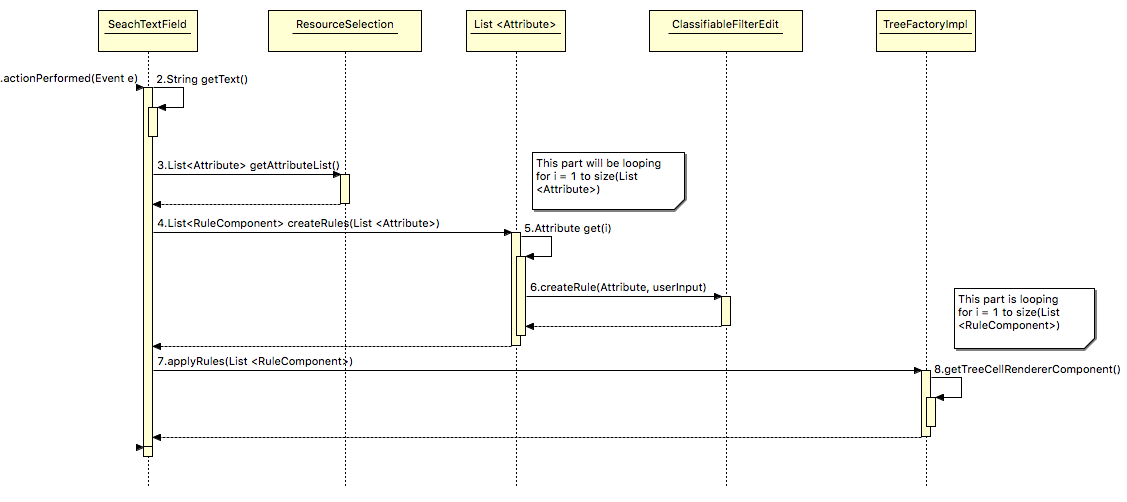
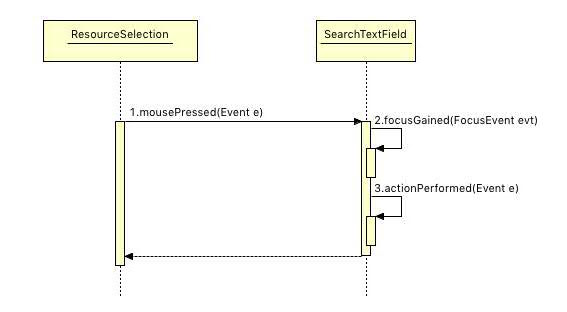
1.   
     
   Currently in Rapla, the user has the option to filter events by a certain resource associated with events; for example, if I choose to filter “people” by “first name” the program *should* only display events with that criteria. Although, it doesn’t quite do that – if a filter isn’t filled in for every other resource, other events will be mistakenly displayed. In this case, the user has to inconveniently apply filters for all the resources.  
     
   Therefore, our project enhancement is to speed up this process. Instead of requiring our user to individually choose each filter option, we have created a text field where a user will type in filter criteria. At that point, the search criteria will act as an OR function – it will set a filter for every single attribute with the input criteria. This makes our enhancement more user friendly, and more effective.  
     
   Onto the sequence diagram – since our enhancement is essentially “replacing” the already existing filter system, most of code already exists that dictates *how* to filter, we just need to tweak it to our liking so that the process is more efficient. That would explain why our sequence diagram isn’t exceptionally large, because there isn’t much we have to add. Additionally, the implementation might not be entirely correct at the moment because we are still working on our enhancement – but the concept is pretty concrete.
2. Our enhancement interacts with the original code in many instances. One such example would be when we call the method applyRulesList. By the time we get to this method, we would have iterated through every attribute that exists in that instance of Rapla, and created a rule applicable to every single one of those attributes.   
     
   This method will pass those rules onto the TreeFactoryImpl class, and perform the pre-existing method getTreeCellRendererComponent. In the already existing filter feature, this is the process that is followed once you type in filter criteria in the pop-up text field. We want to work our enhancement into this process, as its shown to already cooperate with text fields properly when given rules to implement in Rapla.
3. The way that the user interacts with the enhancement is pretty simple. This is just a faster way that the user can use to search/filter events on the screen, so all they need to do is click in the on screen text field and begin typing in search criteria.  
     
     
     
   Note that this diagram simply displays the interaction where we click on the text field (located in the resource selection menu) and the focus is gained within a specific instance of search text field in that menu, then calls on actionPerformed (if you notice from the previous sequence diagram, that’s the method that performs the search filter actions.)