

UML - Unified Modeling Language (1997 -)

"Model-based software design & development"

"Models are built and analyzed prior to the implementation of the system"

"Model-driven architecture"

Standard language, notation

methodology independent, platform independent

Use UML notation for

- software requirements

- analysis

- design models

- use case modeling

- static modeling

- dynamic modeling

Diagrams :

Use case *

Class *

Object *

Communication

Sequence *

State Machine *

Activity

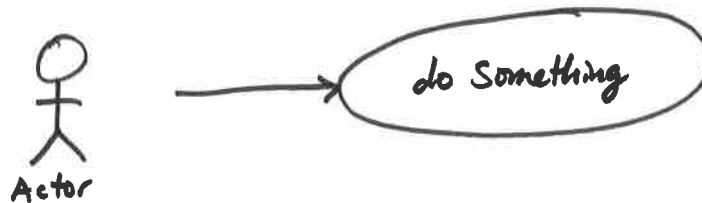
Composite Structure

Deployment *

Use Case Diagrams

Actor initiates a use case.

Defines a sequence of interactions between the actor and the system, to accomplish a goal.



Name: Use case phrase

Primary Actor:

Scope: What's in, what's out

Level: importance

Brief: 2-6 sentence description of use case behavior. Most significant activity, behavior

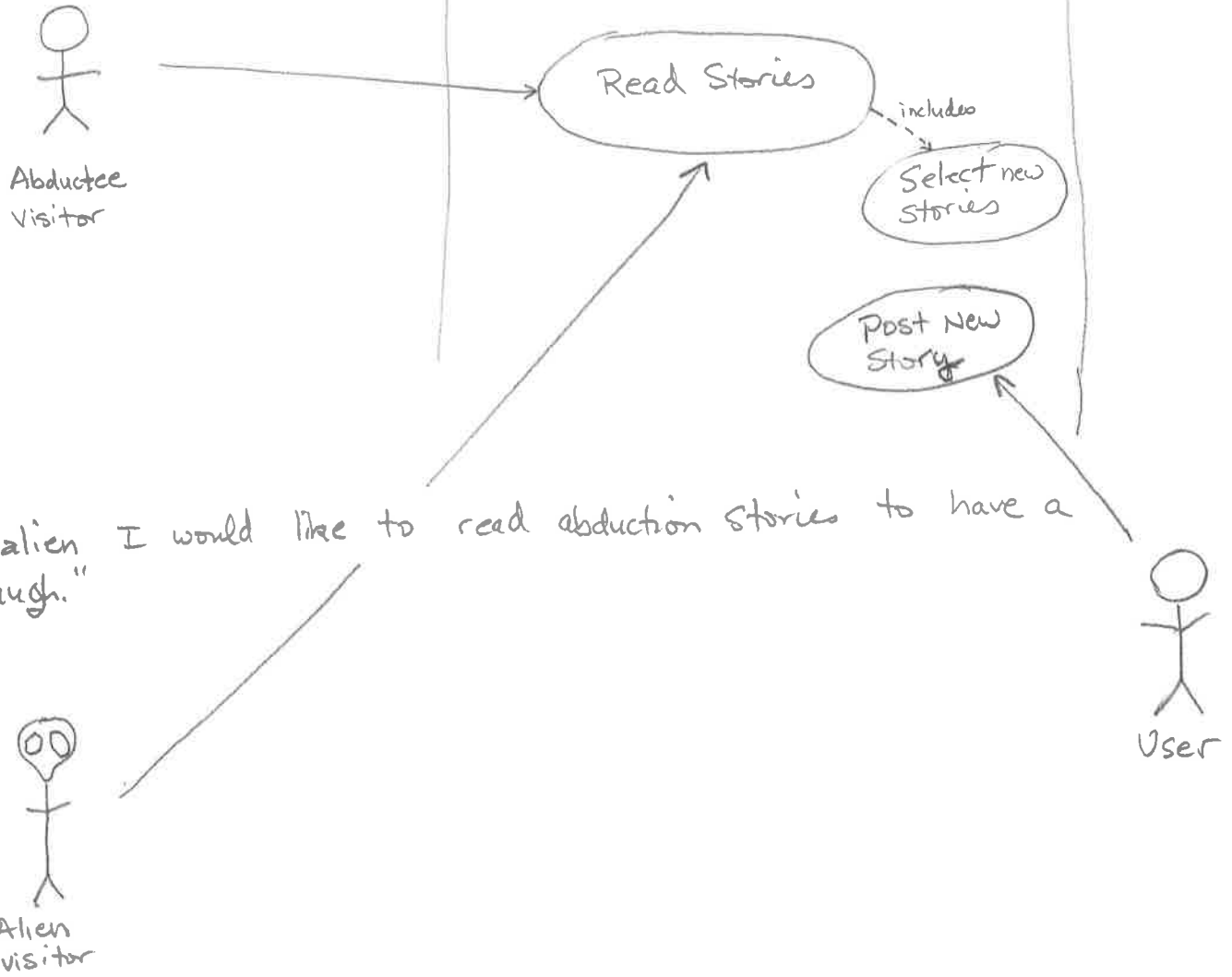
Preconditions:

