



CHANGELOG D06

10/01/2019

Registro de cambios

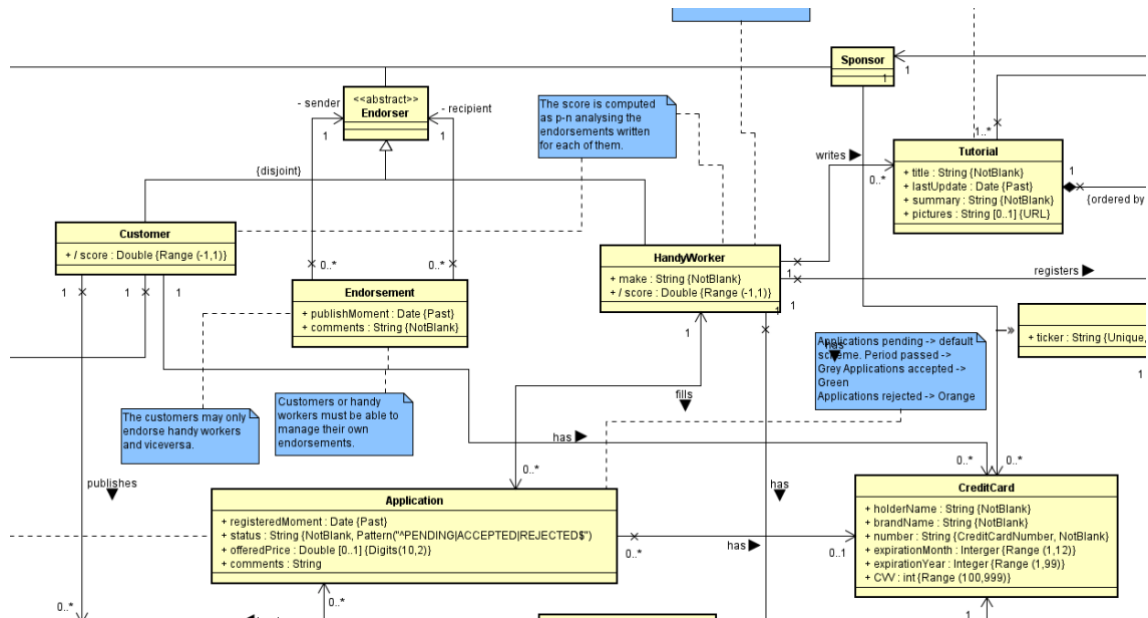
Registro de cambios respecto a la D05 del proyecto.

D&T Grupo 11

Albalat Heredia, Carlos
del Carmen Fuentes, Lucía
Martínez Figueroa, Pablo
Montaño Aguilera, Antonio Manuel
de Paz Galán, Francisco
Rico Catalán, Jesús Andrés

1. Cambios en los modelos de dominio

- Se ha añadido la relación Customer → CreditCard y Sponsor → CreditCard con la finalidad de conocer el propietario específico de una tarjeta de crédito.



- Se ha añadido en los modelos UML la bidireccionalidad Note <-> Report, que ya estaba puesta anteriormente en el modelo de dominio Java.
- Eliminados los cascade = CascadeType.ALL de Message y MessageBox ya que no es una composición.
- Actor::getSocialProfiles() se ha eliminado cascade=CascadeType.ALL ya que no es una composición.
- HandyWorker::getCurriculum(): se ha eliminado la etiqueta @NotNull ya que un currículum no es obligatorio.

2. Cambios en los repositorios

- **ComplaintRepository:** se ha añadido una consulta que devuelve las quejas a partir de un handy worker.

3. Cambios en los servicios

- En `WarrantyService::save()` ahora se comprueba si la garantía está en modo final o no, ya que, de estarlo, no puede ser modificada.
- `PhaseService::save()`: ahora las fases solo pueden crearse cuando una solicitud esté aceptada, tal y como exigen los requisitos. También se cumplen ahora las restricciones de tiempo respecto a su chapuza correspondiente.
- `PhaseService::delete()`: anteriormente no borraba correctamente la fase, ahora sí lo hace.
- `AdministratorService`: se ha añadido un método que calcula la puntuación de un cliente o un manitas para cumplir el requisito 50.1. También se ha añadido un método que calcula las puntuaciones de todos ellos que servirá para mostrar una vista con todas las puntuaciones de todos los clientes y manitas.
- `FinderService`: se ha modificado el método de búsqueda que devuelve un nuevo buscador que incluye las chapuzas que cumplen los filtros indicados.
- Se han modificado los test y se elimina la búsqueda de un elemento por su identificador, para que funcionen siempre a pesar de que se vuelva a poblar la base de datos.
- `ComplaintService::save()`: cuando se crea una queja, esta se añade a la lista de quejas del cliente en cuestión con la finalidad de tener siempre actualizada dicha lista.
- `FixUpTaskService::save()`: se ha añadido un nuevo método que devuelve al cliente creador de esa chapuza que se utilizará en diferentes controladores. También se ha añadido en este método la creación del momento de publicación y del ticker, en lugar de en el método `create()`, para que se guarde la fecha de guardado y no de creación y porque de esta forma aseguramos que el ticker sea único. Por otra parte, se ha añadido la nueva forma de comprobar si una chapuza contiene spam.
- `CurriculumService`: se ha eliminado el método `delete()` ya que al considerarse que el currículum es opcional, no debería ser borrado.
- `PersonalRecordService`: se ha eliminado el método `delete()` ya que es obligatorio en un currículum y se ha considerado que este no puede ser borrado.

- `UtilityService::generateTicker()`: se ha modificado el método ya que con anterioridad daba fallos al obtener el día y el mes correspondiente.
- `NoteService::save()`: se ha modificado el método diferenciando el rol del creador de la nota utilizando el operador `instanceof`. De esta forma, dependiendo de rol de la persona autenticada, hacemos las comprobaciones necesarias para posteriormente guardar la nota. También se ha añadido la nueva forma de comprobar si la nota contiene spam.
- `NoteService::delete()`: se ha eliminado el método ya que una nota no debe ser borrada una vez ha sido creada.
- Se han añadido en varios servicios comprobaciones para que, al guardar una entidad, no se actualicen sus atributos fijos.
- `ApplicationService::save()`: se ha añadido la nueva forma de comprobar si una solicitud contiene spam.
- `MessageBoxService::save()`: se ha añadido la nueva forma de comprobar si un mensaje de una carpeta contiene spam.
- `ReportService`: se ha añadido la nueva forma de comprobar si un informe contiene spam.