In this assignment, I chose to use the combination of InVivo coding and open coding as my coding strategy in this assignment because InVivo coding can preserve the original meaning in codes, which is critical for coders who are not very professional in cameras.

I found that, unlike literature, technical documents are usually tersely detailed, which means there are more key point stones that are needed to be coded while they are well portrayed with less redundant words, so those documents require delicate skills in coding in order to compress an already precise sentence into a shorter one and to exclude the relatively trivial information, otherwise codes become very hard to be interpreted back in this case. In this assignment, the document introduces components of the camera which can be coded briefly and also the operation procedures which require more specific codes. Thus, I started preliminary coding referring to the examples in background reading materials, afterward, I found that the preliminary coding merely provides me an overview of the document, and it is inadequate to establish effective annotations to construct state machines because the transition between states cannot be reflected.

Preliminary coding document can be found here: <https://github.com/HengZ121/App.-of-NLP-ML-in-SE/blob/master/assignment%201/Preliminary%20Codes.docx>

From final coding, I observed the content of codes and established the following annotation legend which can cover all information my final codes contain:

**state, mechanical action, manual action, feature,** and **component**

State is the camera mode selected via command dial or switches, mechanical action is an action executed by the camera (e.g., auto-focusing, pre-winding film around the take-up spool), manual action is an action took by the photographer (e.g., aiming the camera at the subject), feature is literally a feature that the camera provides, and component is either a unit from the parts of camera or a physical function that requires manual operation on it. Also, for component, it is necessary to include the article and the modifiers (if any) of it in annotation in order to distinguish the negative sentence from positive sentences and to specify special conditions, otherwise the confounding data may be sent to A.I. training model later on; for instances: *there is no click stop when the shutter button is being released*, and *There is a click stop at the halfway point*.

Final coding document can be found here: <https://github.com/HengZ121/App.-of-NLP-ML-in-SE/blob/master/assignment%201/Final%20Codes.docx>

By annotating the camera specification document, I found a logically tricky thing is the way we split an active sentence based on annotation legend: for example, applying an annotation legend that contains “action” and “system component” over the text:

*Pressing the start button will turn the engine on*

we can definitely acknowledge that “*Pressing the start button*” is an action, and the “*start button*” is a system component inside the action, but now let try:

*The start button is pressed to turn on the engine*

We therefore need to consider whether the action is “*is pressed*” or “*button is pressed*”, that is, do we need to always include an object within the action annotation, if so, almost all annotation would involve overlapping, or we need to handle the learning model carefully later (not implemented in this assignment) since it may produce difficulties for A.L. to distinguish intransitive verbs (actions) from transitive verbs (also actions). In deliverable 2, I tried to exclude the objects from active sentences for a prettier and neater visual effect. Furthermore, I want to discuss why I would like to use wording of “action” instead of “event”, because “event” sounds relatively macro and is constituted by a series of actions, and in the document that I apply my annotation legend, a large part of text is to introduce how actions are functioning together to enable an event (or call it a bigger action), thus I prefer to use word “action” in order to make better sense (in computer science thinking); for example: if we say “*the built-in flash will pop up and fire automatically*” is an event, we are less like to call “*the built-in flash works by charging up a capacitor with electricity then releasing the stored power in a burst of light*” also an event; just like when we call “press the button” an event, we don’t want to call “button sensing our touch and allowing electronics to pass” an event at the same time.

Is the entire content of the specification covered by your annotations? If not, what parts were not annotated? Why were certain parts not annotated? Is it because those parts were not relevant to camera behaviours?