



Software Engineering

Project

Context

Design and implementation of a computer application based on systematic, disciplined and quantifiable approaches to software development, with the following results.

Requirements specification:

- Vision document: description of the solution to be developed in its main characteristics and context of operation;
- Requirements specification document:
 - Use case model, with a minimum of 2 actors, 10 main use cases (not considering cases of inclusion, extension or specialization) and detailed definition of 7 or more use cases;
 - Supplementary specification, with a minimum of 10 requirements;
 - Glossary, with a minimum of 5 definitions.

Logical architecture:

- Domain model;
- Use case realization: set of diagrams (interaction, timing) describing the implementation of 2 high-priority use cases, corresponding to the main theme of the proposed solution (not considering use cases related to user management, authentication, or other general areas);
- Architecture of mechanisms: set of diagrams (classes, composite structure) that describe the architecture of the mechanisms resulting from the realization of the use cases considered;
- Solution overall architecture: set of diagrams that describe the overall architecture of the solution in terms of subsystems and their relationships.

Detailed architecture:

- Detailed architecture of the application, specifying how the implementation of the developed solution and the mechanisms developed will be implemented, in terms of the application platforms and programming languages chosen for the implementation of the solution, which should include:
 - Dynamics model;
 - Detail of parts and mechanisms;
 - Test architecture;
 - Application deployment model.

Implementation:

- Domain validation test prototype: implementation and testing of 2 use cases (minimum); implementation and testing of 4 domain classes (minimum);
- Applicational prototype: implementation and testing of 3 or more use cases (including those already implemented in the test prototype), according to the requirements specification.

Implementation requirements:

1. Implementation of the test prototype, using command console mode, independent of application platforms and support infrastructures:
 - a. Definition of test cases based on use cases:
 - i. Test steps (corresponding to the scenario to be tested);
 - ii. Test data;
 - iii. Expected results;
 - b. Implementation of resources access adapter (*stub*);
 - c. Implementation of domain layer test cases;
2. Delivery of the source code of the test prototype, with test script directly executable in console mode (must include code compilation commands if applicable)
 - a. The *test script* must have the designation `test.<ext>`
(`<ext>` depends on the execution platform)
3. Application prototype implementation:
 - a. Implementation of presentation mechanisms for use cases selected for testing;
 - b. Implementation of resource access mechanisms for use cases selected for testing;
 - c. Implementation of use cases not considered in the test prototype.
4. Delivery of the source code of the application prototype, with **readme.md** containing the requirements and instructions for deployment and execution of the prototype.