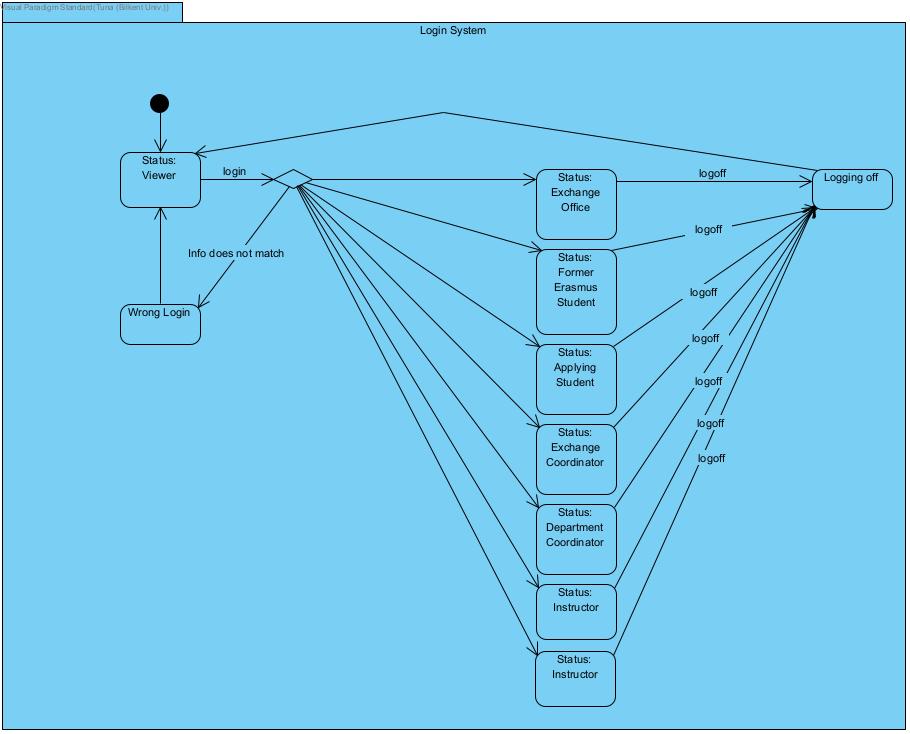
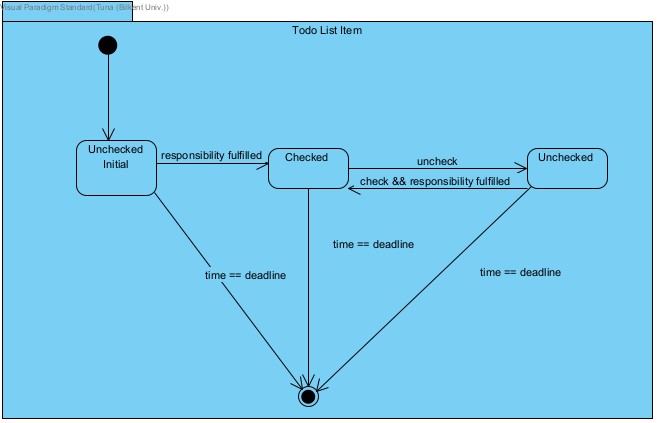
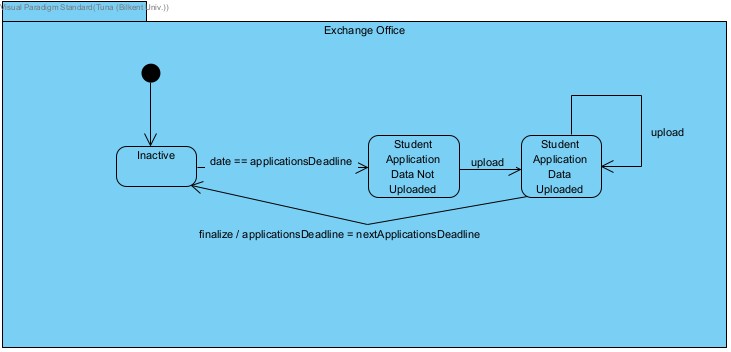
The viewer can attempt to log in, in which case the system checks their information against the information of the users in the database. If it doesn’t match any user, their viewer status is unchanged. Else, depending on which user the information inputted matches, the viewer is matched with a role and is provided appropriate privileges. (Note that this matching process is unrepresented in the diagram for brevity.) Then, the user can log off whenever they’d like.



