UNIVERSIDAD DE OVIEDO
ESCUELA DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA DEL SOFTWARE

TRABAJO DE FIN DE GRADO

IR-Board

Requirements Management Platform

Author:
Javier Carrasco Arango
Tutors:
Jorge Álvarez Fidalgo
Benjamín López Pérez

2026-02-18

Declaration of originality

Special thanks to

Summary of Chapters

Chapter 1. Introduction

1.1 Abstract

1.2 Keywords

1.3 Resumen

1.4 Palabras clave

Chapter 2. Project context and planning

2.1 Project Context and Motivation

2.2 State of the Art and Related Work

2.2.1 Existing Requirement Management Tools

2.2.2 Limitations of Existing Solutions

2.2.3 Key Lessons and Takeaways

2.3 Problem Definition

2.4 Objectives

2.5 Scope and Limitations

2.6 Development Methodology and Planning

Chapter 3. System architecture

3.1 Overall System Architecture

3.2 Technology Stack

3.3 Infrastructure Decisions

3.4 Justification of Technology Choices

Chapter 4. Feasibility study

4.1 Feasibility of Alternatives

4.2 Selected Alternative and Justification

4.3 Risk Assessment (High-Level)

Chapter 5. Project management and planning

5.1 Project planning

5.1.1 ?

5.2 Work Breakdown Structure (WBS)

5.3 Risk Management

5.4 Timeline and Milestones

5.5 Budget and Resources

5.6 Lessons Learned / Reflection

Chapter 6. System analysis

6.1 System Definition

6.1.1 Determination of System Scope

6.1.2 System Context Diagram

6.2 Requirements Elicitation and Specification

6.2.1 Functional Requirements

6.2.1.1 project management

- The system must allow an admin to create a project
  - The following are required:
    - Project name
    - Description
    - Owner
  - The following are optional:
    - Priority, either:
      - Ternary (High, Medium, Low) Predefined
      - MOSCOW (Must, Should, Could or Won't have)
- The system must allow an admin to deactivate a project
  - The system must ask for confirmation before deactivating
  - The system must put the project on read only mode
- The system must allow an admin to reactivate a deactivated project
- The system must allow an admin to modify an active project
- The system must allow to link users with a project
  - The system must allow an admin to link users to a project as project manager
  - The system must allow an admin or project manager linked to the project to link users to a functionality on said project as a stakeholder user
  - The system must allow a project manager to link users to one or more functionalities of a project as requirement engineers.
- The system must allow access to the project description/dashboard to users linked to it or a functionality of it.
  - The system must show the total split of requirements by their states (pie chart)
  - The system must show the different functionalities of the project
- The system must allow a project manager to mark as approved all elements in a project
- The system must allow a project manager to add a functionality to a project
  - A functionality needs a name and unique set of letters for its dynamic identifier.
  - The system must automatically attempt to get the letters for the dynamic identifier from the name
    - The system must take the first letter from every word in the name.
    - If the identifier is already in use by another functionality on the same project, the system will suggest one letter more of each word on the name.
    - If the system cannot generate a new set of letters to identify its requirements, a message must be shown to the project manager.

  - The system must check the letters for the identifier are not the same from another functionality on the same project.
    - The system must deny adding a functionality that breaks the rule above.
  - The system must automatically link the project manager to the new functionality
- The system must allow a project manager to modify a functionality.
- The system must allow a project manager to deactivate a functionality.
  - The system must ask for confirmation before deactivating
  - The system must put the functionality (elements contained by it) on read only
- The system must allow a project manager to reactivate a functionality.
- The system must allow a project manager to mark as approved all elements in a functionality
- The system must allow a project manager to generate a baseline for a project.
  - The system must perform a snapshot of the project once a baseline is set.
- The system must allow a project manager to export the project's requirements onto a pdf file

