**Uppaal model report Dibran Dokter 1047390 & Marnix Lukasse 1047400**

For this practical assignment we created a solution for the readers writer problem. With this solution, starvation for either readers or writers is avoided. A brief summary of the solution: Whenever a reader or writer wants to perform an action, it must first request the turnStile semaphore. In the initial state of the model, the first time a reader requests this turnStile (semWait), it will be immediately granted to the reader (semGo). The reader will then immediately release the semaphore again (semSignal). This might seem pointless on first sight, why does a reader request this turnStile if it is going to immediately release it again? Well this turnStile semaphore actually plays a key role in preventing starvation from happening. To understand this, consider a writer. A writer also first request the turnStile semaphore, but releases it only after completing a writing action. As turnStile is a binary semaphore, this means that no 2 writers can ever be writing simultaneously. Because the semaphore works with queue mechanism, it prevents starvation. Once a reader or writer requests turnStile, it will be added in the queue. This means that both readers and writers will get the turnStile access eventually no matter what.