

DP2 2024

Acme Software Factory

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

Miembro:

- Ismael Gata Dorado (ismgatdor@alum.us.es)

Tutor: José González Enríquez
27/05/2024

GRUPO C1.029
Versión 4.0

Índice

Historial de versiones.....	3
Resumen Ejecutivo.....	4
Introducción.....	4
Contenido.....	4
Conclusiones	6
Bibliografía	7

Historial de versiones

Fecha	Versión	Descripción	Entrega
16/02/2024	V1.0	Inicio del documento	D01
08/03/2024	V2.0	Modelado	D02
26/04/2024	V3.0	Implementación del sistema	D03
27/04/2024	V4.0	Testing	D04

Resumen Ejecutivo

Como alumno de la asignatura Diseño y Pruebas II describo todo lo que he realizado durante esta tercera entrega detallando los análisis, dudas y soluciones para el diseño del modelado del sistema.

Introducción

En este documento se recogerá por parte del alumno un resumen del análisis o desarrollo que se ha realizado durante la tercera entrega del proyecto para la asignatura Diseño y Pruebas II.

Contenido

Durante el desarrollo de esta cuarta y última he ido corrigiendo errores encontrado e implementado cierto desarrollo que faltaba.

A continuación, describiré cada acción realizada:

- Monedas y cambio:

Quiero destacar este punto ya que en la entrega anterior no tenía internacionaliza la moneda al mostrar el detalle del proyecto, en esta ocasión si lo he añadido.

Además, ya que mi interpretación con respecto al dashboard y las monedas había sido otra diferente, en esta entrega he añadido para que se puedan visualizar la información en las diferentes monedas aceptadas del sistema. Por ello he tenido que rehacer el servicio de `AuthenticatedExchangeService`, para mejorar y llamar menos a la Api de lo que se hacía, así como la posibilidad de querer obtener solo las cantidades en la moneda configurada del sistema o en todas las aceptadas de forma que lo hace compatible para los desarrollos de los tres compañeros que tenemos monedas, además de esto y a ultima hora al detectar que los test que había grabado estaban dando problema debido a que al llamar a la api por primera vez y cargar los datos los índices de los valores van fluctuando en el tiempo por lo que encontré en el foro a un compañero que le estaba pasando exactamente lo mismo y en donde se sugiere realiza run Helper que ayuda a devolver los valores reales o simulados dependiendo de en donde se este ejecutando el sistema (develop, tester), en el documento de Testing report he documentado más acerca de estoy y las clases involucradas.

- Traducciones:

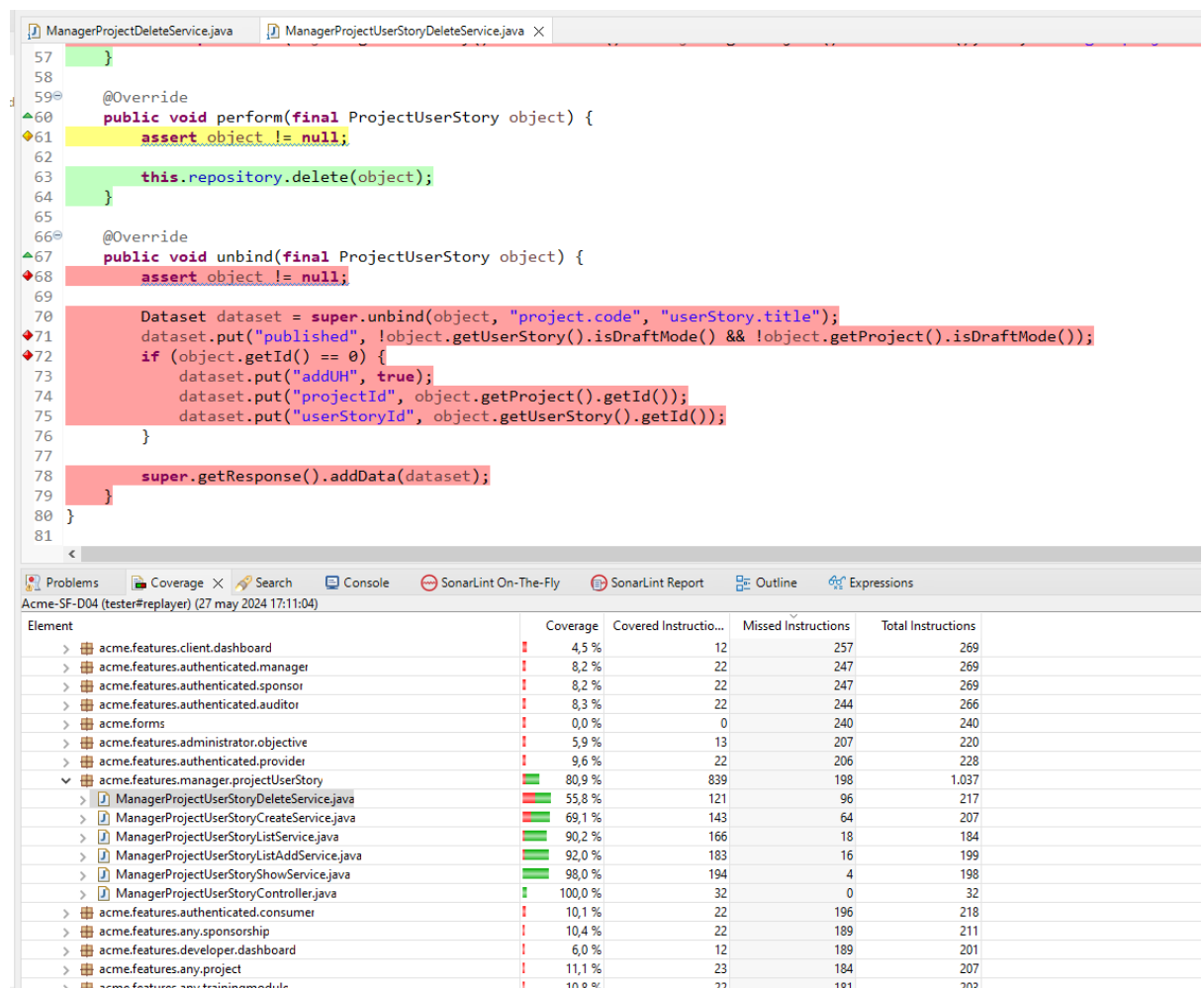
He detectado que faltaban alguna que otra traducción o que estaba mal traducido, por lo que se han corregido.

- Crear proyecto/historia de usuario:

Al realizar las pruebas (también se comenta en el documento de Testing report) se había detectado que si se intentaba hacer un hack al crear una de estas entidades y se le modificaba el id desde la consola del navegador, el sistema actualizaba el registro con ese id y no creaba un nuevo, por lo que se ha solventado este problema setando el Id a 0 justo antes de guardar.

- Actualizar proyecto:
Gracias a las pruebas también se ha podido detectar que la aplicación mostraba un error 500 al actualizar un proyecto con un código existente en el sistema, por lo que se ha solventado este error.
- Cobertura de código:
Destacare las más llamativas:

En este caso y unos cuantos mas obtengo que el unbind no se ejecuta nunca aun con las validaciones y restricciones, por todos los problemas que he ido teniendo con la ejecución de test y puesto que en el proyecto de referencia Acme-Job también están puestos, no los he eliminado y por ello baja la cobertura.

