Selenium Web Driver

Proposal

ในปัจจุบันการนำระบบ Software System มาใช้งานจริงในการทำกิจกรรมเพื่อบรรลุเป้าหมายที่ตั้งเป้าหมายไว้ โดยเฉพาะ Web Application ในการแก้ปัญหาที่ต้องการแก้ไขทั้งในภาคการศึกษา ภาคธุรกิจ ภาคราชการ และอื่น ๆ ซึ่งภายในปัจจุบันระบบ Web Application มีความซับซ้อนกว่าเดิมเป็นอย่างมากจากช่วงต้นของการพัฒนา Web Application ช่วงต้นทศวรรษ 2000 ทำให้การทดสอบการทำงานของ Web Application ในปัจจุบันการทดสอบด้วยการใช้มนุษย์ไม่สามารถทำได้อย่างทั่วถึงและเที่ยงตรง ทำให้เกิดความผิดพลาดจากมนุษย์ขึ้นได้ และยังทำให้เวลาในการพัฒนา Software เพิ่มมากขึ้น โดยปรกติแล้วเวลาในการทดสอบ Web Application เป็นเวลามากถึงร้อยละ 30 - 60 ของอายุขัยในการพัฒนา Software ซึ่งถือว่าเป็นขั้นตอนที่สำคัญ และใช้เวลาเป็นอย่างมากรวมถึงทรัพยากรบุคคลที่ต้องถูกนำมาใช้ในการทดสอบการทำงานของ Software ซ้ำ ภายใน Task เดิม ดังนั้นจึงทำให้มีการเกิดของ Selenium WebDriver Automation Testing Framework ขึ้นมาซึ่งเป็น Software ที่เข้ามาแก้ไขปัญหาดังกล่าว ทั้งช่วยแก้ไขปัญหาในเรื่องของการลดทรัพยากรบุคคลที่ต้องใช้เป็นจำนวนมากในการทดสอบ Software ขนาดใหญ่ และการใช้เวลาการทดสอบที่มากในการทดสอบ และจัดทำรายงานการทดสอบที่ละเอียด และมีประสิทธิภาพ สามารถบ่งชี้ถึงความผิดพลาดของ Web Application ที่ทำการทดสอบนั้น ทำให้สามารถแก้ไขจุดบกพร่องของ Web Application ได้อย่างมีประสิทธิภาพ ทำให้มีเวลาในการขัดเกลาประสิทธิภาพของ Software อีกด้วย รวมถึงการที่ Tester ไม่จำเป็นต้องสร้างชุดคำสั่งสำหรับการทดสอบเองทั้งหมด ทำให้เกิดความสะดวก โดยเฉพาะในด้านของการที่ Selenium WebDriver Automation Testing Framework มีการพัฒนโดยตลอดเวลาจึงทำให้สามารถทำงานร่วมกับ Web Application ที่มีการเพิ่ม Features การทำงานใหม่เข้ามาโดยที่ Tester ไม่ต้องแก้ไขชุดคำสั่งทั้งหมดด้วยตนเอง และ Selenium WebDriver Automation Testing Framework ยังสามารถแสดงการวิเคราะห์ และประเมินประสิทธิภาพในด้านต่าง ๆ ของ Web Application ได้อีกด้วย พร้อมไปถึงจำลองสถานการณ์เมื่อ User เข้ามาใช้งานระบบของ Web Application รวมถึงสถานการณ์ที่เกิดขึ้นได้ทั้งหมด ซึ่งอาจส่งผลถึงเสถียรภาพ ความปลอดภัย ความมั่นคงของระบบ Software System ในรูปของ Web Application

Architectural Patterns

โดยที่ Architectural Patterns ของ Selenium WebDriver เป็นสถาปัตยกรรมแบบ Representational State Transfer (REST) ซึ่งเป็นการประยุกต์ใช้เป็น API โดยจะทำการส่งคำสั่งจาก Selenium Client ไปที่ WebDriver โดยใช้การส่ง HTTP Request และรอรับ Response จากการทำงานของ WebDriver

