

TESTING REPORT



Santos Martín, Javier
Ruíz Garrido, Javier
Jiménez Morales, Francisco Miguel
García de Tejada Delgado, José
Antonio Daniel Porcar Aragón

General project information






PROJECT NAME			Acme-SF	
PARTICIPANTS				
Name	Email	Role	Username	Photo
Antonio Daniel Porcar Aragón	antporara@alum.us.es	Project Manager, Developer	antporara	
Francisco Miguel Jiménez Morales	frajimmor2@alum.us.es	Tester, Developer	frajimmor2	
Javier Santos Martín	javsanmar5@alum.us.es	Developer, Secretary	javsanmar5	
Javier Ruiz Garrido	javruigar2@alum.us.es	Analyst, Developer	Javiruizg	
José García de Tejada Delgado	josgardel8@alum.us.es	Operator, Developer	JoseGTD	
Stakeholders				
Francisco Miguel Jiménez Morales, Javier Ruiz Garrido, José García de Tejada Delgado, Javier Santos Martín, Antonio Daniel Porcar Aragón and José González Enriquez (the professor).				
Start date	Expected date of completion	Deliverables	Date of Document	
12/02/2024	27/05/2024	4	14/02/2024	

Table of contents

REPORT NAME	1
General project information	2
Table of contents	3
Executive summary	4
Revision table	5
Introduction	6
Contents	7
Conclusions:	8
Bibliography	9

Executive summary

En este informe se detalla qué pruebas se han implementado, y cómo han ayudado al desarrollo del proyecto, tanto por mejora en la implementación del código, cómo en encontrar y resolver errores desconocidos.

Revision table

Primera revisión	20/05/2024
Última actualización	

Introduction

A lo largo de este informe veremos las pruebas que se han implementado para las 2 entidades del alumno 5. Se ha cubierto solamente la parte del testeo formal, puesto que por falta disponibilidad de tiempo para esta entrega, se ha probado lo que se ha podido y cómo se ha podido.

Las pruebas implementadas para casos positivos y negativos están guardadas como *acción-a-probar.safe*, mientras que los casos de hackeo/ciberataque están guardados como *acción-a-probar.hack*. Se han implementado un archivo *.hack* y otro *.safe* por las acciones listar, mostrar, crear, borrar, actualizar y publicar de las entidades CodeAudit y AuditRecord.

Tras sus correspondientes arreglos, el 100% de las pruebas se ejecutan con éxito obteniendo los resultados esperados.

Es importante destacar que, aunque el tiempo fue limitado, el proceso de prueba fue exhaustivo y se ha seguido la metodología explicada en clase para garantizar la integridad, la calidad y seguridad de las acciones implementadas.

Estas pruebas servirían en un proyecto real para comprobar que siguen funcionando correctamente si se implementaran nuevas entidades/funcionalidades que afecten a las ya implementadas.

Contents

CODE AUDIT

Show .safe y .hack

Los casos de prueba .safe se han reproducido correctamente demostrando la correcta funcionalidad del sistema. En los casos .hack, gracias a estos se ha corregido una validación que permitía mostrar cualquier codeAudit a cualquier auditor.

List .safe y .hack

Los casos de prueba .safe y .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

Create .safe y .hack

Gracias a los casos de prueba .safe se ha encontrado y corregido un error en las validaciones de la fecha. Los casos de prueba .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

Update .safe y .hack

Gracias a los casos de prueba .safe se ha encontrado y corregido un error en las validaciones de la fecha. Los casos de prueba .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

Delete .safe y .hack

Los casos de prueba .safe y .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

Publish .safe y .hack

Los casos de prueba .safe y .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

AUDIT RECORD

Show .safe y .hack

Los casos de prueba .safe y .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

List .safe y .hack

Los casos de prueba .safe y .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

Create .safe y .hack

Gracias a los casos de prueba .safe se ha encontrado y corregido un error en las validaciones de la fecha. Los casos de prueba .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

Update .safe y .hack

Gracias a los casos de prueba .safe se ha encontrado y corregido un error en las validaciones de la fecha. Los casos de prueba .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

Delete .safe y .hack

Los casos de prueba .safe y .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

Publish .safe y .hack

Los casos de prueba .safe y .hack se han reproducido correctamente demostrando la correcta funcionalidad del sistema.

PERFORMANCE TESTING

Se ha cubierto solamente la parte del testeo formal, puesto que por falta disponibilidad de tiempo para esta entrega y problemas al procesar los datos en excel, se ha probado lo que se ha podido y cómo se ha podido.

Conclusions:

Podemos observar que la creación de estas pruebas no solo ha ayudado a la detección de errores, sino que también ayudará a comprobar en el futuro el correcto funcionamiento de las funcionalidades implementadas para un buen mantenimiento del proyecto, garantizar la calidad del software, seguridad, y reducción de costes a largo plazo; esto último ayudando a encontrar problemas antes de que sea muy costoso resolverlos.

Bibliography

Intentionally blank.