

Audacity

Purposes :

Audacity เป็นโปรแกรม OpenSource ที่ใช้ตัดต่อและบันทึกเสียงด้วยการใช้ Effect ต่างๆ ซึ่งมีอยู่มากมายและครบครัน เช่น Noise Reduction (ลดเสียง) Amplify (เพิ่มเสียง) ปรับระดับของเดซิเบล ฯลฯ ถึงแม้ว่า Audacity จะเป็นโปรแกรมที่มีเครื่องมือในการใช้งานเป็นจำนวนมาก แต่ก็ยังเป็นโปรแกรมที่สามารถใช้ง่าย เพราะโปรแกรมนี้ได้แปลเป็นหลายๆภาษาทั่ว Audacity สามารถดาวน์โหลดฟรีได้ในทุกๆแพลตฟอร์ม ไม่ว่าจะเป็น Windows, Macs, Ubuntu (อ้างอิง:

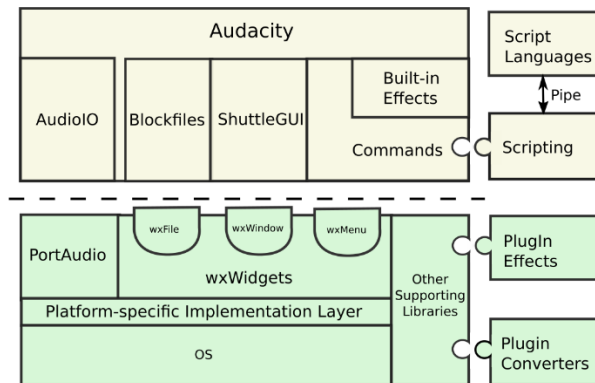
<https://www.mindphp.com/%E0%B8%9A%E0%B8%97%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1/web-app-graphic/8040-audacity-program.html>)

Architectural Patterns/Styles :

Audacity มีพื้นฐานจากการใช้ Library จำนวนมาก โดย Library ที่สำคัญคือ PortAudio ที่ให้ Low-level Audio Interface และ wxWidgets ที่ให้ GUI Components

Library อื่นๆที่ใช้ต่อยอดจาก wxWidgets และ PortAudio

- BlockFile ใช้ OS file system ผ่าน wxWidgets เพื่อให้วิธีการเก็บเสียงเป็นกลุ่มเล็กๆ ซึ่งทำให้สามารถตัดต่อปรับแต่งเสียงโดยไม่จำเป็นต้องปรับแต่งทั้งไฟล์
- ShuttleGUI ใช้ wxWidgets ในการจัดการ dialog, ปุ่ม และการควบคุมอื่นๆ เพื่อการเขียนโค้ดซ้ำ ผ่านการเก็บข้อมูลเป็นตัวแทน
- Command จัดการ Bind ปุ่มในคีย์บอร์ด ผ่าน wxWidgets
- AudioIO จัดการการเคลื่อนย้ายเสียงระหว่าง Sound card, memory, hard disk ผ่าน PortAudio



(อ้างอิง: <https://wiki.audacityteam.org/wiki/ArchitecturalDesign>)

Quality Attributes :

1. Usability

จากจุดมุ่งหมายของตัว open-source ที่ต้องการให้ application ใช้งานง่าย

2. Modifiability

การเป็น open-source และการพยายามเขียนโค้ดในลักษณะเดียวกัน

3. Performance

จากการใช้ BlockFile ให้ ไม่จำเป็นต้องแก้ไขไฟล์ทั้งไฟล์

(อ้างอิง: <https://www.aosabook.org/en/audacity.html>)

Selenium Webdriver

Purpose of the project :

Selenium Web Driver เป็นเครื่องมือที่ช่วยให้เราสามารถสร้างโปรแกรมในการทดสอบเว็บแอปพลิเคชันกับ Web browser ได้หลายตัว ซึ่งถือเป็นคุณสมบัติเด่นที่ดีกว่า Selenium IDE ค่ะ (Selenium IDE จะใช้งานได้เฉพาะ firefox เท่านั้น) โดยจะมี Web Driver เป็นตัวกลางที่มีไลบรารีที่ช่วยในการติดต่อกับ Web browser ได้ดังรูปค่ะ



