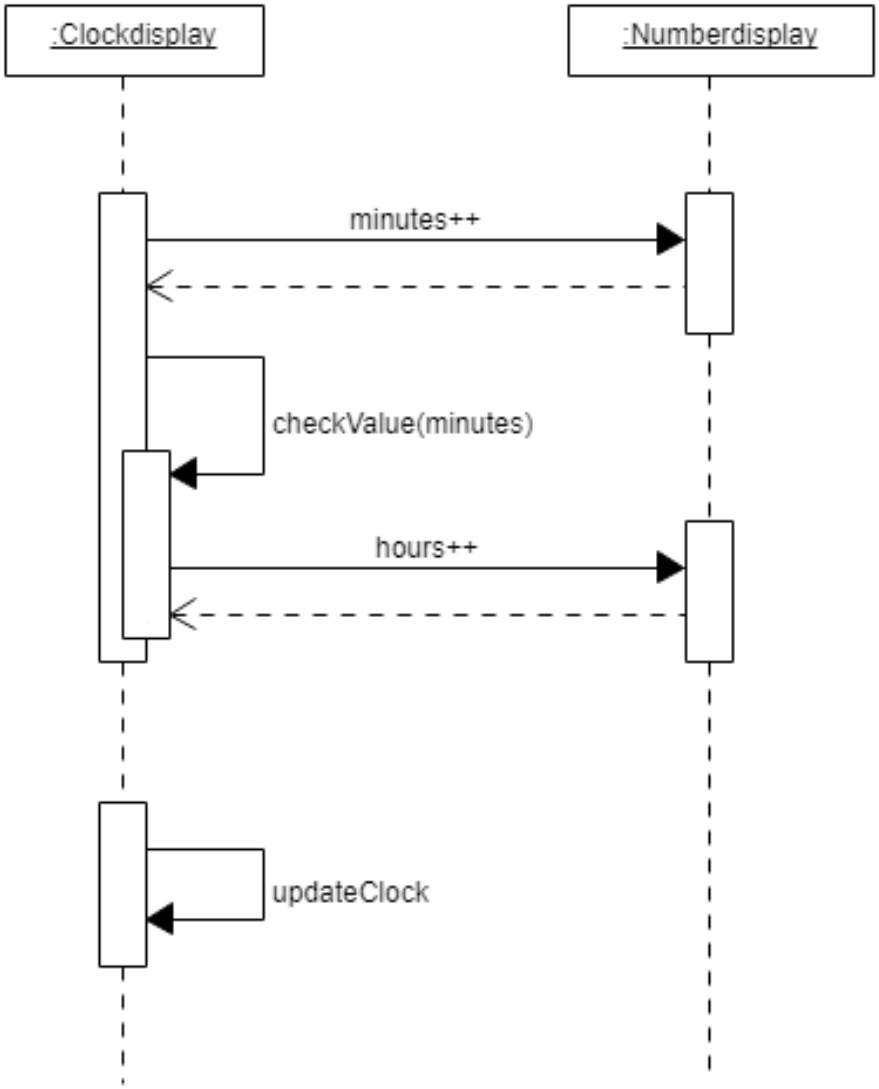
Anh Quang Vu Tuyen 558281
Jakob Lagaly 561455

