**Sequence Diagram rules**

**Entities** *(model)* Objects representing system data, often from the domain model.

**Boundaries** *(view)* Objects that interface with system actors (e.g. a user or external service). Windows, screens and menus are examples of boundaries that interface with users.

**Controls** *(controller)* Objects that mediate between boundaries and entities. These serve as the glue between boundary elements and entity elements, implementing the logic required to manage the various elements and their interactions. It is important to understand that you may decide to implement controllers within your design as something other than objects – many controllers are simple enough to be implemented as a method of an entity or boundary class for example.

**Four rules apply to their communication:**

1. Actors can only talk to boundary objects.
2. Boundary objects can only talk to controllers and actors.
3. Entity objects can only talk to controllers.
4. Controllers can talk to boundary objects and entity objects, and to other controllers, but not to actors

**Communication allowed:**

Entity Boundary Control Actor

Entity Y - X -

Actor - Y - -

Boundary - - Y Y

Control Y Y Y -

actor : person

boundary : rest resource

control : app controller service

entity :