	Diseño y Pruebas 2 Documentación de la entrega D02
	Informe de Análisis Individual

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Documentación de la entrega D02

Informe de Análisis Individual



Grado en Ingeniería Informática – Ingeniería del Software


Diseño y Pruebas 2

Curso 2023 – 2024

Fecha	Versión
8/3/2024	v1r2


Grupo de prácticas: C1.033		
Autores por orden alfabético	Rol	Correo electrónico
Aguayo Orozco, Sergio - 25604244T	Desarrollador	ahydul1@gmail.com
García Lama, Gonzalo - 47267072W	Desarrollador, Tester	gongarlam@alum.us.es
Huecas Calderón, Tomás - 17476993Y	Desarrollador	tomhuecal@alum.us.es
Fernández Pérez, Pablo - 54370557Y	Desarrollador, Analista	pablofp.33@gmail.com
Youssafi Benichikh, Karim -28823709V	Desarrollador, Operador, Manager	karyouben@alum.us.es

Repositorio: <https://github.com/Ahydul/Acme-SF-D02>

	<p>Diseño y Pruebas 2 Documentación de la entrega D02</p>
	<p>Informe de Análisis Individual</p>

Índice de contenido

1. Resumen ejecutivo	2
2. Tabla de versión	2
3. Introducción	2
4. Contenido	3
4.1 MANDATORY Deliverable D02: Functional requirements	3
4.2 SUPPLEMENTARY Deliverable D02: Managerial requirements	6
5. Conclusiones	8
6. Bibliografía	9

	Diseño y Pruebas 2 Documentación de la entrega D02
	Informe de Análisis Individual

1. Resumen ejecutivo

Este informe detalla el análisis llevado a cabo para los requisitos individuales correspondientes al entregable 2. Cada requisito ha sido sometido a un proceso de evaluación previa, destacando aspectos clave para su implementación y cumplimiento. Este análisis proporciona una base sólida para el desarrollo posterior, identificando áreas de mejora y asegurando la alineación precisa con los objetivos establecidos por el cliente.


2. Tabla de versión

Fecha	Versión	Descripción
19/2/2024	v1r0	creación del documento de análisis individual
25/2/2024	v1r1	Añadido resumen ejecutivo e introducción
8/3/2024	v1r2	Añadido contenido y conclusiones

3. Introducción

En esta segunda entrega del proyecto se han abordado los desafíos surgidos al enfrentarme a la comprensión de los diversos requisitos que integran el proyecto de la asignatura Diseño y Pruebas II. En respuesta a esto, se ha realizado un análisis de aquellos requisitos que presenten incompletitud, ambigüedad, contradicción o inexactitud. En concreto, como esta segunda entrega ha conllevado la realización del uml y su implementación en código, se ha hecho un análisis exhaustivo de los tipos de datos, relaciones entre las entidades y los rangos a popular en la carpeta "sample"

En cuanto a la organización del contenido, opté por estructurar el informe asignando una sección a cada requisito dentro del apartado "Contenido". Cada una de estas secciones contiene el análisis correspondiente de cada requisito para facilitar la comprensión y navegación de dicho documento.

	Diseño y Pruebas 2 Documentación de la entrega D02
	Informe de Análisis Individual

4. Contenido

4.1 MANDATORY Deliverable D02: Information requirements


- training module

2)A training module consists of one or several short-term training activities aimed at extending or updating knowledge and skills related to the topic 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), a creation moment (in the past), some details describing the training module (not blank, shorter than 101 characters), a difficulty level (“Basic”, “Intermediate”, or “Advanced”), an optional update moment (in the past, after the creation moment), an optional link with further information, and an estimated total time.

Para cumplir con este requisito, fue necesario realizar un análisis para comprender las restricciones y los conceptos generales. Posteriormente, se completó la tarea individual de UML mencionada más adelante para ejecutarla correctamente. Se ajustaron los atributos junto con sus anotaciones después de entender completamente la magnitud del modelo.

Análisis:

Esta entidad junto al resto de entidades se añadieron en la ruta “src/main/java/entities/trainingModule”, tras un análisis de las entidades a implementar, se implementó los tipo String como @NotBlank si no especificaba que era opcional y el resto de tipo de datos como @NotNull en el mismo caso, el totalTime se implemento como un primitivo int para que al ser primitivo fuera un atributo obligatorio(omitiendo la anotación @NotNull y ser más eficiente) y siendo su mínimo valor “1”, el enumerado, se creo una clase aparte indicando sus posibles valores y se implemento como un enum DifficultyLevel, los datos tipo String y Date se implementaron con su anotación y restricciones correspondientes, siendo los tipos “Date” en un momento del pasado se añadió en ellos la anotación @Past y se implementó dos relaciones ManyToOne unidireccionales de TrainingModule a project y a developer.

