW

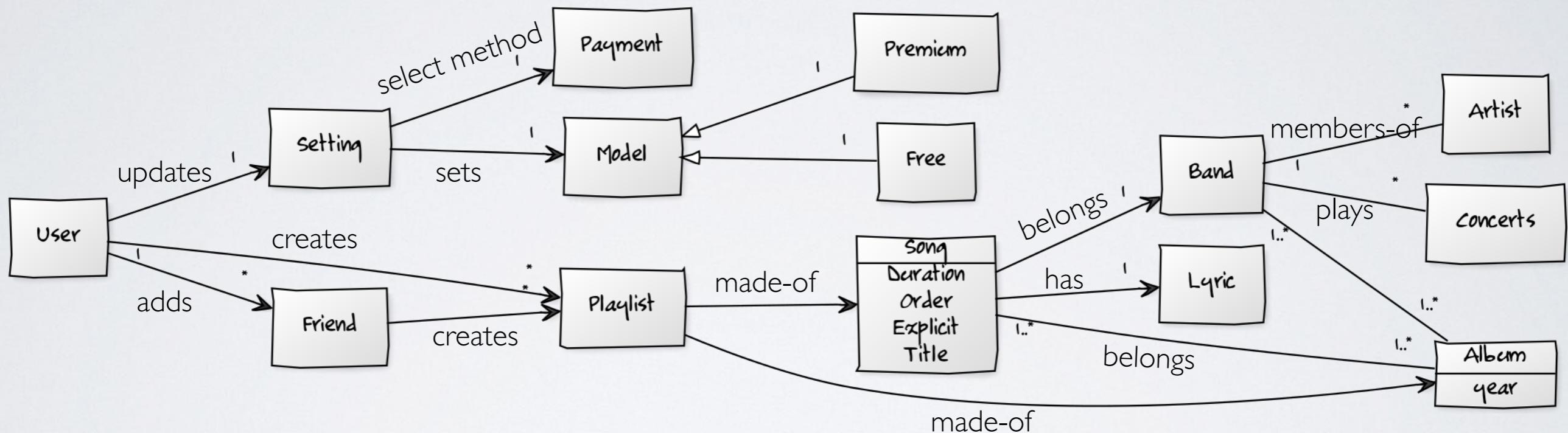


Pay
attention

ALTERNATIVE SOLUTION



MODELLING SPOTIFY



EXTRA EXAMPLE

EXAMPLE

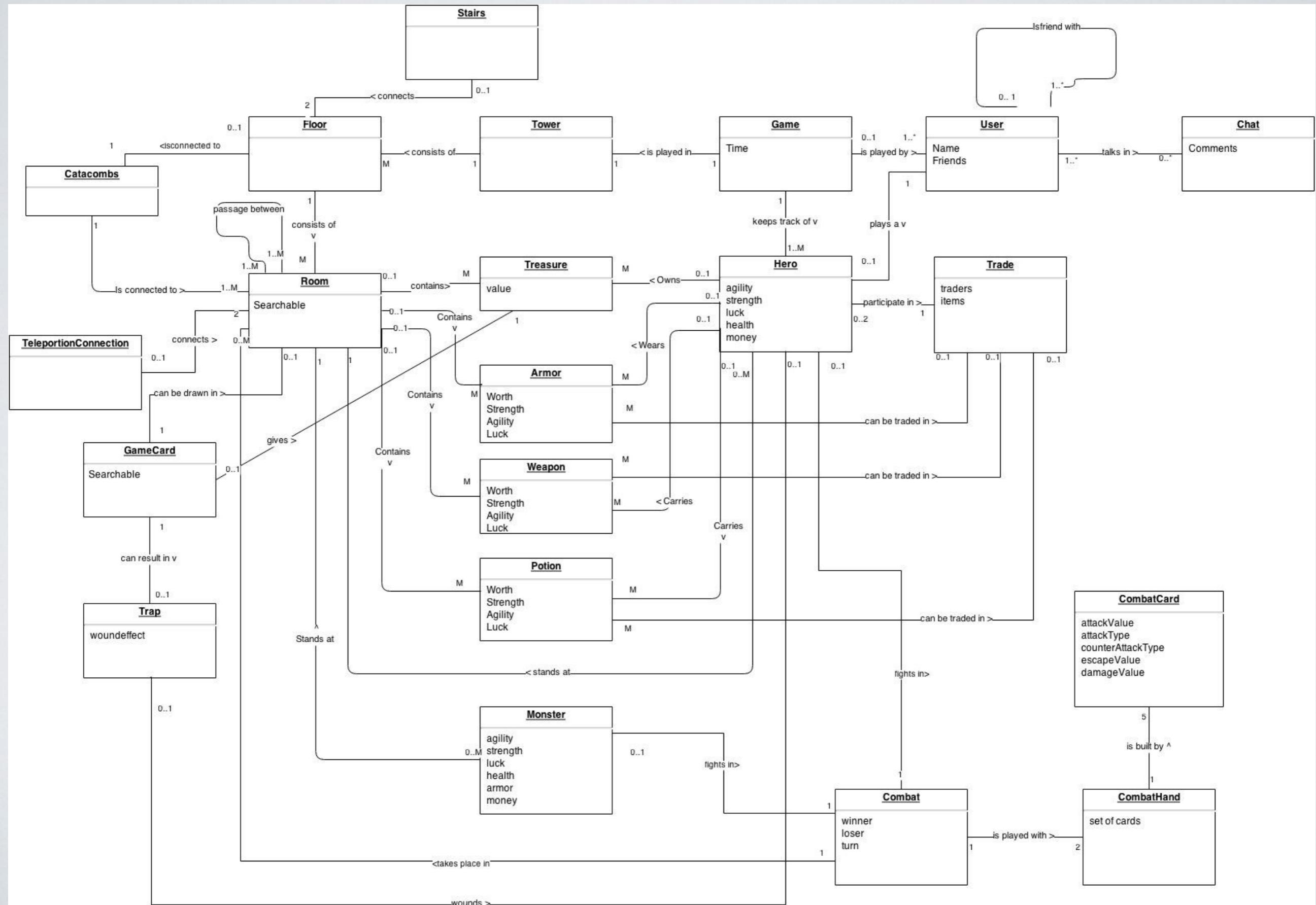


DUNGEONQUEST

- 3D Dungeon with rooms, stairs going up and down
- Multiple players, each player chooses a hero
- Monsters move in the game from room to room
- Players take turns to move
- If you go to a room with a monster, you enter in a combat
 - Draw cards from the combat deck
- Rooms may have traps in it (Trap deck)

LET'S DO IT

POSSIBLE OUTCOME (M = *)



10 - 15 min

CURRENT DESIGN

Work on teams to create a model
for MatchSystem

We will review one or two designs

CONCLUSION

- Domain modelling for:
 - better **understanding of domain** dealing with
 - better **communication** with client
- UML Guidelines:
 - Use domain elements (nouns, physical items, etc)
 - Figure out the relations
 - Add multiplicity