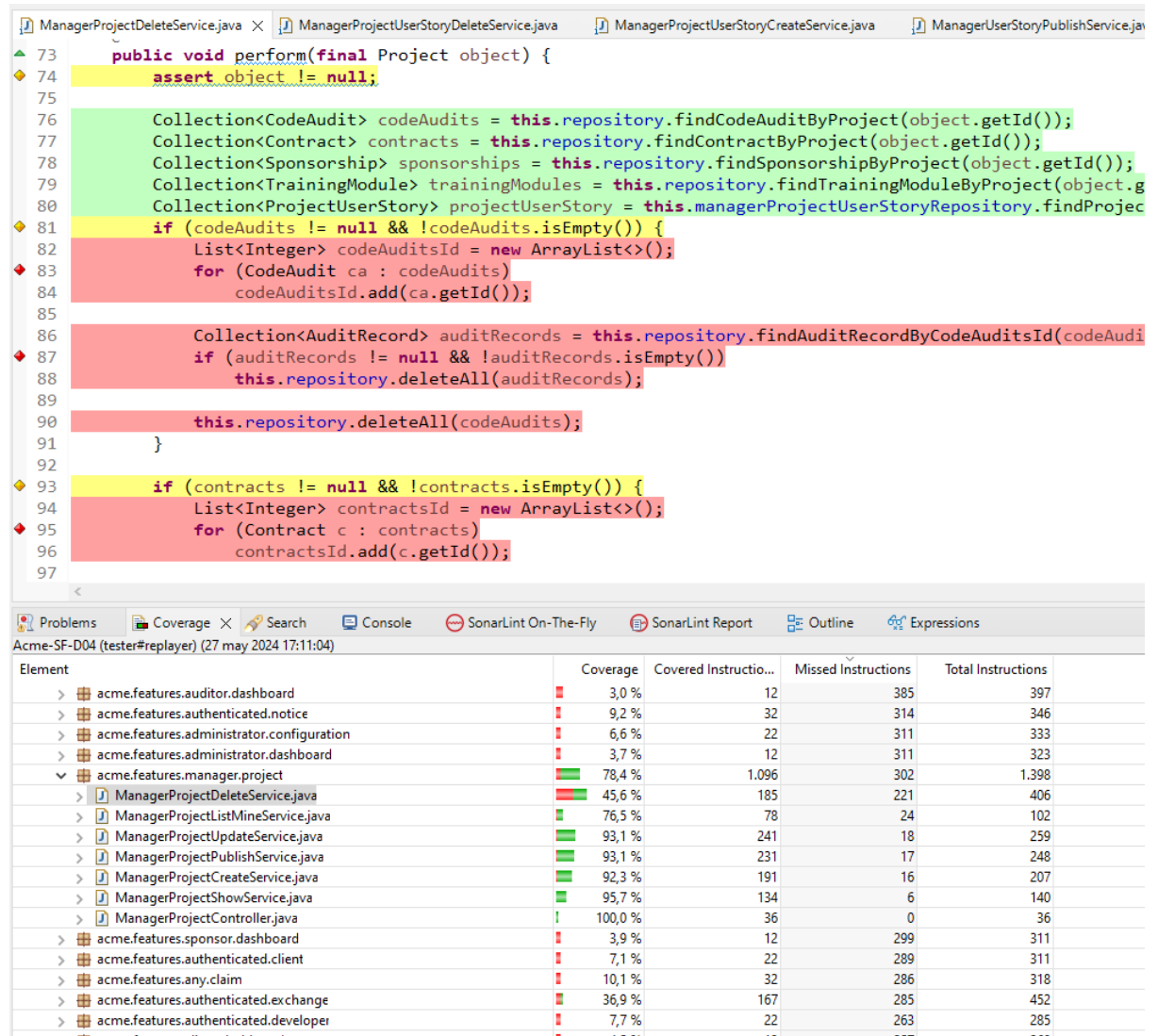


The screenshot displays a Java IDE with two tabs: `ManagerProjectDeleteService.java` and `ManagerProjectUserStoryDeleteService.java`. The code in the active tab shows a `perform` method that calls `delete` and an `unbind` method that calls `super.unbind` and updates a dataset. The `unbind` method includes a conditional check for `object.getId() == 0`.