	Diseño y Pruebas 2 Documentación de la entrega D02
	Informe de Análisis Individual

• training session

3) Each training module is made up of training sessions. The system must store the following data about them: a code (pattern “TS-[A-Z]{1,3}-[0-9]{3}”, not blank, unique), a time period (at least one week ahead the training module creation moment, at least one week long), a location (not blank, shorter than 76 characters), an instructor (not blank, shorter than 76 characters), a mandatory contact email, and an optional link with further information.

Para cumplir con este requisito, fue necesario realizar un análisis para comprender las restricciones y los conceptos generales. Posteriormente, se completó la tarea individual de UML mencionada más adelante para ejecutarla correctamente. Se ajustaron los atributos junto con sus anotaciones después de entender completamente la magnitud del modelo.

Análisis:


Esta entidad junto al resto de entidades se añadió en la ruta “src/main/java/entities/trainingModule”, tras un análisis de las entidades a implementar, se implementó los tipos String como @NotBlank si no especificaba que era opcional y el resto de tipo de datos como @NotNull en el mismo caso, el atributo period se implemento con un periodo de comienzo (startPeriod) y un periodo final(endPeriod) de tipo “Date” y sin añadir anotaciones temporales al ser un periodo.

Los datos tipo String se implementaron con su anotación y restricciones correspondientes y se implementó una relación ManyToOne unidireccional de TrainingSession a TrainingModule.

• Developer DashBoard:

4) The system must handle developer dashboards with the following data: total number of training modules with an update moment; total number of training sessions with a link; average, deviation, minimum, and maximum time of the training modules.

Para cumplir con este requisito, fue necesario realizar un análisis para comprender las restricciones y los conceptos generales. Posteriormente, se completó la tarea individual de UML mencionada más adelante para ejecutarla correctamente. Se ajustaron los atributos junto con sus anotaciones después de entender completamente la magnitud del modelo.

	<div>Diseño y Pruebas 2 Documentación de la entrega D02</div> <div>Informe de Análisis Individual</div>
---	---

Análisis:

Esta entidad junto al resto de entidades se añadió en la ruta “src/main/java/entities/forms”, tras un análisis de las entidades a implementar, se implementó como tipo int primitivo los atributos para el total de updates y sesiones con link, y para la media, desviación, mínimo y máximo del tiempo se implementó en datatypes una clase auxiliar “Statistics” para simplificar código y se implemento en la clase principal como un “Statistics” de “TrainingModulesTime”.

Testing requirements

5)Produce assorted sample data to test your application informally. The data must include two developer accounts with credentials “developer1/developer1” and “developer2/developer2”.


Este requisito es explícito en cuanto a sus exigencias, por lo que no ha sido preciso llevar a cabo un análisis muy detallado.

Análisis:

Simplemente se añadió en el directorio “src/main/webapp/WEB-INF/resources/sample-data” una población para todas las entidades creadas (TrainingModule, TrainingSession y Developer) junto con la creación del usuario en el csv “user-account”.

Para la población, los rangos que se tomaron fueron los que se mencionan en el foro de la asignatura:

(https://ev.us.es/ultra/courses/_85092_1/cl/outline)



RAFAEL CORCHUELO GIL

RE: [Análisis] Rangos de atributos no especificados

Estimado Rafael:

Gracias por seguir los consejos que les proporcionamos a la hora de plantear sus preguntas.

Comenta Ud. sobre un problema que es tremendamente habitual en los documentos de elicitación de requisitos: los límites de los tipos de datos.

Es muy habitual que cuando algún atributo de una entidad tiene un límite claro dependiente del negocio de su cliente dicho límite se haga constar explícitamente. Por ejemplo: "el período de ejecución de algo debe comenzar al menos una semana después de la creación o la mínima duración de dicho período debe ser de un día" o "no se permiten más cinco quejas por cliente en un mismo día".

Pero existen otros muchos límites en los que habitualmente nadie piensa hasta que nos enfrentamos (de forma metódica) a crear datos de prueba. Por ejemplo: no es habitual encontrar escritos los límites para las cantidades de dinero (salvo quizá el inferior), para los momentos (salvo las restricciones habituales con respecto a los momentos de creación), o longitudes de emails o URLs.

Tampoco es habitual que se hagan explícitos los rangos de algunos tipos de datos que se entienden son lo suficientemente conocidos como para no generar dudas. Por ejemplo, si nos comentan que algo es un porcentaje queda claro que es un número en el intervalo [0.00 – 1.00] (quizá [0.00 – 100.00], o si nos comentan que algo es un ratio entonces queda totalmente claro que es un número en el intervalo [0.00 – 1.00]; la única duda podría ser en relación al número de decimales que debemos admitir.

En estos casos, lo mejor es hacer una propuesta razonable al cliente, que habitualmente no tendrá mucho inconveniente dado que esto es un detalle de implementación que poco tiene que ver con su negocio. Simplemente tenemos que confirmar que los rangos elegidos por el equipo de desarrollo son suficientemente grandes, nada más.

