



Analysis report

C1.034

<https://github.com/juanlo017/c1.034>

Apellidos, Nombre	Email
López Quirós, Juan	jualopqui1@alum.us.es

Tabla de contenidos

Resumen ejecutivo	3
Tabla de revisiones	4
Introducción	5
Contenidos	6
Registros de análisis	6
Conclusiones	7
Bibliografía	8

Resumen ejecutivo

Con este análisis buscaremos imprecisiones, incongruencias e indeterminaciones en los requisitos individuales solicitados en la segunda entrega de la asignatura al Student #1. La finalidad será describir los mismos con la mayor precisión posible, buscando optimizar la satisfacción del cliente.

Tabla de revisiones

Versión	Descripción	Fecha
v1.0	Creación del documento	19/02/2024

Introducción

Se ha procedido a analizar los requisitos individuales del Student #1 en la segunda entrega de la asignatura en el cual nos hemos encontrado un error en la clasificación de prioridades de requisitos y nos hemos topado con algún requisito algo ambiguo, el cual tendremos que consultar con el cliente para ver qué es lo que espera del sistema..

Contenidos

Registros de análisis

Requisito	Conclusiones	Decisiones
A project aggregates several user stories elicited by the same manager .	<p>Para satisfacer el requisito del cliente debemos crear una entidad persistente con los siguientes atributos:</p> <p>String: code String: title String: abstractText boolean: fatalErrors double: cost String: link (URL restrictions) Manager: manager (ManyToOne optional=false, Valid, Not Null)</p>	<p>Title será un String con las restricciones solicitadas.</p> <p>Abstract será nombrado abstractText ya que abstract es una palabra reservada de Java y será un String con las restricciones solicitadas.</p> <p>fatalErrors llamaremos a indication, para hacerlo más descriptivo, será de tipo boolean para denotar si hay fallo grave o no, usamos el primitivo ya que no puede ser nulo. No es transient ya que el cliente ha aclarado en el foro que es necesario guardarlo en base de datos. Cito: "Guardar la indicación. Correcto, es necesario teniendo en cuenta lo que nos ha aclarado el cliente". Enlace: https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board&conf_id=_405265_1&forum_id=_234042_1&message_id=_403856_1</p> <p>Cost será un atributo tipo int obligatorio positiveOrZero, derivado de la suma de las user stories asociadas al proyecto. Se consideró la posibilidad de tomar a cost como un tipo money para representar el coste monetario del proyecto, pero tras revisar el foro, se aconsejó que el coste del proyecto debe estar medido en horas enteras resultante de la suma de los user story asociados del conglomerado. enlace: https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board&conf_id=_405265_1&forum_id=_234042_1&message_id=_404105_1 literalmente: "...el coste estimado se mide en horas, por lo que tiene sentido pensar que el coste del proyecto en su conjunto también se mida</p>

		<p>en horas y que sea realmente la suma de las horas correspondientes a las historias de usuario que forman el conglomerado.”</p> <p>Link será un String con restricciones de tipo URL pero no restricciones de Not Null ya que será opcional. Máximo 255 ya que hibernate guarda por defecto los string como varchar 255 y debemos asegurarnos de que no lo sobrepasa para evitar una excepción en el sistema.</p> <p>Project tendrá una relación many-to-many con user story ya que el cliente quiere poder reutilizar las mismas historias de usuario a distintos proyectos. Para ello implementaremos una clase intermedia que relacione project con user story para evitar el many-to-many desaconsejado en clase.</p> <p>draftMode-> Para dar el proyecto como “terminado” debe haber al menos un user story. Como es un conglomerado es necesario este atributo.</p> <p>Manager -> Tiene una relación Many To One optional=false, Valid y NotNull. Optional false comprueba a nivel base de datos y not null a nivel de aplicacion. El valid comprueba que es un manager con atributos correctos.</p>
<p>A user story is a document that a manager uses to represent the smallest unit of work in a project. (MANDATORY)</p>	<p>Para satisfacer el requisito del cliente debemos crear una entidad persistente con los siguientes atributos:</p> <p>String: title String: description double: estimated cost String:acceptance criteria Priority:priority String:link Manager: manager</p>	<p>Title será un String con las restricciones solicitadas.</p> <p>Description será un String con las restricciones solicitadas.</p> <p>Estimated cost será un tipo int que representará el número de horas requeridas para cada tarea. Investigué sobre la posibilidad de usar un tipo double pero en el foro se recomendó que las historias de usuario se deberían modelar como un integer, esto es debido a que 2,3 horas no sabemos si se refiere a 2h 30min o 2.3 horas, es decir 2h 18min. Textualmente: “tiene claro <i>[el cliente]</i> que mide el esfuerzo en horas enteras” enlace: https://ev.us.es/webapps/discussionboard/do/message?action=list_messag</p>