6.2.1.2 Stakeholders management

- The system must allow any user linked with the project access to view its stakeholders
  - The system must show if a stakeholder is flagged as pending review
  - The system must show the identifier, the name and part of the description
  - The system must allow to collapse and expand stakeholders with children
  - The system must allow to view the details of a stakeholder
    - The system must show:
      - All atributes of a stakeholder
      - All requirements linked to it
- The system must allow a project manager to add a new stakeholder to a project
  - The system must only allow a stakeholder to be added to a project the user is linked to.
  - The system must ask for the following:
    - Name
    - Description
  - The system must generate the identifier for the stakeholder
- The system must allow a project manager or requirement engineer to link a stakeholder to one or more requirements on the same project
  - The system must only allow the user to link the stakeholder to a requirement on a functionality they are linked to
- The system must allow a project manager or requirement engineer to unlink a stakeholder from one or more requirements
  - The system must only allow the user to unlink a stakeholder from a requirement of a functionality they are linked to.
- The system must allow a project manager to deactivate a stakeholder from a project the user is linked to
  - The system must show the user the amount of entities affected by the deactivations
  - The system must flag all entities linked as pending review

    - The system must ask for confirmation before deactivating
    - The system must put the stakeholder on read only mode
- The system must allow a project manager or requirement engineer to modify a stakeholder
    - The system must flag as pending review linked entities upon saving with changes.

6.2.1.3 Requirement management

- The system must allow users linked to a project or functionality of a project access to its requirements
    - The system must only allow users linked to the functionality of a functional requirement access to it
    - The system must allow to collapse and expand requirements with children
    - The system must show the dynamic identifier, the name, state and part of the description
        - The system must show as “state” on a non-functional requirement whether it is passed or not
    - The system must allow to view the details of a requirement
        - The system must show:
            - The internal unique identifier
            - All atributes of a requirement
            - All stakeholders linked to it
            - All requirements cross-linked with it
            - All documents linked to it
            - The previous dynamic identifiers
- The system must allow a requirement engineer or a project manager to add a requirement to a project the user is linked to
    - The system must only allow a functional requirement to be added to a functionality the user is linked to.
    - The system must allow the user to generate a requirement as a child of another requirement (nesting).
    - The system must assign automatically the dynamic identifier
        - The identifier must be based on its relation to other requirements.
        - The identifier must represent if it is a functional or non functional requirement (FR or NFR)
        - The identifier must represent the folder/component that holds the requirement (user management → UM)
    - The system must assign automatically the internal unique identifier
        - The identifier must represent the project that will hold the requirement
        - The identifier must represent whether the requirement is functional or non functional
        - The identifier must have a random element to ensure a low colision rate
    - The system must ask for the following data for a functional requirement:
        - The following are required:
            - Name
            - Description
            - a priority as predefined on the project

    - The following are not required:
      - Stability
      - Origin
  - The system must ask for the following data for a non-functional requirement:
    - The following are required:
      - Name
      - Description
    - The following are optional:
      - Measurement unit
      - comparison operator
        - equal to, less than or greater than
      - Threshold value
        - This value represents the minimum value to mark the requirement as passed
      - Target value
        - This value represents the optimal value desired by the team
      - Actual value
        - This value represents the current status of the measurement
- The system must allow a project manager or requirement engineer to link a requirement on a functionality they are linked to, to another entity
  - The system must allow to link a requirement with a stakeholder of the same project
  - The system must allow to un-link a requirement with a stakeholder
  - The system must allow to link a requirement with one or more requirements of functionalities of the same project the user is linked to
  - The system must allow to un-link a requirement with other requirements of functionalities of the same project the user is linked to
  - The system must allow to link a requirement with one or more documents of the same project.
  - The system must allow to un-link a requirement with one or more documents of the same project.
- The system must allow a requirement engineer or a project manager to deactivate a requirement on a functionality they are linked to
  - The system must show the user the amount of entities that will be affected by the deactivation
  - The system must ask for confirmation
  - The system must flag any requirements linked to the deactivated requirement as pending review
  - The system must put the requirement on read only
- The system must allow a requirement engineer or a project manager to reactivate a requirement on a functionality they are linked to
  - The system must automatically flag as pending review the reactivated requirement
- The system must allow a requirement engineer or a project manager to modify a requirement on a project