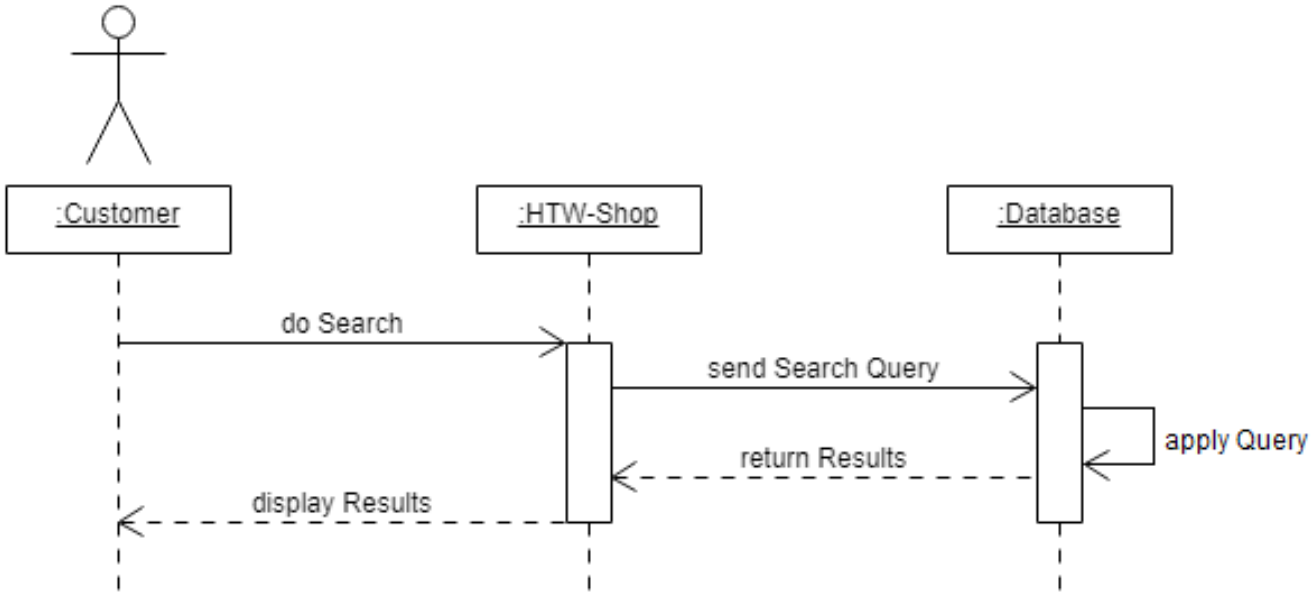
26.11.2017

Lab Report Nr.3

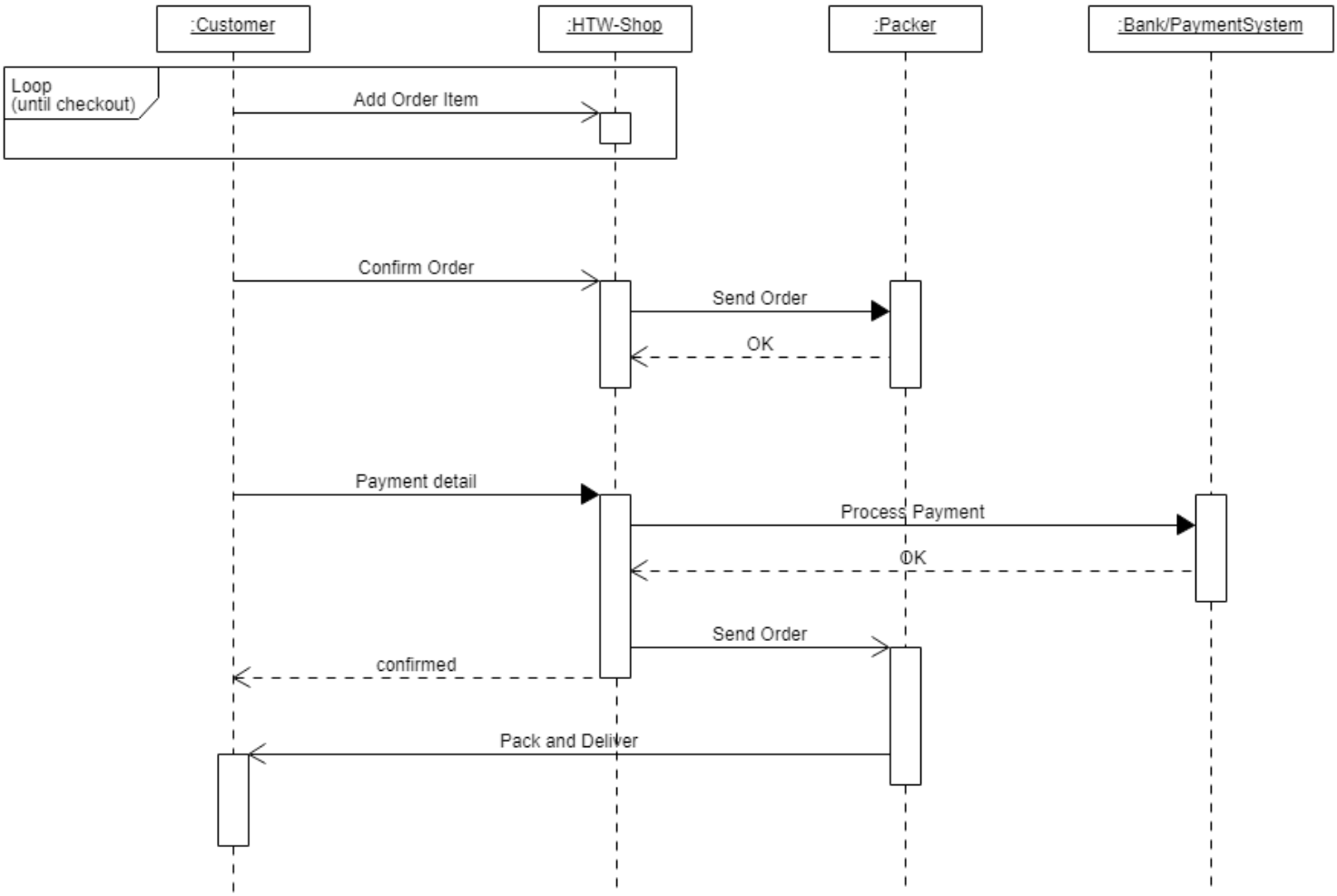
Für uns hat es sich angefühlt, als ob es ziemlich „viele“ Aufgaben waren. Jedoch kannten wir uns schon ein bisschen mit UML und Software-Engineering aus, weswegen wir die Aufgaben recht schnell hinbekommen haben.



