DP2 2024

Acme Software Factory

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

Miembros:

* David Godoy Fernández (davgodfer@alum.us.es)
* Ismael Gata Dorado (ismgatdor@alum.us.es)
* Jaime Varas Cáceres (jaivarcac@alum.us.es)
* José María Portela Huerta (josporhue@alum.us.es)
* Juan José Gómez Borrallo (juagombor@alum.us.es)

Tutor: José González Enríquez

GRUPO C1.029

26/04/2024

Versión 3.0

Índice

[Historial de versiones 3](#_Toc165062844)

[Introducción 4](#_Toc165062845)

[Contenido 4](#_Toc165062846)

[Conclusiones 7](#_Toc165062847)

# Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Entrega |
| 16/02/2024 | V1.0 | Documento inicial | D01 |
| 08/03/2024 | V2.0 | Modelado | D02 |
| 26/04/2024 | V3.0 | Implementación | D03 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introducción

A continuación, se procede a explicar un breve resumen sobre el contenido desarrollado sobre el proyecto “Acme Software Factory”.

En esta entrega se ha trabajado sobre los requisitos obligatorios de la entrega D03, los cuales son la implementación de la lógica del proyecto

Además, se van a detallar los pasos y los errores que nos hemos encontrado el modelado de datos.

# Contenido

Durante el desarrollo de este entregable nos han surgido y hemos resuelto diferentes dudas que detallaremos a continuación.

1. En este entregable hemos modificado lo relacionado con las moneas aceptadas que comentamos en el entregable anterior, adjunto imagen de captura de lo anterior

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Imagen 1.

En esta entrega hemos renombrado la clase denominándola Configuration y haciendo que sea una entidad para poder almacenar por medio de un csv las moneas permitidas y la que esta actualmente en configuración, de esta forma para los requisitos en los que se requería que solo fueran monedas aceptadas serán las únicas que se puedan. Se adjunta imagen de como queda actualmente.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Imagen 2.

1. Se ha creado una entidad Exchange la cual almacena mas de 100 monedas y su valor actual obtenidas por medio de una API la cual se llama para obtener la información de forma que si necesitamos los datos y no ha pasado mas de una hora desde la ultima vez que los necesitamos no consultara en la API y lo hará en la tabla, de esta forma evitamos que nos bloquee/expulse si se hacen demasiadas llamadas a la API.

La api seria la siguiente: https://openexchangerates.org/account/usage

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

1. Hemos modificado el UML para añadir la entidad de Exchange y Configuration.
2. Para la impelmentacion de las "notices" se ha de comentar que no ha habido ningún problema, se ha realiza de la misma manera que la implementación de los requisitos individuales. únicamente teníamos que tener cuidado con la verificación a la hora de crear una "notice", cumpliendo el requisito de que debe estar seguro para publicarlo antes de crearlo. Además, se ha tenido en cuenta y solo se mostrarán las notices que han sido creada en el último mes. También comentar, que lo que ha podido dar más problema ha sido el poder encontrar la manera de realizar el constructor para que tenga el formato que deseaba. Se ha creado correctamente y funciona según el requisito.
3. Para la implementación del dashboard ha habido algún problema para evitar mostrar los datos si no eran válidos. Sin embargo, ese problema se ha podido subsanar y se ha podido comprobar qué ocurre en caso de que no haya ninguna entidad publicada. También se ha de decir, que el Ratio está publicado en tanto por ciento, para ser más visual. Para el conteo de la cantidad de personas con roles que existe en la aplicación, se ha realizado un count de cada tabla correspondiente para poder obtenerlos.
4. El requisito:

*34) The system must prevent the principals from storing any data that can be considered spam. A piece of text is considered spam if the number of spam terms exceeds a predefined spam threshold. The default list of spam terms includes the following ones: “sex”, “viagra”, “cialis”, “one million”, “you’ve won”, “nigeria”, and their corresponding translations into the languages considered for internationalisation; the default spam threshold is 10%. Note that the previous default values can be changed at will by the administrators. Realise that a term must be considered spam irrespective of its case and the blanks in between its words; for instance, “one␣million” is a spam term that matches “one␣million”, “ONE␣MILLION”, “OnE␣␣␣MiLLiOn”, or “One␣⏎␣Million”; it doesn’t match “One␣Millionaire”, “One␣or␣two␣millions”, or “One␣sexy␣million”, though. The spam detector must be reusable across different projects; that is: it must be implemented as an independent project that must be packaged into a reusable dependency. Do not forget to deliver your spam detector project so that it can also be evaluated or, otherwise, this requirement shall not be considered valid. This requirement must be fulfilled in this and every subsequent group or individual deliverable for it to be considered valid.*

No hemos podido implementarlo y lo tendremos en cuenta para la siguiente entrega.

# Conclusiones

Como conclusión se han realizado algunos cambios de lo que había y se ha realizado un manejo de las monedas para el cambio.