Informe de Análisis

GRUPO C1.02.02

Repositorio: https://github.com/antcardia/Acme-L3-D02

Autores:

Guillermo Galeano de Paz (guigalde@alum.us.es)

Sevilla, 16 marzo de 2023

TABLA DE CONTENIDOS

Resumen ejecutivo

Este es un informe de análisis del “Student # 5”, en concreto el miembro Guillermo Galeano de Paz, dentro del segundo Sprint, donde realizaremos un análisis de este. Este informe se ha realizado tras realización de los requisitos necesarios a completar en el documento que se me ha establecido a realizar (“Student # 5”).

Para este en concreto me he basado al completo a la información dada en el anexo ocho de la asignatura, subido en la plataforma de “Enseñanza Virtual”, de él he podido tener la base, guía o estructura para comprender y realizar este informe.

En el documento voy a realizar un análisis previo y final de cada funcionalidad para poder tener una visión más objetiva de cada implementación

Tabla de revisión

Fecha	Versión	Descripción de los cambios	Sprint
16/03/2023	23.1	Creación del documento	1

Introducción

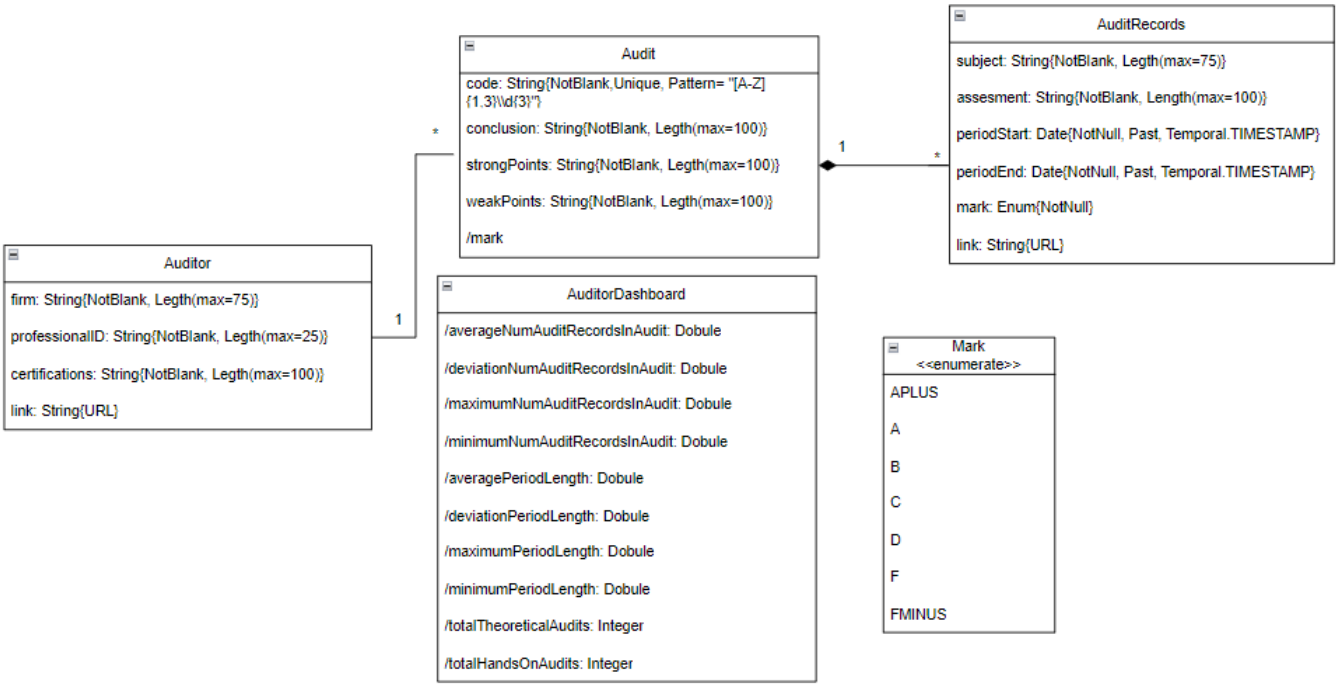
Este es un proyecto dedicado al desarrollo de un sistema de información web para ayudar a la organización Acme Life-Long Learning, S.A (Acme-L3) a gestionar su empresa. Esta organización es ficticia y se especializa en ayudar a los estudiantes en una gran variedad de cuestiones con la ayuda de profesores de renombre.