Post Conditions:

Triggers:

Basic flow:

"As someone who has recently been abducted by aliens I would like to read abduction stories on AlienAbductee.com so I can learn to cope with my experience"



Title: Read stories about abductions

Primary Actor: Any visitor, including aliens

Scope: Primary story viewing system. All public stories, all users/visitors.

Level: ++ (very high)

Brief:	<u>Actor</u>	<u>Goal</u>
	Any visitor abductee alien	Read stories

Brief
Read any public story. Actor should see newest stories automatically. They should be able to click on story link to be able to read the entire story.

Brief cont

Actor should be able to filter stories to read by date, rating, category



Preconditions : None. Any user, any time.

Does NOT need to be a registered user or logged in.

Triggers : Simply viewing the page should bring up the list of newest stories.

Basic Flow : Starting page shows newest stories. Only a title or summary is shown. As many as will fit on one screen. Older items are available on other pages or by scrolling down (YES/NO?).

Actor may click on / select any one story and be taken to a page where they may read the entire story. Going back in the browser takes them back to exactly where they were before.

Actor may filter stories by date, rating or category.

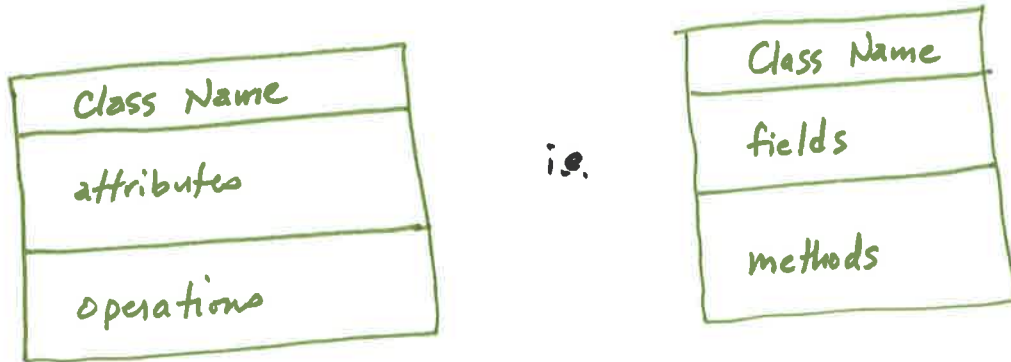
date : ability to select date range

rating : " " " 4 stars, 3 stars, ...

category : " " " any of the available categories

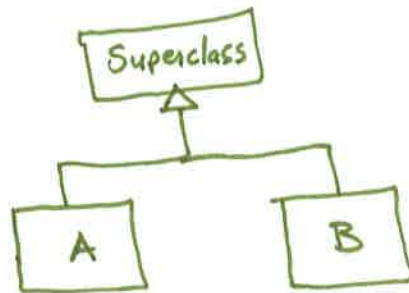
Class Diagrams

What structure is good for supporting/enabling the functionality described in the use cases?

