**Glossary.**

**SDCL (Software Development Life Cycle)**

* Project Planning: Define the project purpose, establish feasibility and scope.
* **Requirements:** Elicit and document requirements. What the stakeholder(s) wants.
* **Analysis (often combined with requirements):** Understand and analyze the implications of the requirements. Amend and negotiate as required, separate wish list from to-be-built.
* **Design:** Plan a possible solution. This may involve structural, database, workflow, interaction, and interface decisions.
* **Implementation:** Building the solution as per the design documents.
* **Integration and testing:** Testing the system and components to ensure they interact appropriately, meeting the requirements.
* Deployment: Software is installed and or deployed ready for use.
* Maintenance: May involve a small maintenance period, with longer term maintenance separated into a new project or contract.

**Use Case Diagram:** Enables a high level, quick overview of system functionality.

**Use Case:** Goals or functions of a system (must use a verb and a noun – “view book’).

**Actor:** A person, or entity, that initiates a use case.

**Association/Relationship:** Connecting line between use case and actor.

**Transactional Association:** When two classes relate to each other through their connection to another class.

**Inheritance:** Refers to extending classes, in particular the child-parent relationship.

**Generalization:** Refers to extracting the commonalities between two classes, extracting them and referring to a parent class.

**Specialization:** Refers to inheritance when the child class has all attributes and methods of the parent class, as well as additional attributes and/or methods.

**Class Diagram:** A diagram that indicates the class attributes, methods, and relationships.

**Object Diagram:** A class diagram that has specific attributes and names based on an instance of a class i.e., bookName = Book Name, authorName = Author Name.