Aufgabe 6

Nathan Ritter
5566519
Lorenzo Tecchia
5581906
2023.05.24

Contents

1 Task 1 **4**

2 Task 2 **6**

3 Task 3 **7**

List of Figures

1.1 Solution Domain Class 4
1.2 Solution Domain Class 4

2.1 Activity Diagram 6

3.1 7

Chapter 1

Task 1

(a) Application domain classes are constructed from the domain/requirements engineering in the system analysis, or inception, phase of a development project. They are mostly solution independent in a sense that are not bound to any platform to be developed with(they are not specifically Java classes for example). Do not represent any type of methods, they are ofter depicted with responsibilities. Often application domain classes don't depict any types for their attributes. See figure 1.1.
Unlike application domain classes, solution domain classes are indeed platform dependent and need methods that translates at least partially their responsibility. Compared to the previous type, SDC can often have arrows that imprint directions and therefore more precision when depicting associations between classes. SDC also have to specify types for their attributes. See figure 1.2.

(b) In particular for our project:

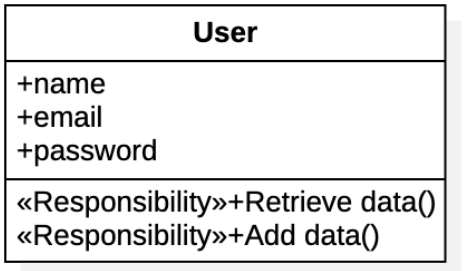

Figure 1.1: Solution Domain Class

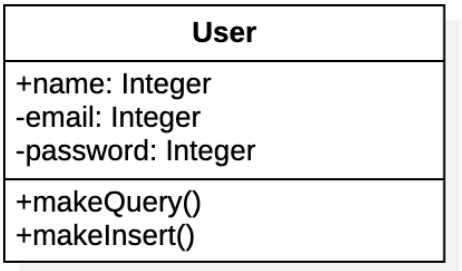

Figure 1.2: Solution Domain Class

(c) This is the table completed:

	Developement Phase		
Aspect	**Analysis**	**Draft**	**Implementation**
Intended Use	idea focusing	idea implementing	idea realizing
Terminology			
Class semantics	application domain	design domain	solution domain
Association semantics			
Detail level	solution independent	platform independent	platform specific
Target group	stakeholders with no technical knowledge	stakeholders with some technical knowledge	developers

Chapter 2

Task 2

(a) Solution:

