# Control Test - Seminar 2

* **You must read the material in this document and answer questions and tasks with the examples of UML.**

**You must process this document in team (2 students).**Name of student 1: Mykhailo Kryhin

Name of student 2: Vladyslav Odynets

* **This document is submitted by one member of team in the Moodle system (Control Test Submission Form 2).**

## Review Questions

Answer the following questions. Write the answers directly to this document.

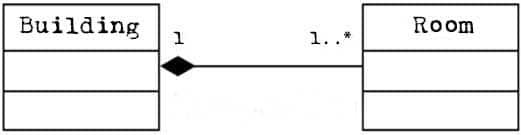
1. *A whole-part* relationship is:
2. Only composition
3. Inheritance
4. Aggregation and composition
5. Class association

Answer: C

1. What is the UML notation for composition?

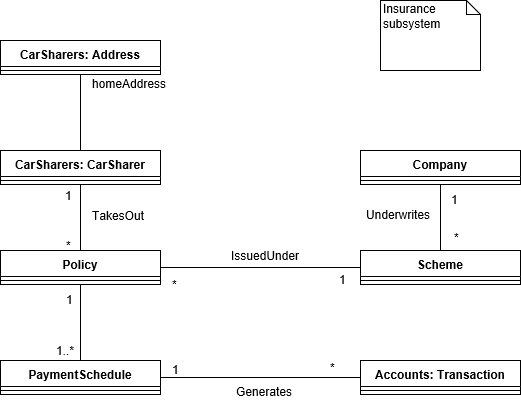
Answer:

Black filled rhombus from the side of “whole” class, for instance, “building *has a* room(s)”



## Solve Problems

The following questions test your understanding of multiplicity notations and have ‘Yes’ or ‘No’ answers. Answer the questions based on the multiplicities shown in the diagram:



1. Can a policy be issued under three schemes?

Answer: No

1. Could we hold the details for a scheme without holding details for a policy issued under that scheme?

Answer: Yes, there could be no policies

1. Does a car sharer have to take out an insurance policy?

Answer: No, he can have 0

1. Could a car sharer take out more than one policy?

Answer: Yes

1. Could we hold details of a policy that has not been taken out by a car sharer?

Answer: No

## Project tasks

The following tasks are the steps for building your class diagram in the project:

1. **First, formulate a problem of your project (purpose of your information system).**

Answer (short paragraph with 4-5 sentences):

“Question-answer system for developers”

Our information system is a “question-answer” informational system, that is used for generating and sharing knowledge about specific topics. The main usage of the system is to allow sharing a knowledge about programming. Also, it will allow users to have “upvote-downvote” system for questions, comments, answers in order to filter the most relevant information and socially reward active users.

1. **Identify an objects, respectively classes in your domain problem.**

Answer (list of classes):  
  