Para comenzar, hemos utilizado el proyecto Hello World como plantilla para dar los primeros pasos en el desarrollo del sistema. Además, vamos a hacer uso de diferentes herramientas como MariaDB y DBeaver, para la creación y gestión de base de datos, Eclipse, como nuestro espacio de trabajo, Java, como nuestro lenguaje principal de programación y Acme Framework, como nuestro framework.

Contenido

- Crear la entidad AuditRecord
  - Primer Análisis: En un primer análisis, se definieron los atributos que se observan en el diagrama UML, aunque tras avanzar un poco se definió el periodo como dos puntos temporales, fecha de inicio y de fin, de forma que el periodo es una derivación de dichos atributos. Además considero que para las notas, la mejor opción es hacer un enumerado con las distintas opciones.
  - Último Análisis: Al terminar la implementación de AuditRecords, he tenido que especificar la forma en la que se realizará la validación ya que aún no se ha dado en teoría.
- Crear la entidad Audit
  - Primer Análisis: Tras analizar el requisito nº5, los atributos definidos se corresponden con los del diagrama UML. Siendo mark un atributo derivado a partir de los AuditRecords que componen a cada Audit.
  - Último Análisis: Al implementar esta entidad, se han tomado dos decisiones de diseño destacables, la primera es la creación de una relación OneToMany con AuditRecords que aunque no es recomendada, usando la notación "mappedBy" se evitan los problemas de persistencia. La segunda es que al computar las notas, en lugar de hacerlo usando la moda, como el profesor Rafael Corchuelo indica aquí, en lugar de calcular la moda, ahora se debe devolver una lista de las notas de los diferentes AuditRecords que componen dicho Audit.
- Crear la entidad AuditorDashboard
  - Análisis previo: Ya que todavía no se ha explicado como hacer esto en las clases teóricas, asumo que se trata de crear las funciones sin realmente desarrollar su funcionalidad:
  - Último análisis: En este caso, el requisito es muy simple ya que por ahora no requiere implementación. Lo único destacable es que en la última versión de la clase, he dividido totalAudits en totalAudits de Hands-On y Theorietical según la naturaleza del curso.
- Hacer el rol Auditor
  - Análisis previo: El rol auditor no presenta ninguna complejidad y se puede implementar sin tomar decisiones de diseño relevantes.
  - Último análisis: Tal y como está indicado en el análisis previo, no ha presentado ninguna complicación ni ha requerido la toma de decisiones de diseño.
- Hacer un informe de planificación.
  - Análisis: Este informe se realizará de la misma forma en la que se realizó el del sprint anterior, mostrando las tareas realizadas con una breve descripción, el tiempo estimado a estas y el tiempo real. Además incluirá una tabla de amortización y una gráfica de clockify mostrando el trabajo total a lo largo del sprint 2.
- Crear un fichero csv con los datos de prueba necesarios
  - Análisis previo: La tarea de creación de datos de testeo en principio es una tarea sencilla, se pondrán valores que cumplan e incumplan las diferentes anotaciones para ver que se comprueba de forma correcta.
  - Último análisis: Tras probar el funcionamiento correcto de todas las entidades, los datos finales de los archivos csv son todos válidos para que la aplicación pueda ser poblada y ejecutada.

- Hacer un modelo conceptual de las entidades.
    - Análisis: En el caso del Student#5 a la hora de hacer un modelo conceptual, al interpretar los requisitos, he llegado al siguiente modelo UML:



Considero que la entidad Audit está compuesta de varios AuditRecords, que a su vez un Auditor está relacionado con el Audit que haya creado y que AuditorDashboard es una clase relacionada con el Auditor en la que todos los atributos son derivados.

Conclusión

Tras la realización del segundo sprint, considero que la complejidad de los requisitos es buena y no presenta muchas dificultades a la hora de la implementación. Como punto negativo diría que resulta confuso que en los requisitos del sprint haya cosas como validación de periodos o las funciones de los dashboards para las que aún no tenemos el conocimiento necesario para implementarlas.