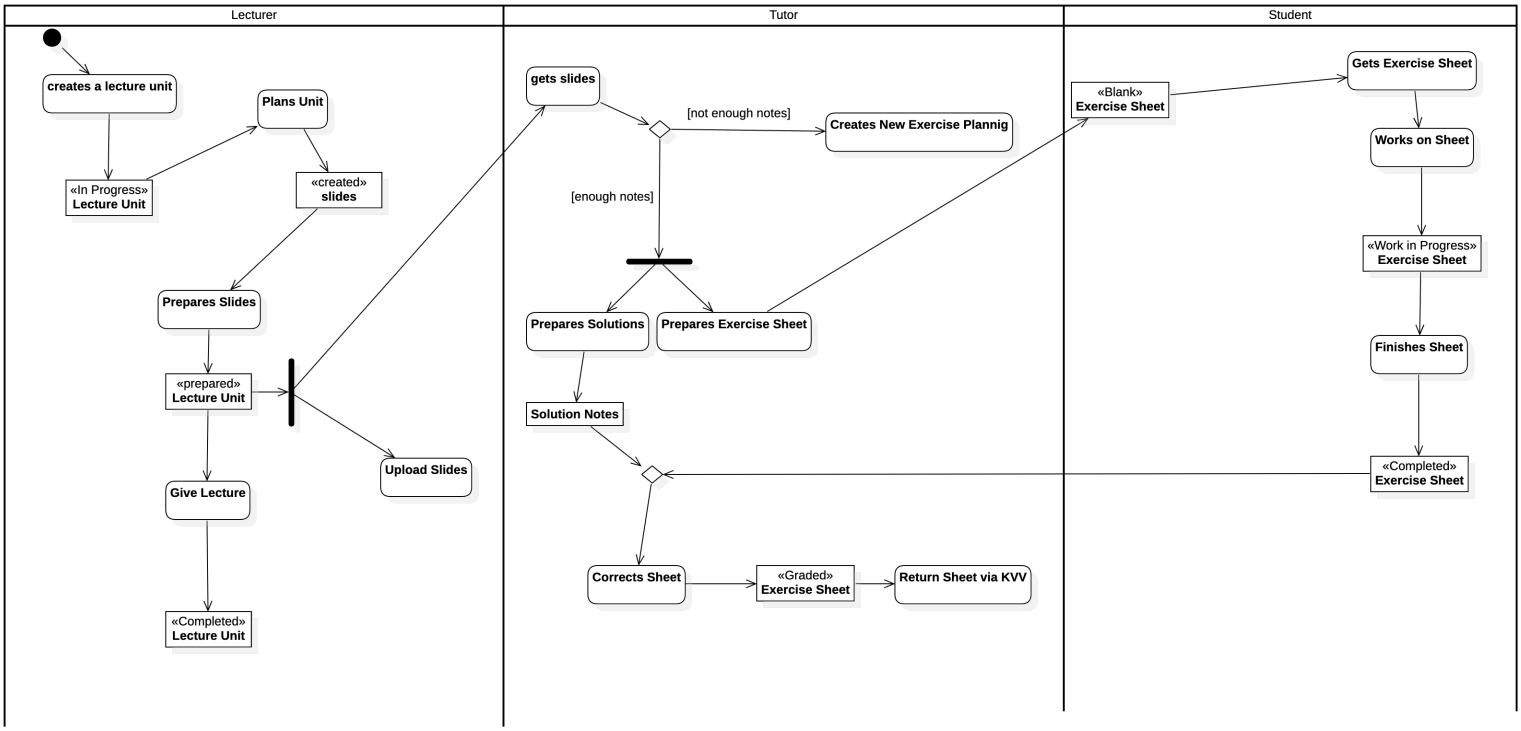

Figure 2.1: Activity Diagram

Chapter 3

Task 3

(a) If actions are bounded to state transitions, their contents would be written onto the transition arrow, meaning that the activity that the arrow is depicting yields a state change. Whereas if the activity is depicted inside the state itself, means that no state change happens and/or the action itself gets activated when the the machine enters the state.

(b) Solution:

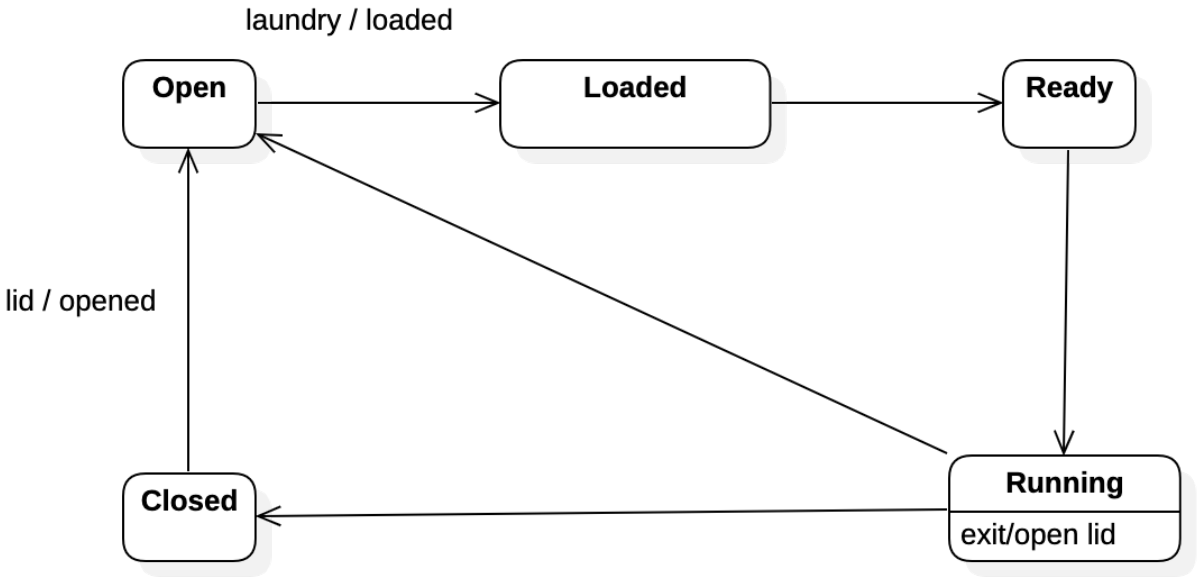

Figure 3.1