Question  
Answer  
Comment

User

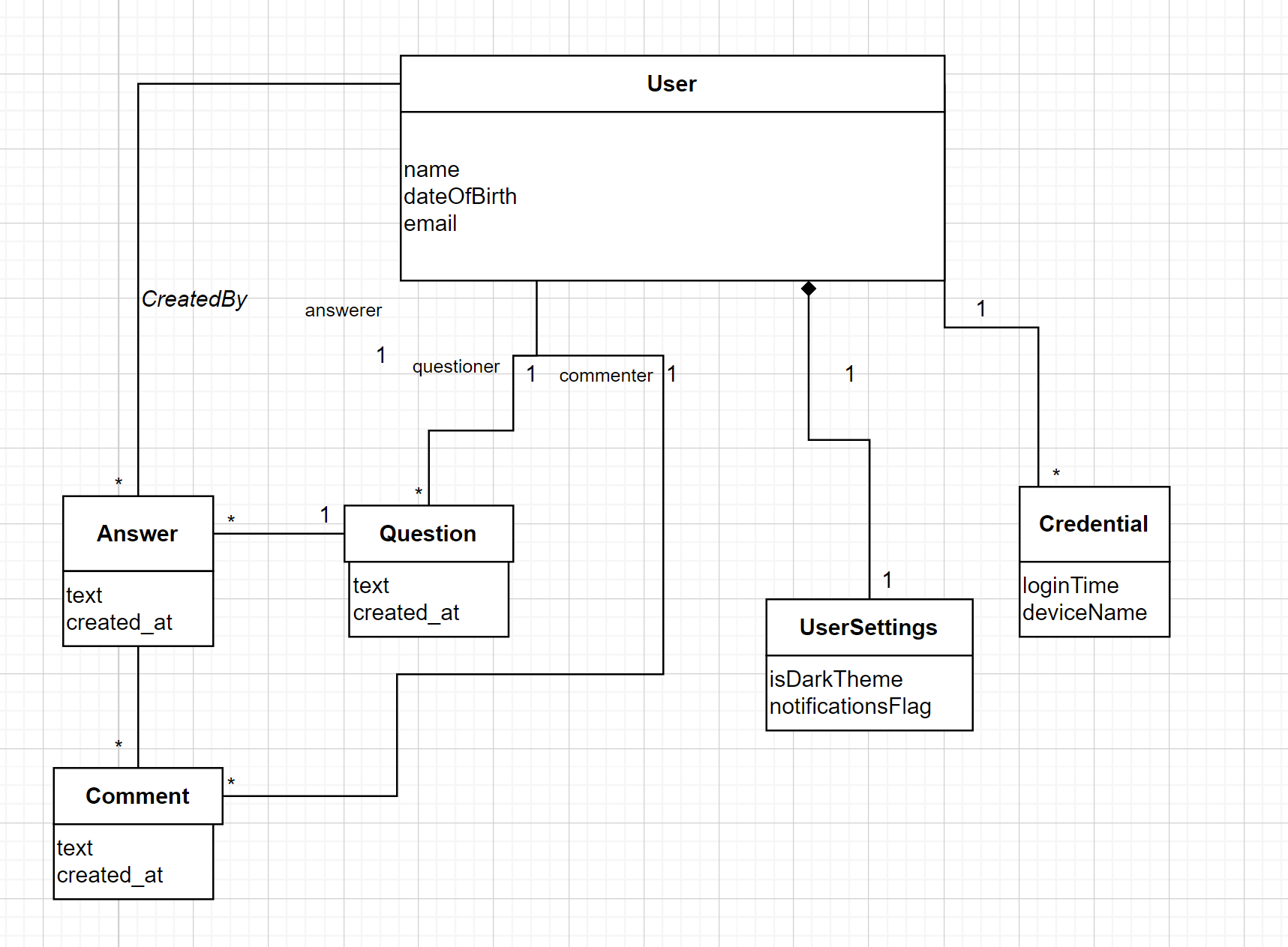
UserSettings  
Credential

1. **Identify an associations between the objects.**

Create first a basic class diagram and put in this document (as figure).

Note: First, you work on paper or with a simple tool (Paint, Draw.io), then you use some CASE tool (MS Visio, MetaEdit+)