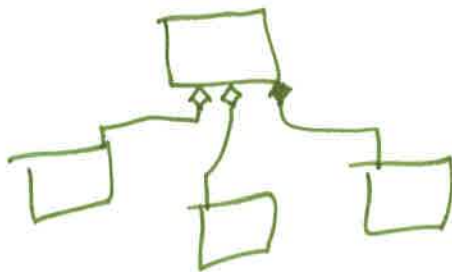


Associations :

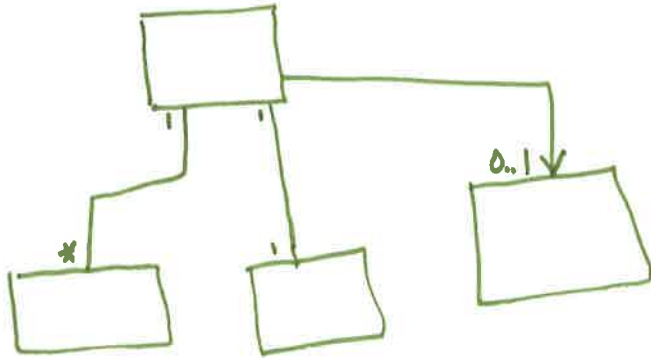
Generalization / Specialization (inheritance) "is-a"



Aggregation / Composition : "has-a" and "must-have-a"



Associations : "uses", "depends on", includes return values from methods



exactly 1

0..1

zero or one (optional)

*

many

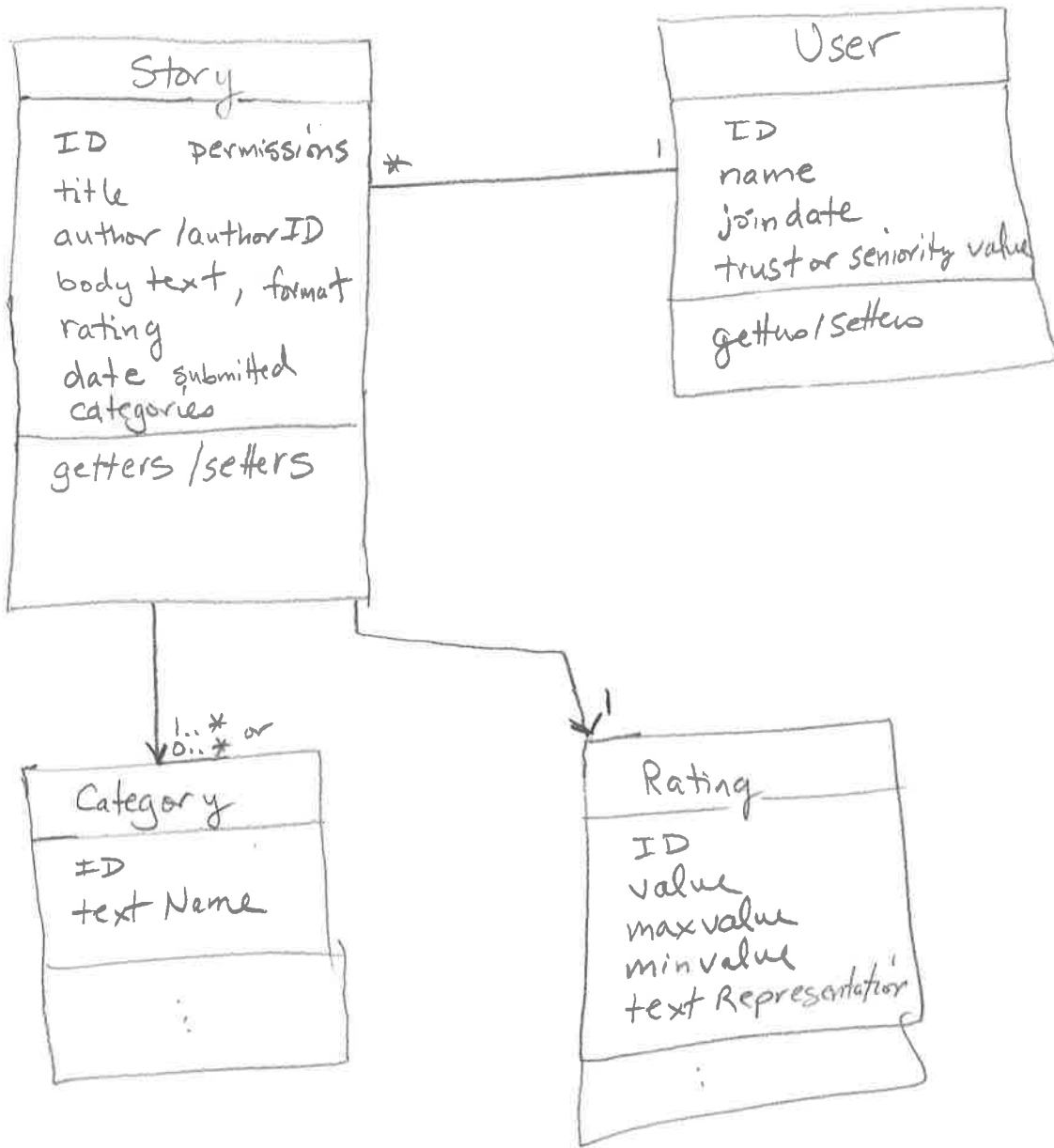
1..*

one or more

m..n

specified range

Class Design :



↗
"model" classes.