Das einzige Problem war eigentlich nur durch den Sourcecode vom Clockdisplay zu gehen und herauszufinden, wie die Methoden miteinander agieren.
Ansonsten haben wir vieles auf ein absolutes minimum gehalten, wie bei dem Sequenzdiagramm HTW-Shop-Suche:

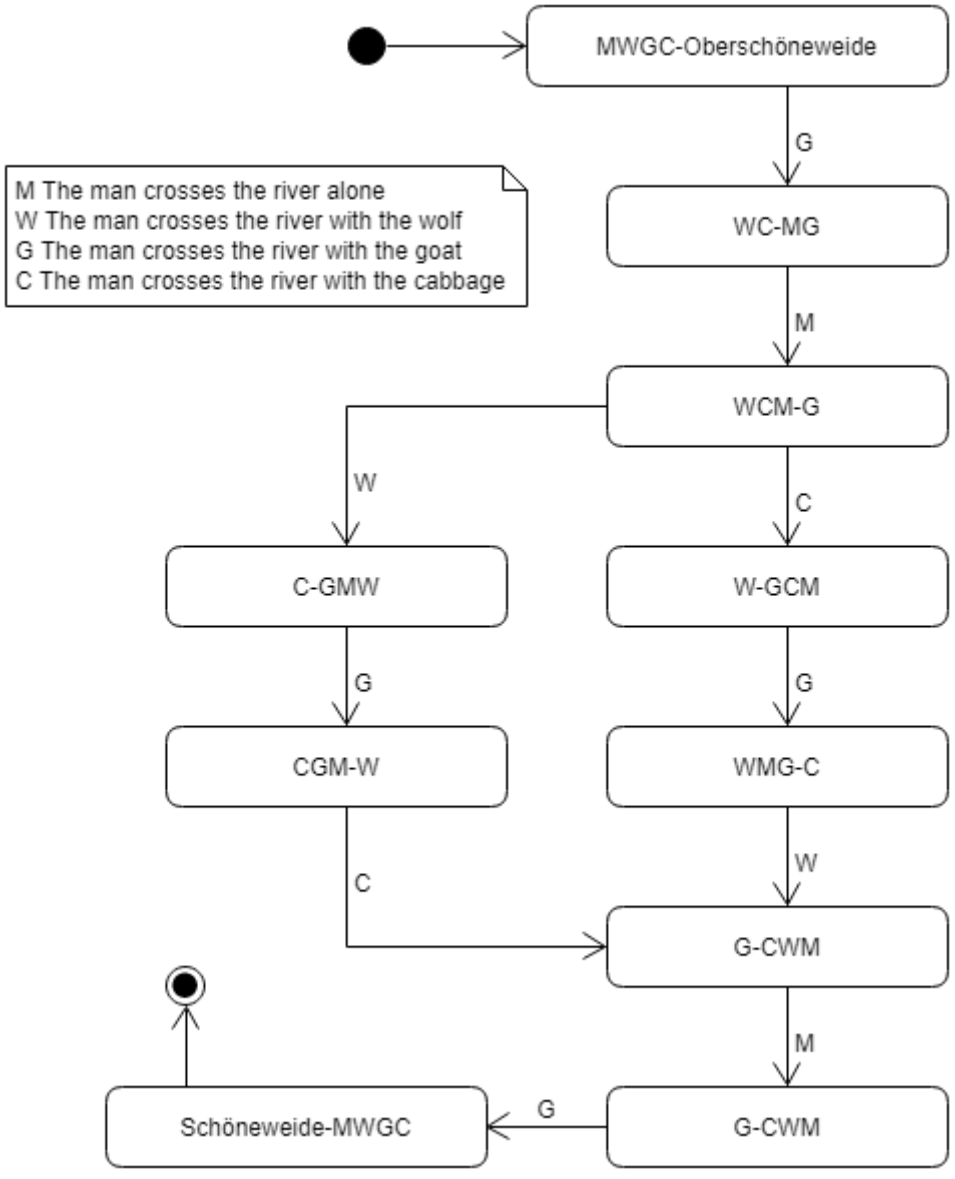


Das am Abstand längste Sequenzdiagramm war vom Htw-Order-System. Dies war jedoch einfach zu modelieren da wir wussten, wie dieses System funktionierte.

