After considering both use cases and user stories, we conclude that the user stories method is a better way to represent the systems requirements for our project. Just as with the selection of our software process model, this comes from consideration of our circumstances compared to the strengths and weaknesses of each model.

The largest, most important draw of the user stories model is that it is extremely simple and to the point. Its structure, where a given user wants to do something for a given reason, is very, very close to how our client expresses what he wants out of the software, and as such is extremely attractive. Along with the emphasis on action and reason, user stories allow for the communication of overarching requirements goals as opposed to more nitty-gritty details, which fits with the iterative model of development we have chosen, since iterative models work towards a general end goal but don’t care about details at each stage.

The largest, most important draw of the use cases model is that done correctly, it can map out the structure of the very application needed, at least on some level. However, this backbone given and its greater emphasis on specific details compared to user stories does not lend it quite as well to an iterative model; and furthermore, it is more technically designed than user stories, which are closer to natural language. As such, while not a terrible alternative, it just doesn’t provide nearly as many advantages as with user stories.

As such, we have selected user stories to represent our system requirements.