Quality Attributes

Portability โดยที่การพัฒนามุ่งเน้นให้ Selenium Web Driver เป็นโครงการ Software ในระยะยาว และการทดสอบควรจะสามารถทำได้จากทุก Platform, Operating System เพื่อสามารถทดสอบการทำงานบนทุกอุปกรณ์ที่ผู้ใช้งานมีได้ เนื่องจาก Function การทำงานของ Web Application มีระยะที่เป็นวงกว้างอย่างมาก

Usability เนื่องจากมีกลุ่มผู้ที่ใช้งานเป็นกลุ่มใหญ่ทั้ง Tester และ Developer ที่ต้องการใช้ Automated Tests และเครื่องมืออื่น ๆ ดังนั้นการทำให้ Selenium WebDriver สามารถใช้งาน และเข้าใจได้ง่ายจะทำให้ผู้ใช้งานเกิดความสะดวก

Compatibility โดยทาง WebDriver อนุญาตให้สามารถมาตราฐาน Web อื่นสามารถรองรับการทดสอบเพื่อทดสอบ Feature ใหม่ต่าง ๆ ที่ถูกเพิ่มเข้ามา และยังสามารถทำให้ Vendor สามารถเพิ่มการทดสอบเฉพาะของ Browser ของตนเองได้อีกด้วย

Reference :

https://www.w3.org/TR/webdriver1/#design-notes
https://www.sciencedirect.com/science/article/pii/S1877050915005396
https://www.tutorialspoint.com/what-is-the-selenium-web-driver-architecture#:~:text=Selenium%20WebDriver%20API%20enables%20interaction,Python%2C%20C%23%20and%20so%20on

matplotlib

Proposal

Matplotlib เป็นการรวบรวม Library ของภาษา Python ซึ่งจำเป็นในการสร้างข้อมูลที่มีอยู่สามารถอธิบายออกมาได้ด้วยภาพต่าง ๆ ซึ่งทำให้สามารถบรรยายข้อมูลที่มีอยู่ให้สามารถเข้าใจได้ง่ายขึ้นโดยใช้ภาพเป็นเครื่องบรรยาย ทั้งเป็นแบบ Static, Animated,Visualizations ที่มีปฏิสัมพันธ์ได้อีกด้วย สามารถทำได้ทั้งการ Zoom, Pan, Update ซึ่งก็คือ Graphic ที่ใช้ในการ Publication รวมถึงยังสามารถใช้ในการเปลี่ยนหน้าตา รูปแบบของ Visualization ได้ตามที่ต้องการ และการนำข้อมูลไปใช้ต่อที่หลากหลายทั้งนำไป Exports เป็นไฟล์สกุลได้มากมาย และด้วยการที่ matplotlib เป็น Library ของภาษา Python จึงสามารถนำไปฝังกับ User Interface ตามการประยุกต์ใช้ที่ต้องการ รวมถึงประสิทธิภาพในการสร้าง Visualization ที่ใช้ทรัพยากรน้อยกว่าการใช้ User Interface Software อื่น ๆ ที่ใช้ทรัพยากรมากกว่า และยังใช้ร่วมกับ Third-Party Package ที่สามารถทำให้ทำงานร่วมกันได้อย่างมีประสิทธิภาพ อาทิ เช่น NumPy ที่ใช้ร่วมกันในการคำนวณและ Plot Graph ต่าง ๆ โดย matplotlib จะช่วยในการ Visualization ข้อมูลที่มาจากการคำนวณให้สามารถอธิบายข้อมูลได้ประสิทธิภาพจากการคำนวณมากยิ่งขึ้น

Architectural Patterns

โดยที่ Architectural Patterns สถาปัตยกรรมของ matplotlib เป็นสถาปัตยกรรมแบบ Layer โดยที่การทำงานของ Layer ซึ่งชั้นการทำงานของ matplotlib จะแบ่งเป็น 3 Layer ได้แก่ Backend Layer, Artist Layer, Scripting Layer ซึ่ง Backend Layer จะเป็น Layer ที่มีความซับซ้อนมากที่สุด เป็น Layer ที่ทำหน้าที่คำนวณ และเป็น Algorithm ในการคำนวณทั้งหมดของ matplotlib Library ซึ่งทำหน้าที่ติดต่อสื่อสารกับ Toolkit เหมือนกับ wxPython หรือภาษาที่ใช้วาดภาพเช่น PostScript ภายในเครื่องที่ใช้ในการคำนวณ เป็น Layer ที่รวม Logical Function ของ matplotlib Libraryไว้ โดยมี 3 Built-In Abstract Interface หลักไว้คือ