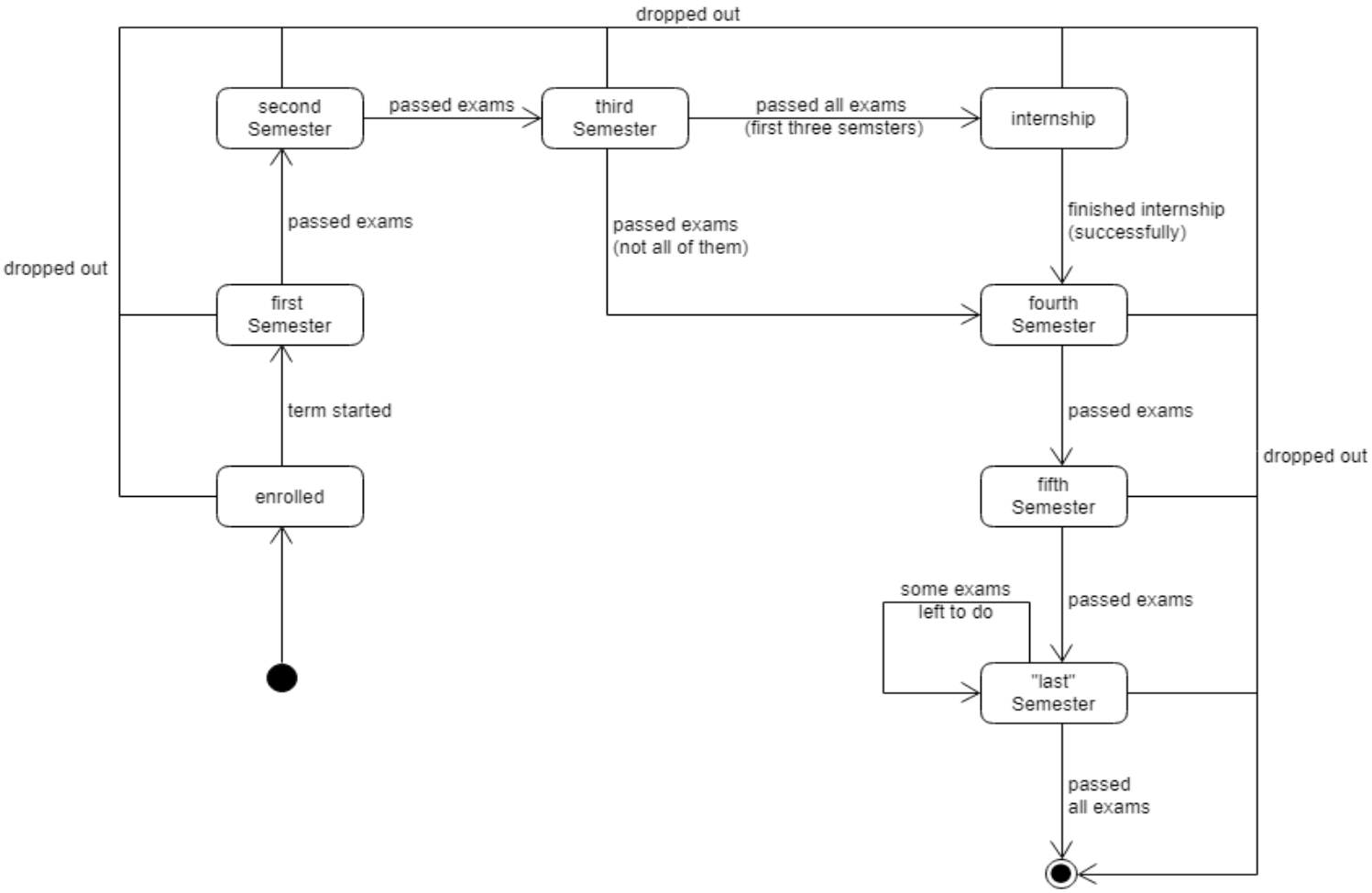


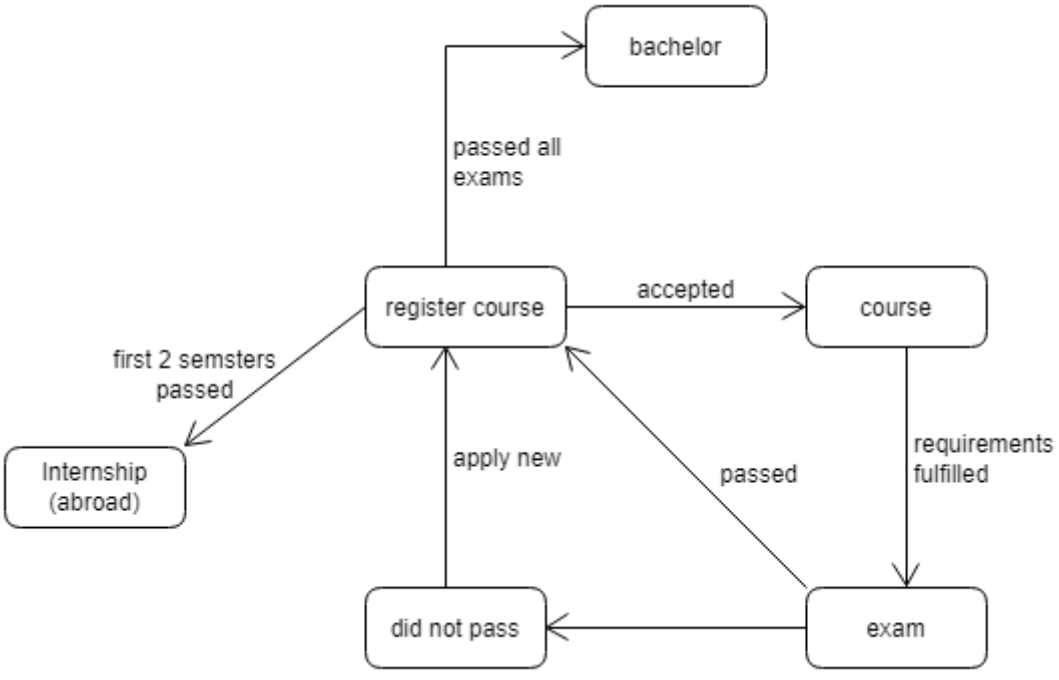
Bei Part 2 war ich(Anh) zunächst verunsichert, da ich dachte wir müssten Sequenz-Diagramme für alle Fälle entwerfen. Jedoch hat Jakob mich darauf hingewiesen, dass es nur State-Machine Diagramme sind, welches den Prozess sehr erleichtert hat.

