

DP2 2024

Acme Software Factory

Repositorio: <https://github.com/DP2-2024-C1-029/Acme-Software-Factory.git>

Miembro:

- David Godoy Fernández (davgodfer@alum.us.es)

Tutor: David Godoy Fernández
08/03/2024

GRUPO C1.029
Versión 1.0

Índice

Historial de versiones.....	3
Introducción.....	4
Contenido.....	4
Conclusiones	6
Bibliografía	6

Historial de versiones

Fecha	Versión	Descripción	Entrega
08/03/2024	V1.0	Inicio del documento	D02

Introducción

Se procede a explicar un breve contenido sobre la entrega.

Estamos trabajando con el proyecto “Acme Software Factory”, hemos implementado unos requisitos definidos por el product owner, la implementación se ha realizado a partir de una descripción detallada de los requisitos, además hemos realizado datos de prueba informales para testear la aplicación.

A continuación, se citarán las decisiones tomadas y los conflictos ocurridos, así como requisitos que se han implementado además de una conclusión.

Contenido

Creación Identidad Contract

Descripción: A contract is one or several agreements between the stakeholders involved in the development 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 instantiation moment (in the past), a provider name (not blank, shorter than 76 characters), a customer name (not blank, shorter than 76 characters), some goals (not blank, shorter than 101 characters), and a budget (less than or equal to the corresponding project cost).

Los atributos donde han tenido que tomar decisiones han sido los siguientes:

- instantiationMoment: Se nos pide implementar una fecha con la restricción se que sea una fecha anterior a la actual, para ello hemos decidido implementarla usando el tipo Date, además de usar la restricción @Past para que la fecha sea anterior a la actual. Además el nombre del atributo nos especifica que ha de ser un momento concreto de instanciación por lo que se ha decidido definir el atributo con @Temporal(TemporalType.TIMESTAMP) para que se almacene tanto la fecha como la hora.

- Budget: Este atributo nos a propuesto varias dudas a la hora de implementarlo, hemos propuesto varias alternativas en el foro de discusiones más concretamente en [Análisis] D02-Student#2-002, donde el product owner nos ha confirmado que la mejor opción es implementar el atributo haciendo uso de la clase Money.

A todo esto, mas los atributos restantes los cuales no han supuesto un problema a la hora de ser implementados hay que sumarle las relaciones implementadas en la clase, una para Project ManyToOne y otra para Client ManyToOne.

Creación Identidad PorgressLogs

Descripción: Every contract has an evolution that is composed of progress logs. The system must store the following data about them: a record id (pattern “PG-[A-Z]{1,2}-[0-9]{4}”, not blank, unique), a percentage of completeness (positive), a comment on the progress (not blank, shorter than 101 characters), a registration moment (in the past), and a responsible person for the registration (not blank, shorter than 76 characters).

En esta clase hemos tenido la misma situación que en el atributo `instantiationMoment` en la clase definida anteriormente, el problema se ha abordado de la misma forma con el uso de las validaciones `@Past` y `@Temporal(TemporalType.TIMESTAMP)`.

Además, el atributo `completeness` se ha implementado con un tipo primitivo `double` para optimizar el rendimiento. Existía cierta incertidumbre a la hora de implementar este atributo ya que al ser un porcentaje no se sabe si el rango del atributo debería comprenderse entre `[0.0-100.0]` o `[0.0-1.0]` para ellos se ha realizado una propuesta de alternativas al product owner el cual nos ha especificado que se deberá usar el rango `[0.0-1.0]`, esto se ha llevado acabo en el tablero de discusión, específicamente en [Análisis] Rangos de atributos no especificados.

Se ha implementado una relación `ManyToOne` con `contract`.

Creación dashboard para client

Descripción: The system must handle client dashboards with the following data: total number of progress logs with a completeness rate below 25%, between 25% and 50%, between 50% and 75%, and above 75%; average, deviation, minimum, and maximum budget of the contracts.

No ha surgido ningún problema a la hora de implementarse.

Creación Identidad Client

Descripción: There is a new project-specific role called clients, which has the following profile data: identification (pattern `"CLI-[0-9]{4}"`, not blank, unique), a company name (not blank, shorter than 76 characters), type (not blank, either `"company"` or `"individual"`), an email (not blank), and an optional link with further information.

El atributo `type`, se ha decidido implementar con el nombre `CompanyType`, siendo este un enum de los 2 valores especificados en el requisito, se ha tomado esta decisión ya que es un nombre que representa mejor al atributo. No se han tomado mas decisiones en este requisito.

Creación de datos informales

Descripción: Produce assorted sample data to test your application informally. The data must include two client accounts with credentials `"client1/client1"` and `"client2/client2"`.

Se han seguido las pautas definidas en la presentación `Sample Data` la cual nos especifica como estructuras los datos informales. No ha existido ningún problema a la hora de desarrollar esta tarea, aunque ha sido una tarea muy ardua.

Conclusiones

Se ha podido implementar todos los requisitos, se ha comprobado que todo se ejecuta correctamente y que la base de datos estaba poblada con los datos correspondientes.

Los conflictos a la hora de tomar decisiones acerca de los requisitos se han resuelto sin problema, debido a que se han propuesto alternativas en el foro de discusión y el product owner ha especificado la alternativa adecuada o en otros casos una solución equivalente.

Bibliografía

Intencionadamente en blanco