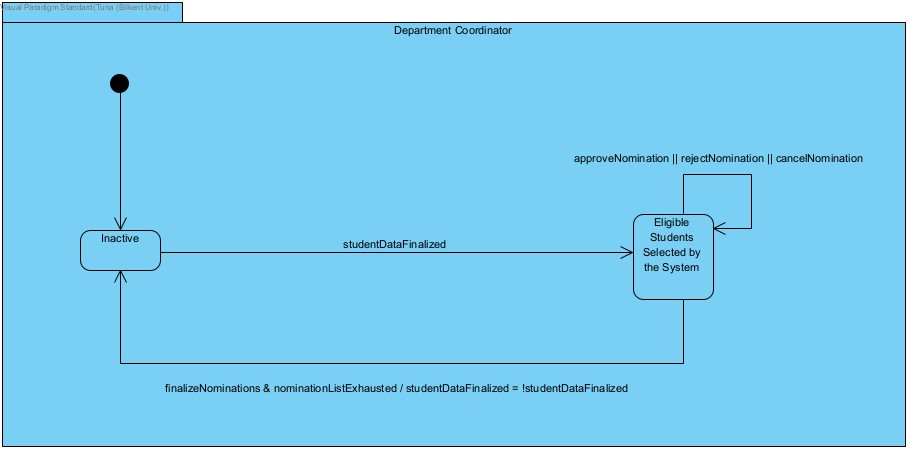
Each to-do list item included under a to-do list is generated by the system: there is no way for a user to create one of their own. Each item starts out unchecked and is automatically checked whenever the system detects that the responsibility the item corresponds to has been fulfilled(ex. course preapproval form uploaded). Then, if the user feels like they should redo this part, they can uncheck the item themselves after which they will be able to check it back themselves if the responsibility is fulfilled. (So initially the item is unchecked, which is automatically checked when the course pre-approval form is uploaded. If the user realizes they made a mistake, they can then uncheck the item themselves, after which they can immediately check it back. However, if they also retract the form after unchecking the to-do list item, the system won’t allow them to check the item again so as to prevent a mistake. When they upload it again, they will be able to check the list themselves.) Once the deadline for the responsibility is reached, the item terminates and is removed from the list.

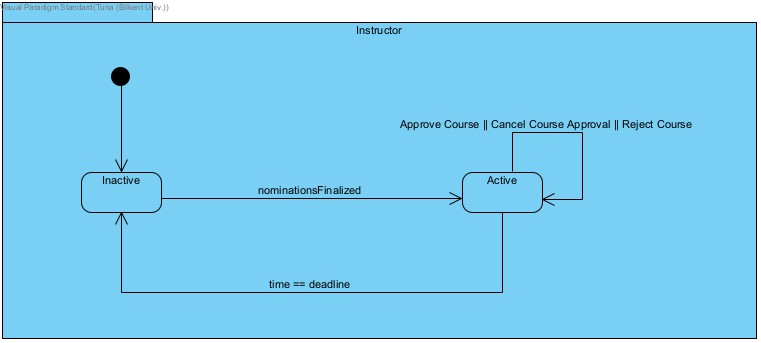


Exchange office is initialized as inactive(no responsibilities), and when the date is equal to the application deadline, i.e. the applications have been collected, it becomes active. It can then upload the excel file with the student application data as many times as it wants(in case of erroneous uploads). When it finalizes the upload, it again becomes inactive, and the application deadline is set to the next application deadline.



Once the student data is finalized by the exchange office, the students are ranked by the system and those to be nominated are promptly chosen. Once this happens, the department coordinator becomes active and can either approve or reject nominations or cancel a nomination choice so that it may be remade. When the nomination list is exhausted(i.e. every student nominated is either approved or rejected), nominations can be finalized, at which point the state machine becomes inactive again.

Note that as in any other state diagram representing a role, the basic capabilities of this role are not included in this diagram, as they do not advance the state machine to a new state. For example, a department coordinator can send DMs and use the forums, but as these actions do not change its state, they are not represented here. Note also that these actions can be used even in the inactive state.

Once the nominations are finalized, the students can propose courses. As proposals arrive, instructor can either approve or reject course proposals or cancel an approval choice so that it may be remade. Once the deadline arrives, the instructor becomes inactive again.

If a student is already in the system as a former Erasmus student, they stay as one. If the excel file uploaded includes this student, they are now in the system as an applicant(Did Apply). Everyone else is a non-applicant. If a student goes through the entire Erasmus application process successfully, they become a former Erasmus student and stay as one.

