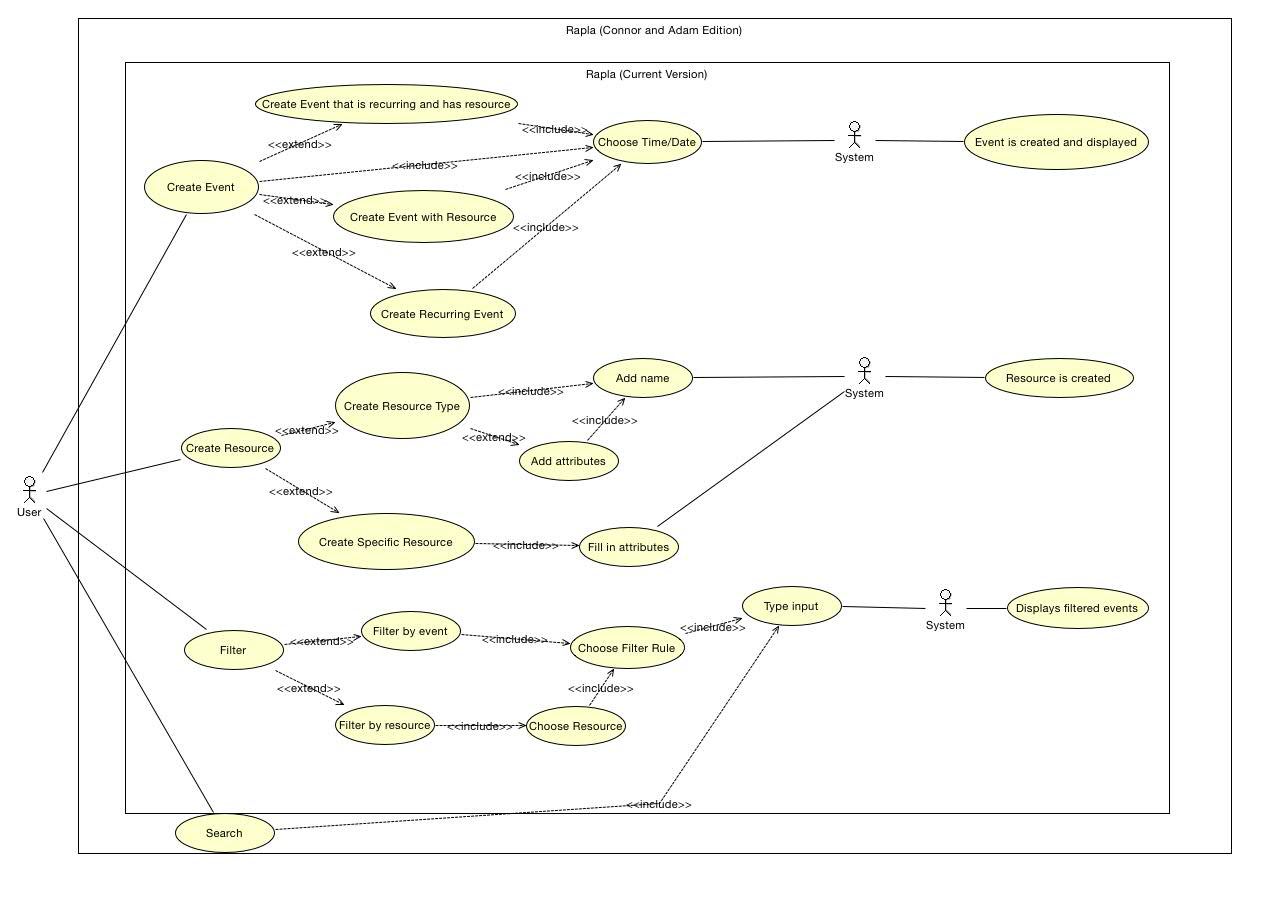
1. For our software project, we have identified 3 major use cases, and multiple actions that are included in the use cases.  
     
   a) Create Event: The user can create an event, which will be displayed in the Rapla calendar organizer.   
     
   Extended from this use case are three specialized use cases: the user can add a resource to the event, they can make the event recurring, or they can apply both to an event.  
     
   Whether the user adds a resource, makes an event recurring, both or none, included in this process is the necessity to select a time and date for the event when it occurs.  
     
   Then it is sent to the system, where it is created and displayed in the interface.  
     
   b) Create Resource: The user can create a resource, which can be attached to an event to help describe its details. For example, a resource may be a person, location, etc.  
     
   This use case can be specialized in two ways. We can either create a resource type (location, person, operating system) or a resource instance (more concrete instance of those categories like Laurel Hall, Adam Smith, Windows 10).  
     
   If we are creating a resource type, we can specialize it to have more than one attribute besides the name. For example, a location could be called Laurel Hall, and we can add an attribute “Room number” and “Floor Number” but we don’t HAVE to. Whether or not we include extra attributes, a name for the type must be included. Then, it is sent to the system to create the resource type.  
     
   If we are creating a resource instance, all we have to do is fill in the attribute values. For example, for a location, if it asks for “Name”, “Room Number”, and “Floor Number” we should provide all those details. Once we do, the system will create that resource instance.  
     
   c) Filter: Allows the user to filter out certain events by provided criteria. For example, a user can choose to filter displayed events to only show events containing the characters “Lunch.”  
     
   When choosing how to filter, we can specialize it so we filter either by event or by resources.  
     
   If we are filtering by event, we just have to specify the filter rule (contains/starts with). If we are filtering by resource, we have to choose the resource we wish to filter and then specify the rule (contains/starts with).  
     
   Then, the user will input their filter criteria, such as “Lunch.” The system will take this input and only display events containing that relevant criteria.
2. Our enhancement will include the fast resource search feature. This is considered more of an add-on than a replacement.  
     
   We will keep all the previous use cases in our implementation. As seen in the use case diagram, our search feature will fall into the filter sub-tree, but skipping most of the use case steps. We will be making our search feature simply so that the user will type in their search criteria for a resource, and it will be filtered out by the system and display only relevant events. This effectively makes the resource search “fast” as it skips 3 steps.
3.   
   Use Case Diagram. If this is not clear enough, the .jpg & .cld will be uploaded to Husky CT. They will also reside in our Github Repository