**DELIVERABLE 5:** Stage 3 requirements: **Final system delivery**

The specifications for the final system are:

* *Title page*: Course name and number; year and semester; project title; group name; member names; instructor name.
* *Table of contents*
* *Requirements Implemented*: A list of the requirements that have been implemented in Stage 3 (final) system. Please refer to Phase II Goals in the customer’s presentation slides (Mandatory Requirements 2-9 and Stretch Requirements 1-5).

Please indicate which implemented requirements are *mandatory*, and which are *stretch*

(optional).

At its minimum, a requirement should have the following:

* It’s ID
* A description of what the requirement is.
* It’s origin (where did this requirement come from (give specific reference or name of author, etc.)?)
* Supporting example – as needed: screenshot, diagram, etc. to explain this requirement.

As a starter, please see the following, but there is tons on the web on requirements:

<https://hubtechinsider.wordpress.com/2011/07/28/how-do-you-write-software-requirements-what-are-software-requirements-what-is-a-software-requirement/>

* *Test cases*:
* Create test cases, test the Stage 3 system, and log the test results.
* If there are test cases that have failed but not resolved by the time of submission of Stage 3 then these should be identified explicitly.
* Draw a matrix showing requirements of the Stage 3 system X test cases involved in its verification.
* *System design* as at Final delivery
* Use case diagram
* Class diagram
* Sequence diagram(s) for specific scenarios.
* Package diagram

**NOTE**: Each diagram must be accompanied by an explanation: (1) *what* the diagram conveys and (2) the *rationale* for the design (e.g., the choices made for the classes; modularity; the correspondence between the use case and customer’s requirements; how the functions interconnect to satisfy the scenario; criteria used to create packages; etc.).

* *Design patterns*:

Give graphical representation (together with its explanation) of any design patterns used in the implementation; if not, justify giving technical reasons why the design of your program could not be implemented with design patterns.

* *Code and design inspection* data:
  + *For a sample subset of implemented requirements (mandatory or stretch), conduct inspection of the design and code that implements these requirements. Include a copy of all the completed inspection sheets in with this Deliverable.*
* *Implementation* in C++
  1. Does the code satisfy the design? Explanation and justification of the implemented parts of the system with specific references to the code *and* the design of the system. (Maximum 1 page)
  2. Code files.
* *Development plans* from project start to completion: Timeline and Agent-task view. For specifications, see Stage 2 requirements.
* *Lessons learnt* and retrospective analysis (see above section on: **Lessons learnt and retrospective analysis [NOTE: Part of Stage 3]**).
* *Executable:* Phase II system installation file.
* *System Installation Guide:* Simple to understand and follow instructions on how to install the system, giving platform requirements.
* *User Guide:* Imagine writing a step-by-step guide for a non-specialist. Show screenshots to make it easy to understand how to use the system and what results the system produces. Consider writing several key scenarios and then showing, using screenshots, the execution of these scenarios.
* *(If stretch goal implemented) Training Video:* A short video demonstrating to the user how to install and use the software.

**NOTE:** For the last three items (*System Installation Guide, User Guide, and Training Video*), please bundle with the executable and ensure that they are not inside the report itself.