

Estimado estudiante:

He tomado nota de su grupo y de su Hackathon. Por favor, es importante que en todas sus comunicaciones nos indique el número de su grupo y al menos los apellidos de los integrantes; en EV no podemos localizar esta información sabiendo tan sólo su email y este año, además, hemos tenido innumerables problemas para recuperar los grupos y sus entregables.

En relación con su propuesta de Hackathon, en general la veo bien, por lo que la apunto como ACEPTADA. Pero veo conveniente que tenga Ud. en cuenta los siguientes consejos de cara a la versión final del mismo.

- La solución que aportan para que el mensaje de bienvenida se pueda adaptar al lenguaje en que se está usando la aplicación no es adecuada. Tan sólo contemplan la posibilidad de inglés o español y añadir un nuevo lenguaje implica cambiar su código fuente (Clase Configuration), la estructura de su base de datos (tabla configuration), y, además, tienen Udes. que implementar en la vista correspondiente a la página de bienvenida la lógica necesaria para poder mostrar en mensaje en uno u otro lenguaje ... o lo que es peor, en un nuevo lenguaje que se añada al sistema. Este es un problema que llevamos viendo desde el primero proyecto de la asignatura y ha llegado el momento de darle una solución mejor, más sencilla, que utilice mejor los recursos de internacionalización que proporciona Spring. No es un problema sencillo de resolver, por lo que si quieren atacarlo perfectamente les podría valer como A++. Mi consejo es que al menos le dediquen un tiempo a intentar resolverlo; no es EL PROBLEMA, pero sí que es un problema al que antes o después se enfrentarán en su vida profesional y es importante que diseñen una buena solución (con su análisis de alternativas, pros y contras)
- Modelan las tarjetas de crédito como un datatype, lo que está bien desde un punto de vista conceptual... pero no siempre es una buena solución desde el punto de vista del modelo de dominio (problemas con tarjetas de crédito que son opcionales). Imagino que a día de hoy ya deben tener una solución clara para este problema.
- El sistema de mensajería que plantean es nuevo. Mi sugerencia es que se planteen como objetivo reutilizar el que ya diseñaron en algún proyecto anterior. Esto les permitirá enfrentarse a problemas muy comunes en sus futuros puestos de trabajo. Mi sugerencia es que aprovechen para reutilizar, mejorar el diseño para que sea más sencillo de incorporar en un nuevo proyecto, y mejorar (por ejemplo con el flag de leído que han incorporado en este proyecto).
- El modelado de las palabras de spam no me queda claro. Conceptualmente lo que proponen puede ser una solución, pero desde el punto de vista del modelo de dominio mi sugerencia es que exploren una alternativa: meterlas todas en la clase Configuration como un atributo multi-evaluado que después se implementará como una simple cadena en la que los spam words se separan por comas, por ejemplo. Por favor, estudien los pros y contras de ambas alternativas.
- Identificar a los drivers/mechanics/sponsors mediante su DNI es una solución que tan sólo funcionará en España. Creo que es una buena idea que le den un vistazo a cómo funcionan sistemas como UBER, Grab o Cabify y que mejoren el sistema de identificación de los drivers. No les pido un sistema como el que tiene UBER que realiza un reconocimiento del documento de identificación, pero sí que piensen en algo más

general que el DNI, que es la forma de indentificación tan sólo en España. En todos los países hay alguna forma de identificación de los ciudadanos, pero no tienen por qué ser DNIs. Si quieren hacerlo un poco más realista, mi sugerencia es que permitan al administrador guardar en la configuración del sistema un conjunto de patrones de validación de IDs (que sería el nombre genérico de un documento de identificación), de forma que cuando un driver se da de alta en el sistema indica su nacionalidad y entonces el sistema busca en la configuración el patrón para validar su ID. Fíjense en que la misma técnica se puede usar para validar el teléfono. Esto les obligará a investigar un poco más el mecanismo de validación que proporciona Spring, por lo que podría valer como un A++. En su puesto de trabajo, realizar validaciones que dependerán del administrador va a ser algo muy frecuente, puesto que a día de hoy es muy normal que casi todas las empresas desarrollen sistemas multi-tenant (sistemas que se configuran con facilidad para ser desplegados con clientes diferentes) y cada cliente (tenant) generalmente tiene unos patrones de validación diferentes. Por ejemplo, piense en un típico sistema de comercio electrónico; todos los vendedores tienen algún tipo de ID, SKU o Ticker para identificar los productos, pero cada uno tiene una estructura diferente y debe ser validada.

- La parte del modelo relacionada con los comentarios y las respuestas a los mismos no la veo clara. Por favor, consideren un modelo más sencillo en el que hay una única clase Comment con una asociación reflexiva de tipo 0:* -- 0:1 que indica si un comentario es respuesta a otro. Por favor, analicen los pros y contras de cada alternativa antes de tomar una decisión final.
- En su modelo actual hacen una distinción explícita entre DriverReview y UserReview. Es una solución, pero anticipo que van a tener que repetir mucho código. Por qué no piensan en algo más reutilizable, más fácil de adaptar, algo que les permita poner revisiones a casi cualquier entidad de su sistema. Por qué no piensan en modelar el concepto de Reviewable como una interfaz, de manera que un Review hace referencia a un Reviewable, que puede ser un Driver, un User... lo que sea necesario en cada caso.
- En relación con su A+, veo perfecto integrar Google maps con el objeto de dar una estimación de precio realista. Lo que no veo claro en sus requisitos es este detalle. Por favor, denle un nuevo vistazo a los requisitos en que hablan sobre este tema y mejoren la redacción de forma que quede lo más claro posible cómo se hace el cálculo y que en su modelo conceptual/dominio estén todas las entidades/atributos necesarios para poder hacerlo.
- Por favor, revisen el requisito no funcional 25. No se requieren que cumplan con todas las regulaciones puesto que algunas de ellas les requerirían darse de alta en la Cámara de Comercio. Por favor, revisen el enunciado de cualquiera de los proyectos que les hemos facilitado a lo largo de la asignatura y redacten el requisito de una forma parecida.

Saludos. RC