Document Delivery for a project:

**Use Case (required):**

* A use case is a “narrative” that describes the experience of a user of the system as if it existed.
* The scenario of a use case covers the user’s entire experience from approaching (logging in to) the system with some intention, to leaving (logging off) having satisfied the user’s intention.
* A use case is written entirely in third person present tense.
* A “top level” use case only describes what the user actually sees, hears, and does. It must not mention anything about what the system does that is not immediately visible to the user.
* The use case must not use technical jargon, but rather it must simply describe, like an announcer at a sporting event or as observed by a “fly on the wall.”

**Project Charter:**

A document with the following sections:

1. Executive Summary (needed actions or decisions)
2. Justification
   1. Statement of problem or opportunity,
   2. Business case (why us, why now, consistent/aligns with out mission)
3. Project description
   1. Project objectives Goal statement, how to measure success)
   2. Constraints, assumptions, risks
4. Scope and deliverables (what will be delivered, what will not be addressed)
5. Timeline (summary schedule, milestones)
6. Budget (how much will it cost, who will pay, how will we pay for it, breakeven when)
7. Team (skills each member brings to the project)

**Requirements document:**

1. Stakeholders: who are they, what are their interests
2. Requirements can be documented in different ways, but they usually form some kind of list.

**Initial “block diagram”(s).**

System architects draw boxes and lines or blocks and arrows. They say software engineers communicate by drawing boxes and lines in PowerPoint.

Data:

Before modeling a database, we must

1. collect the information that we want to have,
2. identify relationships, and what can be derived and from what,
3. decide which things to store and which things to derive.
4. Identify transitive dependence and redundancy (normalization)
5. Assign data to modules (tables) and identify keys (primary and foreign)

Physical Models:

A physical description describes the hardware, network components, and existing things (e.g. software products) you buy or otherwise use.

Logical Models:

A logical description describes functional and data modules and the relationships among them. There are many different model representations. They always use shapes and arrows, but the shapes and/or arrows may indicated different kinds of things depending on the model “formalism”.

Our curriculum says that we will teach you “**UML**”, but there are many models within UML and even more within the OMG umbrella (UML’s parent).

*Structure*: Class diagram, component diagram

*Behavior*: use case diagram, activity diagram, sequence diagram, StateCharts

Common **database** diagrams:

*Structure:* Relational model, Entity relationship model, hierarchical or network model.

*Behavior:* (business process / workflow models) Petri, BPMN, BPEL, IFML