โดยใช้การเขียนโปรแกรมในภาษาต่างๆ ที่เรารู้จักกัน ไม่ว่าจะเป็น JAVA, .Net (VB/C#), Ruby ติดต่อกับไลบรารีของ WebDriver เพื่อเข้าถึงคอนโทรลที่แสดงผ่าน Web browser ได้ ทำให้เราสามารถสร้างโปรแกรมการทดสอบได้หลากหลายมากขึ้น เช่น

- ดึงข้อมูลที่ใช้สำหรับกรอกข้อมูลบนฟอร์ม จากฐานข้อมูลได้
- สามารถใช้ทดสอบหลายๆ กรณี ได้อย่างต่อเนื่อง

(อ้างอิง: <https://sysadmin.psu.ac.th/2015/08/31/selenium-webdriver-part1/>)

Architectural Patterns/Styles :

เพื่อให้เข้าใจ Selenium Webdriver Architecture เราควรรู้ก่อนว่า Webdriver API คืออะไร Selenium Webdriver API คือตัวช่วยในการสื่อสารระหว่างภาษากับเบราว์เซอร์ โดยเบราว์เซอร์แต่ละตัวมี ตรรกะ (Logic) ในการดำเนินการบนเบราว์เซอร์ต่างกัน



Selenium Webdriver API ประกอบด้วยองค์ประกอบหลัก 4 ส่วน ได้แก่

1. Selenium Client Library/Language Bindings

Selenium รองรับไลบรารีหลายตัว เช่น Java, Ruby, Python เป็นต้น โดย Selenium Developers ได้ พัฒนาการเชื่อมโยงภาษาต่างๆเพื่อให้ Selenium รองรับหลายภาษา

2. JSON WIRE PROTOCOL Over HTTP Client

JSON ย่อมาจาก JavaScript Object Notation ใช้ในการถ่ายโอนข้อมูลระหว่าง Server และ Client บนเว็บ JSON Wire Protocol เป็น REST API ที่ถ่ายโอนข้อมูลระหว่าง HTTP Server WebDriver แต่ละตัว มี HTTP Server เป็นของตัวเอง

3. Browser Drivers

แต่ละเบราว์เซอร์จะมี browser driver แยกต่างหาก driver ของ Browser จะสื่อสารกับเบราว์เซอร์ที่เกี่ยวข้องโดยไม่เปิดเผยตรรกะภายใน (Internal Logic) ของฟังก์ชันการทำงานของเบราว์เซอร์ เมื่อ browser driver ได้รับคำสั่งใดๆ มา คำสั่งนั้นก็จะถูก executed บนเบราว์เซอร์นั้นๆ และตอบกลับไปในรูปแบบของ HTTP response

4. Browsers

Selenium รองรับหลายเบราว์เซอร์ เช่น Firefox, Chrome, IE, Safari เป็นต้น

ที่มา : <https://www.edureka.co/blog/selenium-webdriver-architecture/>

Quality Attributes :

- Supportability

One Selenium script for all browsers คือสิ่งที่ชุมชน Selenium ได้ดำเนินการและดำเนินการทุกวัน ตาม StatCounter เบราว์เซอร์ Chrome, Firefox, Safari, Internet Explorer, Opera และ Edge เป็นเบราว์เซอร์ที่ใช้กันมากที่สุดในโลกและสคริปต์ Selenium เข้ากันได้กับเบราว์เซอร์ที่กล่าวถึงทั้งหมด คุณไม่จำเป็นต้องเขียนสคริปต์ใหม่สำหรับทุกเบราว์เซอร์ เพียงแค่สคริปต์เดียวสำหรับทุกเบราว์เซอร์

- Usability

ใช้งานในการง่าย และมีแหล่ง open-souce มากมายให้ศึกษาที่เข้าใจได้ง่าย

- Performance

ทำการทดสอบอัตโนมัติบน application ได้อย่างรวดเร็ว

Joomla

Purpose of the project :

