Supplementary Specification (FURPS+)

Functionality

Functionality refers to the ability of a software system to deliver features and functions that meet user requirements and ensure user satisfaction and system performance, even under specific conditions. The following requirements focus on authentication, authorization, and reporting:

• The system should implement robust authentication and authorization mechanisms for all users and functionalities.

• The program will undergo careful and systematic auditing to ensure effective implementation and alignment with the planned objectives.

• Reports generated by Jacoco will provide valuable insights on project progress, performance, issues, and risks.

Usability

Usability emphasizes the ease of understanding, learning, and using a software system under specified conditions. The requirements for usability include aesthetics and design, error prevention, and documentation:

Aesthetics and design

• The user interface should have a simple, intuitive, and consistent design.

• The application should support the English language.

Error prevention

• All methods will be thoroughly tested using JUnit to prevent errors in different scenarios.

• The team should follow a test-driven development approach and develop a comprehensive set of automated tests for user stories, classes, and methods.

Documentation

• Project documentation should be available in the project repository's "docs" folder, in Markdown format, and, when applicable, using UML notation.

• The documentation should include the development process for each user story.

• A readme.md file in the root folder of the repository should provide instructions for building, deploying, and executing the solution.

Reliability

Reliability refers to a software system's ability to maintain performance levels over time and meet user requirements without degradation or issues. The requirements for reliability are as follows:

• Project documentation should always be accessible in the project repository.

• The system must support data persistence, either in memory or in a relational database. The final solution should include a deployment of a persistent relational database and the ability to initialize default data.

• Each sprint should include a presentation during the sprint review, providing further specifications and client feedback.

Performance

Performance testing is essential for measuring and analyzing a system's performance under different loads and conditions. The requirements for performance include developing relevant tests for user stories, classes, and methods to ensure optimal performance and functionality.

Supportability

Supportability refers to a program's flexibility to undergo changes and adapt to new conditions, ensuring it meets evolving user needs. The requirements for supportability include:

• The product should support localization and multiple human languages, including at least English.

• The GitHub repository should provide nightly builds with published results and metrics.

• The repository should include necessary scripts for building and deploying the solution on different systems, supporting at least Linux and Windows.

• Jacoco reports should provide insights on project progress, performance, issues, and risks, aiding in tracking and communicating the project status to stakeholders.

+

Design Constraints

Programming Languages

• The program will be primarily implemented in Java, but other languages may be used for specific requirements.

Mandatory Standards/Patterns

• Best practices for requirement identification, object-oriented software analysis and design should be adopted during system development.

• Recognized coding standards, such as CamelCase, should be followed.

• Javadoc should be used to generate useful documentation for Java code.

Software Process

• The application should ensure data persistence between runs.

Use of Development Tools

• JUnit should be used to test all methods and prevent errors.

• ANTLR can be used for design support, feedback, and grading.

Implementation Constraints

• The application must be programmed in Java using IntelliJ IDE.

• The source code, documentation, and related artifacts should be versioned in a GitHub repository.

• The use of C programming language is specific to SCOMP requirements.

The solution design and implementation should be based on threads, condition variables, and mutexes.

• The solution should be deployed using multiple network nodes to enable the deployment of the relational database server and shared board server in different nodes, preferably in the cloud.

Interface Constraints

Interface constraints specify the limitations and requirements for the system's interaction with external systems, following predefined standards and protocols. The interface constraints are as follows:

• The JaCoCo plugin should be utilized to generate the coverage report.

• The application should be supported by Linux and Windows operating systems.

• The application is to be developed using Oracle OpenJDK 19.0.2 version.

• The client should automatically generate the "views" of boards without user intervention, utilizing AJAX.

Physical Constraints

Physical constraints refer to predetermined requirements and limitations regarding the hardware that will host the software system. These constraints ensure the safe and efficient functioning of the system, and include aspects such as material, shape, size, and weight.