Im Wolf-Cabbage-Goat Fall haben wir gemerkt, dass es zwar unendlich viele Wege zum Ziel gibt aber nur 2 die keine „Loops“ beinhalten und haben deshalb nur diese modelliert.

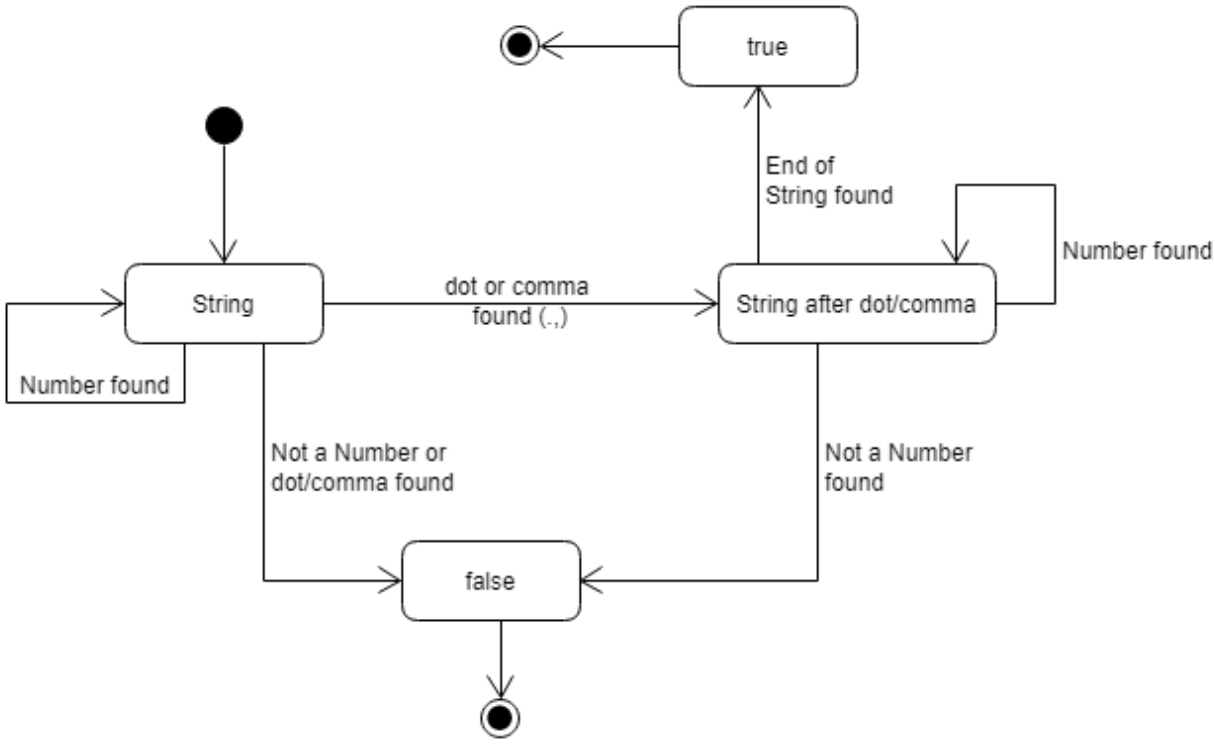


Bei der IMI-Student-Semester Aufgabe haben wir jedoch 2 Varianten gemacht, davon ist eine relativ simpel und die andere etwas absrakter gehalten

