Analysis report

Grado en Ingeniería Informática - Ingeniería del Software

Diseño y Pruebas II

Curso 2023 - 2024

Código de Grupo: C1-001		
Autor	**Correo**	**Rol**
Pedro Pablo Santos Dominguez	pedsandom@alum.us.es	Desarrollador

Repositorio: <https://github.com/DP2-2023-2024-C1-001/Acme-SF-D02.git>

Índice de Contenidos

1. Resumen Ejecutivo

El informe de análisis es un registro detallado de lo que se hace en cada etapa del proyecto. Ayuda a respaldar las decisiones que se toman, asegurando que el trabajo se haga bien desde el diseño hasta la implementación. También se asegura de que el producto final sea lo que realmente necesitan y quieren los usuarios. En resumen, es una herramienta importante que guía el progreso y éxito del proyecto.

Participantes:

Desarrolladores: Pedro Pablo Santos Dominguez

Rol y Responsabilidades:

Desarrollador : Crear la configuración de desarrollo, personaliza el proyecto inicial, implementa características y realiza testing informal.

Plan de Acción:

Crear e inicializar el repositorio en Github.

Añadir las tareas a realizar en Github.

Asignación, por parte del manager, de las tareas a los desarrolladores correspondientes.

Inicializar las ramas para ir resolviendo las tareas

Crear los pull request asignando un reviewer para comprobar que la tarea está resuelta de forma correcta. Si no, esa tarea en concreto pasa a estar resuelta y el manager se encarga de crear otra tarea para realizar la corrección de la misma.

2. Control de Versiones

Fecha	Versión	Descripción
07/03/2024	V1.0	Creación inicial del documento.

3. Introducción

El propósito de este informe de análisis es evaluar en qué medida se cumplen los requisitos definidos para el proyecto Acm-SF. Para lograrlo, se ha realizado un análisis minucioso de cada requisito, identificando áreas de mejora y explicando las decisiones tomadas.

El informe presenta un registro detallado para cada requisito, que incluye la descripción original del requisito, las conclusiones del análisis y las acciones tomadas para abordar cualquier problema identificado.

Se presta especial atención a los requisitos considerados más importantes y complejos, tanto en términos de funcionalidad como de documentación.

Además, al final del informe se proporciona una conclusión general y se lista la bibliografía consultada. Esto sirve como una referencia para la toma de decisiones y para mejorar continuamente el proyecto.

Esta introducción tiene como objetivo facilitar la comprensión del informe de análisis y destacar los aspectos más significativos del proyecto.

4. Contenido

Tarea-002:

Code audits are essential pieces to ensure the quality of a project. The system must store the following data about them: a code (pattern “[A-Z]{1,3}-[0-9]{3}”, not blank, unique), an execution date (in the past), a type (“Static”, “Dynamic”), a list of proposed corrective actions (not blank, shorter than 101 characters), a mark (computed as the mode of the marks in the corresponding auditing records; ties must be broken arbitrarily if necessary), and an optional link with further information.

Tras estudiar el requisito, se decidió crear dentro del package entities, un package para la entidad “CodeAudit”.

Para implementar la entidad “CodeAudit”, hubo que crear una clase de java, “CodeAudit”, en la que hay que configurar las clase con la anotación de “javax.persistence”: “@Entity”; y las anotaciones de “lombok”: “@Getter” y “@Setter”. Además, se debe extender la clase de “AbstractEntity”, que proporciona un id y una versión a la entidad. También, se debe añadir “serialVersionUID”. Luego, se modelan los atributos con sus respectivas restricciones, que son notaciones importadas de “”javax.validation.constraint”. Despues, habia que hacer dos relaciones ManyToOne desde esta entidad: una a “Auditor” y otra a “Project”. Finalmente, se comprueba que todo funciona correctamente, y se realiza el “commit and push”, correspondiente, y la “pull request”.

En este requisito, se encontró un inconveniente con el atributo “mark”, ya que es un atributo derivable y que para calcularlo todavía no disponemos del conocimiento necesario.

