Escuela Técnica Superior de Ingeniería Informática

Diseño y Pruebas II

Report on prior knowledge of WIS testing

C1.03.06 - Entregable 1

Campos Mejías, Gonzalo (GM), goncammej@alum.us.es

Gallardo Pelayo, Alejandro (AG), alegalpel@alum.us.es

López-Benjumea Novella, Alberto Miguel (ALB), a.lopezbenjumea@gmail.com

Medina Durán, Alejandro (AM), alepridur@alum.us.es

Vargas Castro, Francisco José (FJV), fravarcas1@alum.us.es

Repositorio: https://github.com/alejandromd/acme-L3-D01

17 FEBRERO - 2023

Historial de Revisiones

Versión	Fecha	Autor(es)	Descripción
1.1	12.02.2023	AM	Creación del documento
1.2	12.02.2023	AM	Versión incial completa de acuerdo a los requisitos
2.0	15.02.2023	GM	Revisión y formato formal del report
3.0	16.02.2023	GM, AM	Versión final del documento

Resumen ejecutivo

El presente informe tiene como finalidad exponer el conocimiento previo que el grupo posee sobre las pruebas en un WIS (Web Information System). Dicho informe se elabora con el propósito de documentar y rememorar los conceptos adquiridos con anterioridad a esta asignatura, los cuales pueden resultar de utilidad tanto en un contexto laboral como en el marco del proyecto actual.

Tabla de Contenidos

I. Introducción

En este documento se realizará un resumen tanto del conocimiento previo de los integrantes del grupo sobre testear un WIS(Web Integrated Software) como en la experiencia del desarrollo de pruebas para el mismo. En cuanto a experiencia de desarrollo de una aplicación web contamos con el proyecto de la asignatura ya cursada Diseño y Pruebas 1 en el cual hemos trabajado con repositorios, servicios, controladores, vistas, entidades, validadores y pruebas(explicado en el apartado de Contenido), y la asignatura de Introducción a la Ingeniería del Software y los Sistemas de Información 2 donde usamos otro framework, React.

Cuando decimos WIS nos referimos a una arquitectura de software basada en la web donde se integran múltiples aplicaciones y servicios de software, normalmente a través de una interfaz del navegador web. Permite un acceso a componentes de software y fuentes de datos, por ejemplo el patrón MVC, modelo-vista-controlador se usa en Spring, donde a través de la interfaz de usuario se envían peticiones HTTP al controlador que comprueba el formato de los datos enviados y en el caso de estar correctos, la capa de servicios invoca a un repositorio de una entidad el cual devuelve unos datos de la base de datos a la capa de servicios y este a su vez al controlador, actualizando la vista correspondiente. La arquitectura WIS se utiliza en sistemas software complejos que requieren la integración de múltiples componentes de software. Ayudando a la flexibilidad y escalabilidad para construir y gestionar grandes sistemas de software.

II. Conocimiento previo sobre el testing WIS

Durante nuestra formación en la titulación de Ingeniería del software de la Universidad de Sevilla la realización de pruebas solo se ha estudiado en la asignatura de Diseño y Pruebas 1. Dichas pruebas han sido unitarias y hemos usado el bean MockMvc para simular peticiones HTTP al controlador y verificar las respuestas como los datos y redirección.

Las pruebas unitarias se usan para testear de forma aislada al resto del sistema unidades o componentes individuales de una aplicación software. Consiste en escribir casos de prueba automatizados que verifiquen el comportamiento y la funcionalidad de cada unidad, normalmente utilizando un marco de pruebas. Con el objetivo de detectar defectos o errores en el código en una fase temprana del proceso de desarrollo de software, antes de que puedan causar problemas mayores, al ser exponencial el coste de arreglo de fallos conforme avanza el proyecto.

Al aislar las unidades individuales y probarlas de forma independiente, se pueden identificar y corregir rápidamente los errores, mejorar la calidad del código y reducir el riesgo de introducir nuevos problemas a medida que evoluciona la aplicación. Esto mejora la calidad del software y facilita su futuro mantenimiento.

III. Conclusión

Contamos con poca experiencia previa respecto a la implementación de test pero tenemos el conocimiento proporcionado de la asignatura antes mencionada, esto engloba a la arquitectura WIS, como distintos tipos de pruebas que los antes mencionados, arquitectura por capas, etc

IV. Referencias

Intentionally blank