More UML Relationships

`<<Subsystem>>` - a type of stereotyped component that represents independent, behavioral units in a system
`<<include>>` - behavior that is based on another, ex. including view cart after checkout
`<<extend>>` - by several options, provide optional functionality, ex. view items -> browse items, add to wishlist, etc

Drawing a Use Case Diagram

- name should communicate purpose
- minimize line crossing
- layout behaviors and roles that are related to each other
- minimize the number of relationships in one view

Type	Description
Actors	Identify actors (types of users)
Use Cases	Identify use cases (functionality)
Access	Determine which user type has access to which functionality
External	Identify external systems
Include	Identify common functionality (`<<include>>`)
Extend	Identify optional/extending functionality(`<<extend>>`)
Generalize	Identify generalizations between actors and use cases

The software requirements document is the official al statement of what is required of the system developers/ Should include both a definition of user requirements and specification of those system requirements