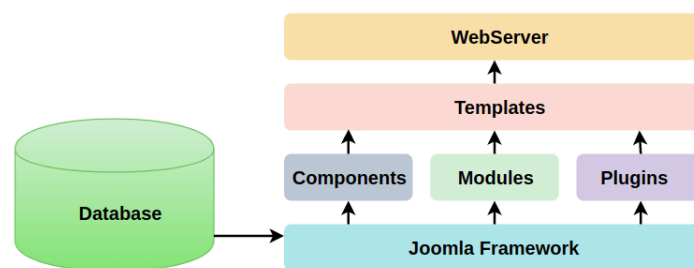
Joomla (จoomla) คือ CMS (ซีเอ็มเอส) ตัวหนึ่งจากหลายๆ ตัวที่ได้รับความนิยมในปัจจุบัน สำหรับคุณที่ยังไม่รู้จักรว่า CMS คืออะไร ขออธิบายสั้นๆ เพิ่มเติมดังนี้ครับ CMS นั้นเป็นอักษรย่อของ คำว่า "Content Management System" (คอนเท้น เมเนจเม้น ซิสเต็ม) ซึ่งเมื่อแปลเป็นภาษาไทย หมายถึง ระบบบริหารจัดการเนื้อหาเว็บไซต์ นั้นหมายความว่า สิ่งที่เราจะต้องดูแลก็คือเนื้อหาของเว็บไซต์ เช่น การเพิ่มบทความ การเพิ่มรูปภาพ หรือการปรับแต่งโยกย้ายโมดูลต่าง ๆ ไม่จำเป็นจะต้องมานั่งเขียน Code ด้วยHTML (เอ็ชทีเอ็มแอล), JavaScript (จาวาสคริปต์), PHP (พีเอ็ชพี), SQL (เอสคิวแอล) เพียงแต่เรียนรู้วิธีการติดตั้ง การใช้งาน และการปรับแต่ง CMS เท่านั้นเอง สำหรับ Code (โค้ด) ต่างๆ ที่นำมาสร้าง และ ออกแบบเว็บไซต์ จะทำโดยทีมงานของผู้พัฒนาจoomlaของแต่ละทีม ซึ่งทำให้ประหยัดเวลาในการสร้างเว็บไซต์ และ ออกแบบเว็บไซต์ ได้อย่างมาก

(อ้างอิง:<https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2620-joomla-cms.html>)

Architectural Patterns/Styles :

Joomla เขียนด้วย PHP และอิงตามรูปแบบการออกแบบ MVC (Model-View-Controller) ใช้ MySQL (MS SQL เวอร์ชัน 2.5 ขึ้นไป และ PostgreSQL เวอร์ชัน 3.0 ขึ้นไป) เพื่อจัดเก็บข้อมูล มีคุณสมบัติหลากหลาย (เช่น การแคชหน้า บล็อก โพล การสนับสนุนภาษาสากล และฟีด RSS เป็นต้น) ซึ่งทำให้ Joomla เป็นตัวเลือกที่ยอดเยี่ยมสำหรับ CMS (ระบบการจัดการเนื้อหา)

แผนภาพด้านล่างแสดงสถาปัตยกรรมโครงสร้างของ Joomla:



Joomla Architecture

ประกอบด้วย :

-Database ประกอบด้วยข้อมูล ยกเว้นไฟล์รูปภาพและเอกสารที่สามารถจัดเก็บ จัดการ และจัดระเบียบในลักษณะเฉพาะ ประกอบด้วยข้อมูลผู้ใช้ เนื้อหา และข้อมูลที่จำเป็นอื่นๆ ของไซต์ นอกจากนี้ยังมีข้อมูลการดูแลระบบเพื่อให้ผู้ดูแลระบบสามารถเข้าถึงไซต์และจัดการได้อย่างปลอดภัย Joomla database layer เป็นหนึ่งในปัจจัยที่สำคัญที่สุดที่รับประกันความยืดหยุ่นและความเข้ากันได้สูงสุดสำหรับ

-Joomla Framework ประกอบด้วยคอลเล็กชัน library/open-source software library ซึ่งสร้างระบบจัดการเนื้อหาของ Joomla ไม่จำเป็นต้องติดตั้ง Joomla Framework เพื่อใช้ CMS หรือในทางกลับกัน Joomla Framework มีกลุ่มของไฟล์ที่เป็นประโยชน์ในการสร้างทั้ง web และ command-line applications มันแบ่งกรอบงานออกเป็น single modular package และยิ่งไปกว่านั้น ยังช่วยให้แต่ละแพ็คเกจพัฒนาได้ง่ายขึ้น

-Component เรียกว่า mini-applications ซึ่งประกอบด้วยสองส่วน:

- 1.ผู้ดูแลระบบ (Administrator)