  - The system must only allow a project manager or requirement engineer to modify functional requiremets of a functionality the user is linked to.
  - The system must flag linked requirements as pending review upon saving with changes.
- The system must allow a project manager marking as approved one or more requirements
  - The system must only allow to mark as approved a requirement that is pending approval, not pending review nor deactivated.
- The system must allow a project manager or requirement engineer linked to a functionality of the project to change the position of a requirement
  - The system must allow reordering of functional requirements to users linked to the same functionality.
  - The system must update the dynamic identifier automatically
  - The system must set the order of requirements using a floating point order value
- The system must allow a project manager or requirement engineer to review an requirement flagged as pending a review
  - The system must allow removing the flag if no changes are required.
  - The system must allow modifying the requirement upon review.
    - The system must remove the flag upon saving with changes.
    - The system must flag the linked entities as pending a review.

6.2.1.4 User management

- The system must allow an admin to invite new users to the system
  - The system must provide different levels of authorisation.
    - The system must have the levels: Admin, project manager, requirement engineer and stakeholder user
  - The system must ask the admin to set the name, surname, and email of the invited user
    - The system must generate an signup code as a temporal password
    - The system must automatically send an invitation with the signup code to the email of the invited user
- The system must allow an admin to modify the name and surname of a user from the system
- The system must allow an admin to generate a new invite with a signup code for a user
- The system must allow any user with valid credentials to sign in to the system
  - The system must prompt any user signing in with a signup code to set a permanent password.
    - The system must ensure the password is between 15 and 64 characters long.
    - The system must make use of a random salt specific of each user.
    - The system must remove any password or signup code of the user upon seting a permanent password.
  - The system must temporally block the user after 3 consecutive failed attempts
- The system must allow an admin to deactivate a user from the system
  - A deactivated user remains on the system but cannot access it
- The system must allow an admin to reactivate a user from the system

6.2.1.5 Document management and modelling

- The system must allow users linked to a project access to documents of that project
  - The system must show entities linked to the document.
- The system must allow a project manager or a requirement engineer to add document to a project
  - The user must be linked to the project
- The system must allow a document to be linked to one or more requirements of the same project
  - The system must flag those requirements linked to it as pending a review if the document is altered
- The system must allow a project manager or requirement engineer to update a document
  - The user must be linked to the project the document is on.
  - The system must flag as pending a review any requirements linked to the document
- The system must allow a project manager to disable a document
  - The system must flag as pending a review any requirements linked to the document
- The system must allow a requirement engineer to model diagrams using a Draw.io integration

6.2.1.6 Concurrency

- The system must block other users from modifying an entity that another user is already modifying
  - The system must release automatically the entity if the user modifying it saves and exits (stops modifying).
  - The system must release automatically the entity after a predetermined timeout period
  - The system must release automatically the entity if the user editing it modifies another entity
  - The system must only accept changes to the entity from the user who holds the entity
- The system must display for other users who is modifying the entity

6.2.1.7 Search and filtering

- The system must allow searching an entity by internal unique identifier.
  - The system must search lexically
  - The system must allow the user to see the details of the found entity
    - only if an exact match occurs,
    - only if the user has access to it.
- The system must allow users to filter entities they have access to
  - The system must allow filtering out deactivated requirements
  - The system must allow filtering requirements based on priority
  - The system must allow filtering requirements based on state
  - Any filter must be reversible; ascending or descending order