- **FigureCanvas** ทำให้สามารถ Render Canvas ได้
- **Renderer** เป็น Abstract Class ทำหน้าที่ป็น Handler ในการ Operation Render หรือการวาด รับผิดชอบการทำงานภายใน FigureCanvas
- **Event** เป็น Event Handler Input ของผู้ใช้งาน matplotlib Library จาก Keyboardและ Mouse Click เป็นต้น

Artist Layer เป็น Layer ที่อนุญาตให้ผู้ใช้สามารถควบคุมการทำงาน และดึงประสิทธิภาพของ Element ที่อยู่ใน Figure ให้มากที่สุดเท่าที่เป็นไปได้ ช่วยให้ผู้ใช้สามารถปรับเปลี่ยน Element ที่อยู่ใน Renderer บน Canvas ซึ่งมีการสร้าง Artist Instance บน matplotlib Figure ได้แก่ The title, the lines, the tick labels, the images รวมไปถึง instance ที่เฉพาะทางในด้านอื่นที่เฉพาะเจาะจงด้วย และมี Artist Object 2 ประเภท ซึ่งประเภทแรกคือ Primitive Type เช่น Line2D, Rectangle, Circle, และ Text และประเภทที่สองคือ Composite Type เช่น Axis, Tick, Axes และ Figure

Scripting Layer เป็น Layer ถูกออกแบบมาใช้ matplotlib สามารถทำงานได้เหมือน Matlab Script โดยจะเป็นการรวบรวม Command ทำให้สามารถใช้งาน Layer นี้ได้ง่ายเหมือนเป็นการเขียนชุดคำสั่งทั่วไปผ่าน Command ที่ matplotlib Library Built-in ไว้ให้

Quality Attributes

Performance Efficiency โดยที่มุ่งน้นให้ matplotlib จะมีประสิทธิภาพในการทำ Real-Time Plotting โดยใช้จำนวนจริงใด ๆ ในการ Plot องค์ประกอบต่าง ๆ ของ Visualization

Usability โดยที่ matplotlib จะทำการรวบรวม Function สำหรับการทำ Visualization ทั้งหมดไว้ใน Scripting Layer สามารถทำให้ผู้ใช้สามารถเรียนรู้การใช้งานได้ง่าย และยังสามารถใช้เครื่องมือต่าง ๆ ที่อยู่ภายใน Library ได้

Integrability จากการที่ matplotlib สามารถทำงานได้ร่วมกับ Python Library ได้หลากหลาย Library Third-Party Package อาทิเช่น Pandas, NumPy โดยที่สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพซึ่ง Function ของแต่ละ Library จะทำงานร่วมกันจาก Built-In Function ที่ได้สร้างไว้แล้วจาก Developer

Reference :

https://ieeexplore.ieee.org/document/4160265
https://medium.datadriveninvestor.com/data-visualization-with-python-matplotlib-architecture-6b05af533569
https://matplotlib.org/stable/index.html#

Kill Bill

Proposal

Kill Bill เป็น Admin User Interface ซึ่งเป็น Open-Source ที่ถูกสร้างขึ้นมาเพื่อเป็นแนวทางแก้ปัญหาเกี่ยวกับการวางบิล และการชำระเงินบน Online Application ที่อยู่บนระบบ Software System สำหรับองค์กรที่มีองค์ประกอบเกี่ยวกับธุรกิจในด้านการเงินหลายชนิด เช่น บัญชี, ใบแจ้งหนี้, การชำระเงิน และยังรวมไปถึงการบริการลูกค้า การดำเนินการ เกี่ยวกับการจัดการของหน่วยงานในองค์กรในการจัดการเกี่ยวกับการเงิน เจ้าของโครงการที่รับผิดชอบเกี่ยวกับสินค้า ผลิตภัณฑ์ ซึ่งต้องมีการจัดการเกี่ยวกับการวางบิล การชำระเงินเพื่อดำเนินการกิจการ โดยสามารถจัดการกับระบบที่อยู่ภายใน Kill Bil ผ่าน Kaui ได้อีกด้วย เพื่อแก้ปัญหาในเรื่องของการจัดการกับหลังบ้านของธุรกิจที่ต้องใช้การจ่ายเงินผ่าน Online Application และยังรวมไปถึงการทำสมัครสมาชิกรายเดือนที่ต้องใช้ระบบในการเรียกเก็บค่าบริการเป็นรายเดือน ช่วยแก้ปัญหาเรื่องการออกบิลที่ซับซ้อนภายในองค์กร และยังมีการเก็บประวัติการทำรายการต่าง ๆ ย้อนหลังเพื่อตรวจสอบข้อมูล นำไปวิเคราะห์เพื่อต่อยอดในการทำกิจกรรมต่าง ๆ นั้นให้มีประสิทธิภาพมากขึ้น พร้อมทั้งยังสามารถติดตั้ง Plugin ถอนการติดตั้ง Start, Stop, และ Restart ได้ขณะใช้งาน