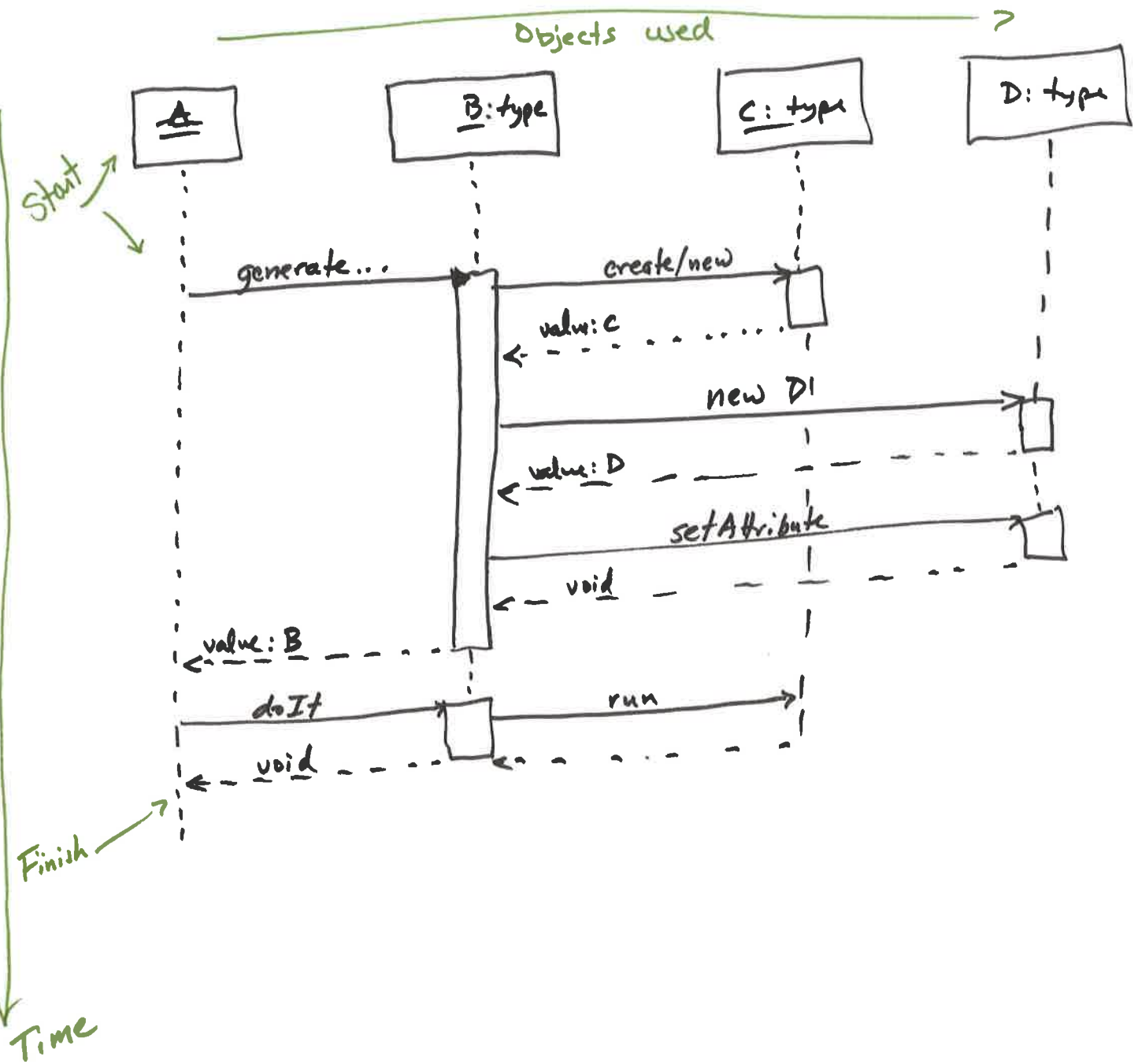
What about the "system"?