6.2.2 Non-Functional Requirements

6.2.3 Actors and Use Cases

6.3 Analysis of Subsystems

6.3.1 Subsystem Description

6.3.2 Interfaces Between Subsystems

6.4 Use Case Analysis

6.4.1 Use Case 1

6.4.2 Use Case 2

6.5 Class Analysis

6.5.1 Class Diagrams

6.5.2 Class Descriptions

6.6 User Interface Definition

6.6.1 Layout and Components

6.6.2 Interaction Design

6.6.3 Navigation Flow

6.7 Test Planning

6.7.1 Unit Test Plan

6.7.2 Integration Test Plan

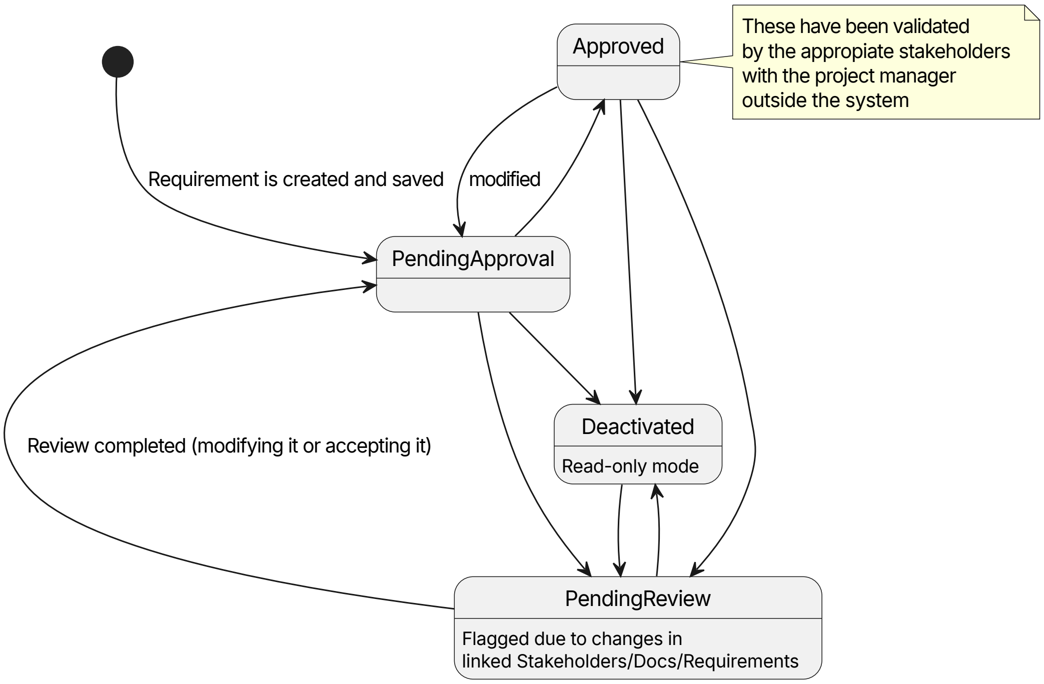
6.7.3 System Test Plan

Chapter 7. System design

7.1 Use Case Design

7.2 Class Design

Lifecycle of a Requirement (IR-Board)


Approved
Requirement is created and saved
modified
PendingApproval
Deactivated
Read-only mode
Review completed (modifying it or accepting it)
PendingReview
Flagged due to changes in
linked Stakeholders/Docs/Requirements

These have been validated
by the appropiate stakeholders
with the project manager
outside the system

7.3 Module Architecture Design

7.4 Data Design (Database / ER diagrams)

7.5 Detailed Test Plan / Test Cases

Chapter 8. System implementation

8.1 Development Environment Setup

8.2 Code Implementation

8.3 Unit Testing

8.4 Integration Testing

8.5 System Testing

8.5.1 Usability Tests

8.5.2 Accessibility Tests

8.6 User and Developer Manuals

8.7 Migration Scripts Implementation

Chapter 9. Deployment and system acceptance

9.1 Deployment Plan

9.2 Acceptance Testing

9.3 Maintenance Considerations

Chapter 10. Conclusions and future work

10.1 Conclusions

10.2 Future Work / Enhancements

10.2.1 Document management

- (future work) The system must allow a requirement engineer fill a tabular use case
  - The system must have the following fields:
    - Name
    - Description
    - Actors
    - Initial Condition
    - Preconditions
    - Postconditions
    - Normal flow
    - Alternative flows
    - Exceptions
- (future work) The system must allow a requirement engineer fill a scenario
  - The system must have the following fields:
    - Name
    - Description
    - Actors
    - Initial Condition
    - Preconditions
    - Postconditions
    - Flow
- The system must allow a project manager generate a customized pdf of requirements for a stakeholder
- The system must allow a project manager define a stencil for pdf generation of the srs export

10.2.2 Variant control

Highly ambitious—only if time permits. Keep this in mind even if not implemented: requirement reuse. Defining templates or abstract projects. In the industry, “copy and paste” is the standard approach.

10.2.3 Search and filtering

The system must allow a user to search requirements by text