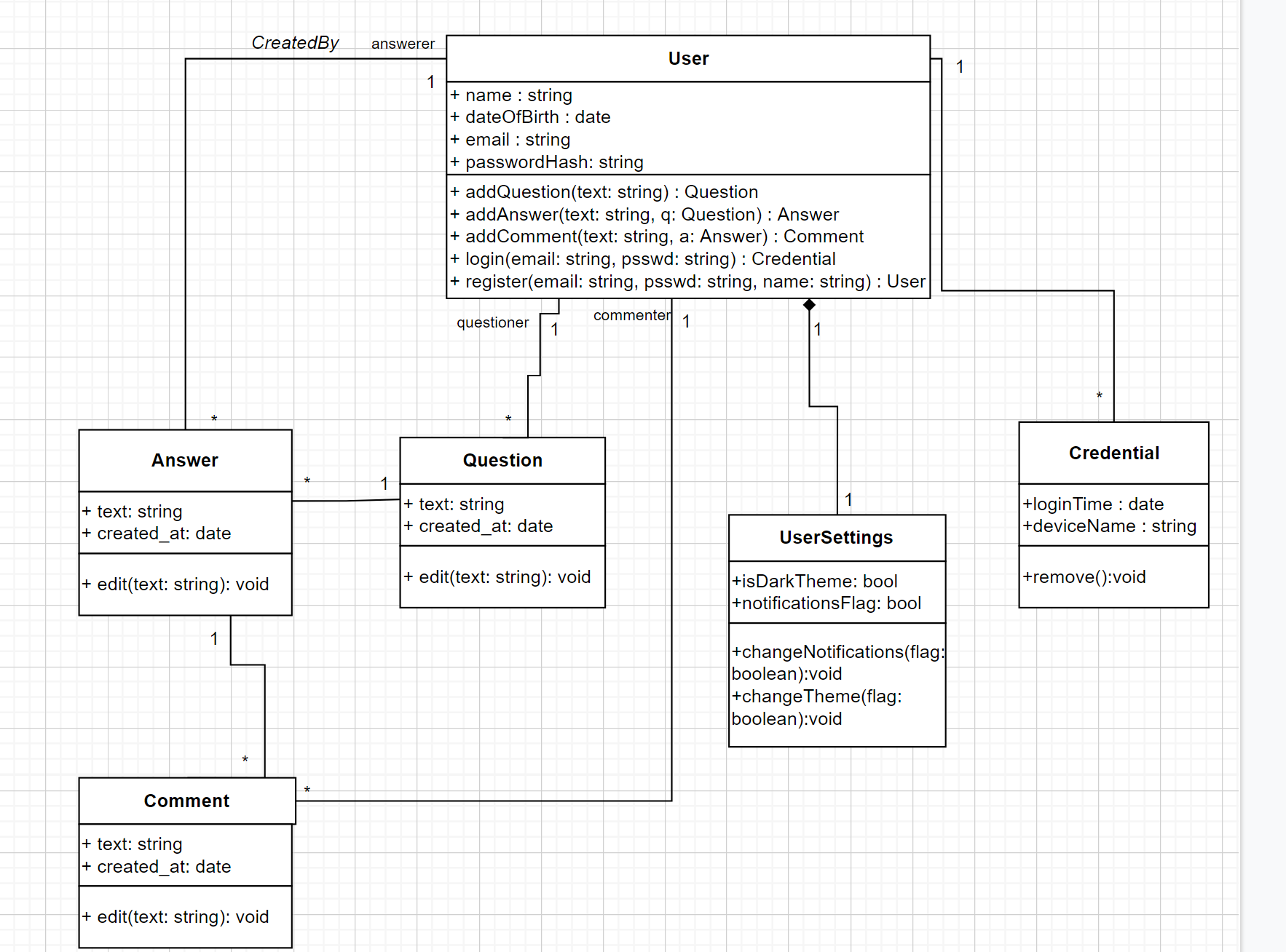
Your class diagram:



1. **Identify object attributes and operations.**

Add attributes and operations into your class model and put in this document (as figure).

Your finite class diagram:



1. **Create Data Dictionary.**

Add description of classes in your model (as paragraph or table).

Your Data Dictionary:

For example:

|  |  |
| --- | --- |
| Class name | Description |
| User | Class that allows objects to use basic methods typical of user registration. Contains methods for logging user to the system. Contains methods for creating question, answer, comment. |
| Question | Class that represent an asked question. Contains method to edit itself. |
| Answer | Class that represent an asked question. Contains method to edit itself. |
| Credential | Class that holds all users sessions. Every device which users logging in from. Contains methods to remove old sessions, checking on fraud etc. |
| UserSettings | Class that allows user to store the user’s settings. Is used to switch dark/light theme, turn off or turn on notifications |
| Comment | Class that represent an comment to an answer. Contains method to edit itself. |
|  |  |
|  |  |
|  |  |
|  |  |

1. **Save the document (in Word or PDF format) and upload to the Moodle.**