	(ManyToOne optional=false, Valid, Not Null	<p>es&course_id= 85092_1&nav=discussion_board&conf_id= 405265_1&forum_id= 234042_1&message_id= 404105_1</p> <p>Acceptance criteria será un String con las restricciones solicitadas.</p> <p>Priority será un tipo enumerado custom del mismo nombre para albergar las 4 posibilidades que necesita el cliente.</p> <p>Link será un String con restricciones de tipo URL pero no restricciones de Not Null ya que será opcional. Máximo 255 ya que hibernate guarda por defecto los string como varchar 255 y debemos asegurarnos de que no lo sobrepasa para evitar una excepción en el sistema</p> <p>User story tendrá una relación many-to-many con project ya que el cliente quiere poder reutilizar las mismas historias de usuario a distintos proyectos. Para ello implementaremos una clase intermedia que relacione project con user story para evitar el many-to-many desaconsejado en clase.</p> <p>Manager -> Tiene una relación Many To One optional=false, Valid y NotNull. Optional false comprueba a nivel base de datos y not null a nivel de aplicacion. El valid comprueba que es un manager con atributos correctos.</p>
The system must handle manager dashboards (MANDATORY)	El dashboard no se debe almacenar en base de datos y por lo tanto debe ser un form a diferencia de los anteriores requisitos que se modelan con un entity.	No hay muchas dudas se creará una clase con los atributos correspondientes tipo int, y double para aquellas que sean desviaciones o medias.

Produce assorted sample data to test your application informally. (MANDATORY)	Habr� que crear un conjunto de datos variados que comprueben bien los casos l�mites de cada atributo.	-
There is a new project-specific role called manager (SUPPLEMENTARY) (MANDATORY)	<p>Este requisito aparece como supplementary, pero no deber� serlo ya que todas las funcionalidades y entidades obligatorias del entregable dependen de que el rol de Manager exista.</p> <p>Por lo tanto consideraremos este requisito como obligatorio tambi�n y tendremos que crear un nuevo Rol con los siguientes atributos:</p> <p>String: degree String: overview String: certifications String: link</p>	<p>Degree ser� un String con las restricciones requeridas</p> <p>Overview ser� un String con las restricciones requeridas</p> <p>Certifications ser� un String con las restricciones requeridas</p> <p>Link ser� un String con restricciones de tipo URL pero no restricciones de Not Null ya que ser� opcional. M�ximo 255 ya que hibernate guarda por defecto los string como varchar 255 y debemos asegurarnos de que no lo sobrepasa para evitar una excepci�n en el sistema</p>

Produce a UML domain model (SUPPLEMENTARY)	Producir el modelo UML de las entidades anteriores. Debemos modelar la relación de project-user story como un agregado One-to-Many unidireccional desde user story a project	Hemos de introducir una tabla intermedia para modelar el many to many como varios many to one.
Produce an analysis report. (SUPPLEMENTARY)	El requisito no presenta ninguna imprecisión en su descripción y es fácil de comprender.	Implementar el requisito siguiendo las guías de documentación de la asignatura.
Produce a planning and progress report. (SUPPLEMENTARY)	El requisito no presenta ninguna imprecisión en su descripción y es fácil de comprender.	Implementar el requisito siguiendo las guías de documentación de la asignatura.

Conclusiones

En esta entrega se han encontrado alguna incompletitud en la definición de los requisitos individuales del Student #1 e incluso un fallo grave de categorización de prioridad de un requisito. A pesar de ello, gracias al análisis hemos podido identificarlos y enmendarlos consultando al cliente sobre estos.

Bibliografía

Intencionalmente en blanco.