Tarea-003:

The result of each code audit is based on the analysis of their audit records. The system must store the following data about them: a code (pattern “AU-[0-9]{4}-[0-9]{3}”, not blank, unique), the period during which the subject was audited (in the past, at least one hour long), a mark (“A+”, “A”, “B”, “C”, “F”, or “F-”), and an optional link with further information.

Tras estudiar el requisito, se decidió crear dentro del package entities, un package para la entidad “AuditRecord”.

Para implementar la entidad “AuditRecord”, hubo que crear una clase de java, “AuditRecord”, en la que hay que configurar las clase con la anotación de

“javax.persistence”: “@Entity”; y las anotaciones de “lombok”: “@Getter” y “@Setter”. Además, se debe extender la clase de “AbstractEntity”, que proporciona un id y una versión a la entidad. También, se debe añadir “serialVersionUID”. Luego, se modelan los atributos con sus respectivas restricciones, que son notaciones importadas de “”javax.validation.constraint”. Despues, habia que hacer una relación ManyToOne desde esta entidad hacia “CodeAudit”. Finalmente, se comprueba que todo funciona correctamente, y se realiza el “commit and push”, correspondiente, y la “pull request”.

Tarea-004:

The system must handle auditor dashboards with the following data: total number of code audits for “Static” and “Dynamic” types; average, deviation, minimum, and maximum number of audit records in their audits; average, deviation, minimum, and maximum time of the period lengths in their audit.

Este requisito se estudió, y se observó que debía realizar un formulario. En principio, los atributos tenían tipos como Integer y Double, pero se cambiaron a tipos primitivos como int y double. Esto se debió a que los atributos del formulario no deben ser nulos, y como sabemos los tipos primitivos no permiten valores nulos.

Este formulario es “AuditorDashboard”, está creado en “src/main/java/acme/forms” y todos los formularios deben extender de “AbstractForm”.

Tarea-005:

Produce assorted sample data to test your application informally. The data must include two auditor accounts with credentials “auditor1/auditor1” and “auditor2/auditor2”.

Tras estudiar el requisito, se decidió crear todos los sample-data de las correspondientes entidades. Además, de en el csv de “user-account.csv”, añadir dos líneas con dos nuevas credenciales.

La implementación se basaba en la creación de ficheros csv, en la carpeta “src/main/webapp/WEB-INF/resources/sample-data”, de cada entidad, y la escritura de sus atributos en el correcto formato. También, la realización de los nuevos roles auditor, debían añadirse en el csv de “user-account.csv”, y la posterior creación del csv de la entidad “Auditor” correspondiente.

Tarea-013:

There is a new project-specific role called auditor, which has the following profile data: firm (not blank, shorter than 76 characters), professional ID (not blank, shorter than 26 characters), a list of certifications (not blank, shorter than 101 characters), and an optional link with further information.

Tras estudiar el requisito, se decidió crear dentro del package roles, la entidad “Auditor”.

Para implementar la entidad “Auditor”, hubo que crear una clase de java, “Auditor”, en la que hay que configurar las clase con la anotación de “javax.persistence”: “@Entity”; y las anotaciones de “lombok”: “@Getter” y “@Setter”. Además, se debe extender la clase de “AbstractEntity”, que proporciona un id y una versión a la entidad. También, se debe añadir “serialVersionUID”. Luego, se modelan los atributos con sus respectivas restricciones, que son notaciones importadas de “”javax.validation.constraint”. Finalmente, se comprueba que todo funciona correctamente, y se realiza el “commit and push”, correspondiente, y la “pull request”.

5. Conclusión

En conclusión, el análisis detallado realizado para evaluar el cumplimiento de los requisitos del proyecto Acm-SF ha permitido identificar áreas de mejora y tomar decisiones importantes para abordar posibles problemas. Se ha prestado especial atención a los requisitos más críticos, tanto en términos de funcionalidad como de documentación. Este informe proporciona una guía útil para la toma de decisiones y la mejora continua del proyecto, destacando la importancia de seguir revisando y ajustando en función de las necesidades del mismo.

6. Bibliografía

Intencionadamente blanco.