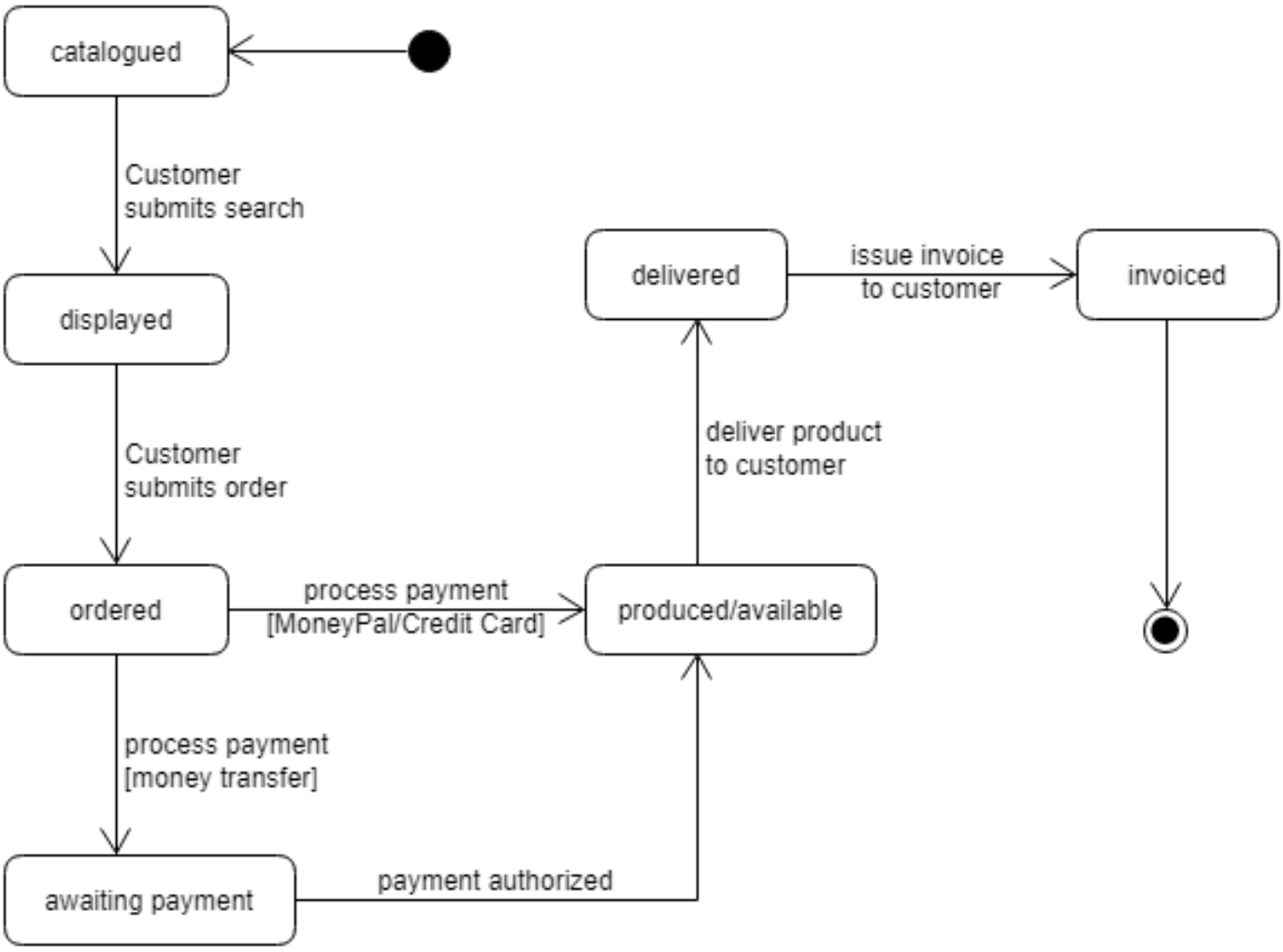




Das String-Floatingpoint-Diagramm war die beste Aufgabe um beim Modelieren viele „Was passiert wenn...?“ Fragen zu stellen (und zu beantworten :-)).



Beim letzten State Machine Diagramm ging es nur darum vorzustellen, was wir vor dem Computer machen und auf was wir warten würden, falls wir was bestellen sollten.



Meistens haben wir, wie schon gesagt, ein Minimum angestrebt. D.h. dass unsere Diagramme noch ausbaufähig sind, jedoch haben sie notwendige Funktionalitäten die den Sachverhalt eindeutig erläutern.