Sequence Diagram

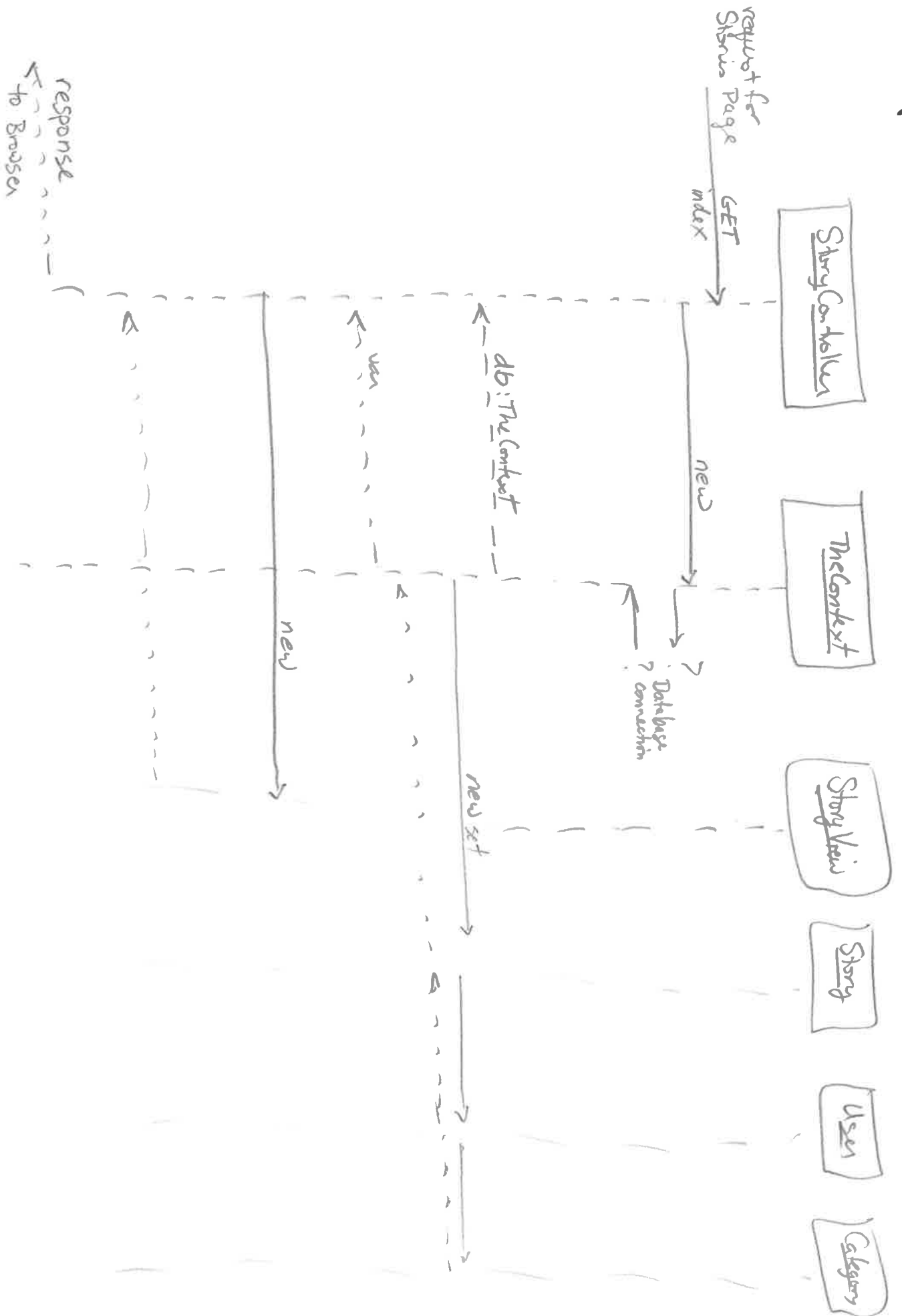
Show interaction among objects over time



object, not class



Sequence Diagram for Our Use Case *



* with my limited knowledge of MVC at the moment