Below the code editor, a coverage report is visible for the project `Acme-SF-D04 (tester#replayer)` as of 27 May 2024 17:11:04. The report lists various elements and their coverage percentages, with some elements showing 0.0% coverage.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
acme.features.client.dashboard	4,5 %	12	257	269
acme.features.authenticated.manager	8,2 %	22	247	269
acme.features.authenticated.sponsor	8,2 %	22	247	269
acme.features.authenticated.auditor	8,3 %	22	244	266
acme.forms	0,0 %	0	240	240
acme.features.administrator.objective	5,9 %	13	207	220
acme.features.authenticated.provider	9,6 %	22	206	228
acme.features.manager.projectUserStory	80,9 %	839	198	1.037
ManagerProjectUserStoryDeleteService.java	55,8 %	121	96	217
ManagerProjectUserStoryCreateService.java	69,1 %	143	64	207
ManagerProjectUserStoryListService.java	90,2 %	166	18	184
ManagerProjectUserStoryListAddService.java	92,0 %	183	16	199
ManagerProjectUserStoryShowService.java	98,0 %	194	4	198
ManagerProjectUserStoryController.java	100,0 %	32	0	32
acme.features.authenticated.consumer	10,1 %	22	196	218
acme.features.any.sponsorship	10,4 %	22	189	211
acme.features.developer.dashboard	6,0 %	12	189	201
acme.features.any.project	11,1 %	23	184	207
acme.features.any.trainingmodule	10,8 %	22	181	203

En este caso mas especial la parte del código que no se esta ejecutando es debido a que para mis pruebas individuales no se ha testado los test del resto de los estudiantes por tanto a priori no se ejecuta, en el momento que se ejecuten todos los test al completo es cuando este código debería aumentar su cobertura.



```

73 public void perform(final Project object) {
74     assert object != null;
75
76     Collection<CodeAudit> codeAudits = this.repository.findCodeAuditByProject(object.getId());
77     Collection<Contract> contracts = this.repository.findContractByProject(object.getId());
78     Collection<Sponsorship> sponsorships = this.repository.findSponsorshipByProject(object.getId());
79     Collection<TrainingModule> trainingModules = this.repository.findTrainingModuleByProject(object.getId());
80     Collection<ProjectUserStory> projectUserStory = this.managerProjectUserStoryRepository.findProjectUserStoryByProject(object.getId());
81     if (codeAudits != null && !codeAudits.isEmpty()) {
82         List<Integer> codeAuditsId = new ArrayList<>();
83         for (CodeAudit ca : codeAudits)
84             codeAuditsId.add(ca.getId());
85
86         Collection<AuditRecord> auditRecords = this.repository.findAuditRecordByCodeAuditsId(codeAuditsId);
87         if (auditRecords != null && !auditRecords.isEmpty())
88             this.repository.deleteAll(auditRecords);
89
90         this.repository.deleteAll(codeAudits);
91     }
92
93     if (contracts != null && !contracts.isEmpty()) {
94         List<Integer> contractsId = new ArrayList<>();
95         for (Contract c : contracts)
96             contractsId.add(c.getId());
97     }

```

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
> acme.features.auditor.dashboard	3,0 %	12	385	397
> acme.features.authenticated.notice	9,2 %	32	314	346
> acme.features.administrator.configuration	6,6 %	22	311	333
> acme.features.administrator.dashboard	3,7 %	12	311	323
> acme.features.manager.project	78,4 %	1.096	302	1.398
> ManagerProjectDeleteService.java	45,6 %	185	221	406
> ManagerProjectListMineService.java	76,5 %	78	24	102
> ManagerProjectUpdateService.java	93,1 %	241	18	259
> ManagerProjectPublishService.java	93,1 %	231	17	248
> ManagerProjectCreateService.java	92,3 %	191	16	207
> ManagerProjectShowService.java	95,7 %	134	6	140
> ManagerProjectController.java	100,0 %	36	0	36
> acme.features.sponsor.dashboard	3,9 %	12	299	311
> acme.features.authenticated.client	7,1 %	22	289	311
> acme.features.any.claim	10,1 %	32	286	318
> acme.features.authenticated.exchange	36,9 %	167	285	452
> acme.features.authenticated.developer	7,7 %	22	263	285

Finalmente hay clases en las que me aparece la cobertura en amarillo debido a que sería posible refactorizar y tendrá validaciones o restricciones de mas que no se ejecuta en todas las posibilidades, pero como he comentado al principio no he llegado a modificarlo porque me supondría rehacer todos los test de nuevo con el tiempo que conlleva.

Conclusiones

Como conclusión se podría decir que ha sido una entrega de correcciones y testing dando una aplicación más robusta y segura.

Bibliografía

Intencionadamente en blanco