Architectural Patterns

โดยที่ Architectural Patterns ของ Kill! Bill เป็นสถาปัตยกรรมแบบ Hybrid ระหว่างสถาปัตยกรรมแบบ Layer และ Microkernel โดยที่ในส่วนของ Kernal จะเป็น Layer 2 Layer ประกอบด้วย Core Foundations และ Core Services ซึ่ง Core Foundations จะเป็น Function พื้นฐานทั้งหมดสำหรับ Software Payment โดยที่จะทำหน้าที่เป็น Base Class ให้กับ Core Services โดยที่ Core Services จะทำหน้าที่ตามที่ Services ที่เป็น Built-In Services ถูกเรียกใช้ในขณะนั้นโดยการเรียกใช้ Services ต่าง ๆ เช่น account management, invoicing, entitlement, dunning จะถูกเรียกผ่าน Event ซึ่งเป็น Set ของ Event ที่ถูกเก็บไว้ใน Persistent Event Services Bus ภายใน Core Services Layer และในส่วนของสถาปัตยกรรมแบบ Microkernel จะเป็นการที่ระบบ Core Layer ที่ Core Services Layer จะสามารถเชื่อมต่อกับ Plugin ภายนอกซึ่งถูกพัฒนาจากทีมพัฒนาอื่นได้ ทำให้เกิดการที่ Kil Bill สามารถสร้าง Function การทำงานได้เอง และสามารถนำ Services ส่วนอื่นมาใช้ได้ และ

สามารถเปลี่ยนแปลง Services ที่เชื่อมต่ออยู่ได้ตลอดเวลา โดยที่การติดต่อกับ Application อื่น ๆ เพื่อทำการดำเนินการกับ Services ที่ทำงานผ่านการทำงาน HTTP Request โดยใช้ API

Quality Attributes

Modularity เนื่องจาก Kill Bill สามารถติดตั้ง Modules ที่เป็น Plugin ต่าง ๆ ได้อย่างอิสระในขณะที่ Software ยังทำงานอยู่โดยไม่จำเป็นต้อง Restart โดยผ่าน OSGi Framework ทำให้เกิดความสะดวกในการทำงานเรื่องส่วนเสริมซึ่งสามารถควบคุมการทำงานได้ตลอดเวลา Runtime

Integrability โดยที่ Software สามารถทำงานร่วมกับ Online Application อื่นได้โดยการติดต่อผ่านทาง API โดยใช้ HTTP Request ทำให้การทำงานร่วมกันบน Software System สามารถทำงานได้สะดวก และสามารถทำงานร่วมกับ Application อื่นที่พัฒนาในภายหลังได้ โดยใช้การเชื่อมต่อที่ API เดียวกัน

Recoverability มาจากการที่เนื่องจากการทำงานที่เป็น Hybrid ทำให้เมื่อ Plugin ที่เชื่อมต่อกับ Microkernal จะสามารถสั่งให้ Plugin Service ตัวนั้นหยุดการทำงานเพื่อจำกัดวงความเสียหายไว้ที่ตรงนั้นทำให้ แก้ไขความเสียหายที่เกิดขึ้นได้ โดยเป็นการ Recover From Fault ทำให้การทำงานจะทำความเสียหายไว้ที่ Plugin Service เท่านั้น

Reference :

https://killbill.io/solutions/
https://docs.killbill.io/latest/internal_design.html