Por lo tanto, la alternativa más adecuada es la A1. La alternativa A2 no es muy buena porque al dejar los intervalos abiertos no podemos hacer pruebas sistemáticas. De forma general, le propongo lo siguiente:

- Cantidades de dinero: de -1.000.000,00 XXX a +1.000.000,00 XXX, donde XXX es una unidad monetaria aceptada por el sistema.
- Momentos: de 01/01/2000 00:00 a 31/12/2200 23:59.
- Emails o URLs: de 0 a 255 caracteres.

Por desgracia, no tenemos ninguna forma sencilla de fijar estos rangos mediante anotaciones estándar, por lo que tenemos que implementarlos como restricciones custom en los correspondientes servicios.

Como ingenieros de software debemos estar siempre atentos a las ideas que nos permitan mejorar nuestras herramientas de trabajo; cada mejora en dichas herramientas debe ir orientada a reducir los costes de desarrollo, decrementar las posibilidades de cometer errores y, en definitiva, mejorar la calidad de nuestros productos.

En este caso particular, está claro que sería interesante investigar de qué forma podríamos introducir esas restricciones implícitas dentro del framework. Pero no durante el desarrollo de un proyecto. Durante el desarrollo de un proyecto tenemos que damos cuentas de los problemas, es al final de los mismos cuando debemos recapitular y tomar decisiones sobre cómo mejorar nuestro framework de desarrollo para el siguiente proyecto.


Saludos. RC

Momentos : de 01/01/2000 00:00 a 31/12/2200 23:59

Urls : de 0 a 255

emails: de 0 a 73 (está restringido por la misma implementación @Email)

String: de 0 a 255

	Diseño y Pruebas 2 Documentación de la entrega D02
	Informe de Análisis Individual

Se han probado dichos datos extremos y valores intermedios en todas las poblaciones.

En caso de si tiene restricciones de tipo de longitud o temporales, se han probado los rangos extremos con datos validos con la restricción mencionada (si length= 75 y NotBlank de 1 a 75 por ejemplo) o si es temporal como un @past hasta el tiempo base de eclipse (01/01/2000 00:00 a 2022/07/29 23:59) pues el present en el proyecto es el 2022/07/30 00:00.

También se ha probado en no poner los valores en aquellos atributos que eran opcionales y diferentes idiomas para los tipos String.

4.2 SUPPLEMENTARY Deliverable D02: Information requirements

- Developer Role:


13)There is a new project-specific role called developer, which has the following profile data: degree (not blank, shorter than 76 characters), a specialisation (not blank, shorter than 101 characters), list of skills (not blank, shorter than 101 characters), an email, and an optional link with further information.

Para cumplir con este requisito, fue necesario realizar un análisis para comprender las restricciones y los conceptos generales. Posteriormente, se completó la tarea individual de UML mencionada más adelante para ejecutarla correctamente. Se ajustaron los atributos junto con sus anotaciones después de entender completamente la magnitud del modelo.

Análisis:

Se añadió en el directorio "src/main/java/entities/roles" una clase/entidad Developer con sus atributos correspondientes, se implementó los tipos String como @NotBlank si no especificaba que era opcional y el resto de tipo de datos como @NotNull en el mismo caso.

Los datos tipo String se implementaron con su anotación y restricciones correspondientes.

	<p>Diseño y Pruebas 2 Documentación de la entrega D02</p>
	<p>Informe de Análisis Individual</p>

Managerial requirements

- 14) Produce a UML domain model.


Este requerimiento ha implicado una investigación inicial para entender las interacciones entre las entidades, atributos y limitaciones. Asimismo, se ha requerido familiarizarse con la herramienta UMLet, empleada para crear diagramas UML.

- 16) Produce a planning and progress report

Este requisito es explícito en cuanto a lo que se pide, por lo que no se ha requerido realizar un análisis previo muy extenso.

Análisis:


Este requisito consiste en producir un informe de planificación. Para cumplir con este requerimiento, he elaborado un informe siguiendo la estructura proporcionada en las guías de la asignatura. Este documento detalla cada una de las tareas realizadas en preparación para la segunda entrega del proyecto.

	<p>Diseño y Pruebas 2 Documentación de la entrega D02</p>
	<p>Informe de Análisis Individual</p>

5. Conclusiones

Este segundo análisis ha sido más complejo que la que se realizó en la primera entrega, ya que me he enfrentado a varias ambigüedades o requisitos que no se especificaba muy bien como implementar algunas relaciones, o a la hora de popular los datos, tampoco se detallaba cual serían los rangos óptimos de prueba.

Como resalte en la primera entrega, este análisis proporciona una base sólida sobre la cual construir, estableciendo los cimientos necesarios para abordar con mayor profundidad los desafíos y requerimientos futuros del proyecto. A medida que avancemos en las etapas subsiguientes, es probable que nos enfrentemos a análisis más detallados y complejos, pero este primer paso es esencial para orientarnos en la dirección correcta.

	<p>Diseño y Pruebas 2 Documentación de la entrega D02</p>
	<p>Informe de Análisis Individual</p>

6. Bibliografía

Intencionalmente en blanco.