- 2.เว็บไซต์ (Site)

เมื่อใดก็ตามที่โหลด page ขึ้น component จะถูกเรียกเพื่อแสดงเนื้อหาของหน้าหลัก ส่วน Administrator จัดการส่วนต่างๆ ของส่วนประกอบ และส่วนของ Site ช่วยในการแสดง page เมื่อมีผู้เยี่ยมชม site ส่งคำขอ component เรียกว่าหน่วยการทำงานที่สำคัญของ Joomla

- Modules ใช้เพื่อแสดงหน้าเว็บใน Joomla ใช้เพื่อแสดงข้อมูลใหม่จาก component พวกเขาสามารถจัดการส่วนของตัวเองได้ และได้รับการจัดการโดย 'ตัวจัดการโมดูล' ซึ่งเป็นส่วนประกอบเอง พวกมันดูเหมือนกล่องต่างๆ เช่น Login module นอกจากนี้ยังช่วยในการแสดงเนื้อหาและรูปภาพใหม่เมื่อโมดูลเชื่อมโยงกับส่วนประกอบ Joomla

- Plugin สามารถอธิบายได้ว่าเป็นส่วนขยาย Joomla ที่ยืดหยุ่นและทรงพลังมาก ซึ่งใช้ในการขยายเฟรมเวิร์ก plugin คือโค้ดบางตัวที่รันใน event triggers โดยทั่วไปจะใช้เพื่อจัดรูปแบบผลลัพธ์ของ component หรือ module เมื่อมีการพัฒนา page ฟังก์ชัน plugin ที่เกี่ยวข้องกับเหตุการณ์มักจะดำเนินการตามลำดับเมื่อใดก็ตามที่มีเหตุการณ์เกิดขึ้น

- Templates ใช้สำหรับจัดการรูปลักษณ์ของเว็บไซต์ Joomla โดยทั่วไปมีเทมเพลตสองประเภทให้เลือก front-end และ back-end เทมเพลต Front-end

- Web Server เป็นเซิร์ฟเวอร์ที่ใช้เชื่อมต่อผู้ใช้กับไซต์ ให้บริการหน้าเว็บแก่ลูกค้า HTTP (HyperText Transfer Protocol) ใช้ในการสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์

(อ้างอิง:https://www.tutorialspoint.com/joomla/joomla_architecture.htm#:~:text=Joomla%20Framework%20%E2%88%92%20Framework%20is%20a,are%20considered%20as%20mini%20applications.)

Quality Attributes :

-Security

ความปลอดภัยเป็นปัจจัยสำคัญที่ควรพิจารณาเมื่อคุณสร้างเว็บไซต์ Joomla ให้การรับรองความถูกต้องสองปัจจัยแก่คุณเพื่อหลีกเลี่ยงโอกาสในการแฮ็ค ดังนั้นเว็บไซต์ของคุณจะไม่ได้รับ hacked แต่ในกรณีที่ คุณจะออกชื่อผู้ใช้และรหัสผ่านและคนที่พบบ่อยมากได้รับในคุณสามารถ Joomla เว็บไซต์

- Useability

Joomla เป็นโอเพ่นซอร์สและใช้งานได้ฟรีทั้งหมด แต่ฟรีไม่ได้หมายความว่ามันขาดคุณสมบัติ อันที่จริง อินเทอร์เฟซที่เป็นมิตรต่อผู้ใช้อย่างสมบูรณ์จะทำให้คุณประหลาดใจด้วยคุณสมบัติ WYSIWYG ที่ให้ผลลัพธ์เหมือนกันทุกประการ ปัจจัยที่ทำให้รู้สึกดีอีกประการหนึ่งก็คือการอัปเดตบ่อยๆ Joomla นำเสนอการอัปเดตใหม่ในรูปแบบของคุณสมบัติและฟังก์ชันใหม่ ด้วยการอัปเดตใหม่ทุกครั้ง การทำงานจะง่ายขึ้น

- Modifiability

การเพิ่มเติมหรือเปลี่ยนแปลงเนื้อหาไม่ว่าจะเป็นรูปภาพหรือตัวอักษร (Content) ก็สามารถทำได้โดยง่าย มีลักษณะการใช้งานเป็นแบบ WYSIWYG (what you see in what you get) เช่นเดียวกับ MS Word ซึ่งผู้ใช้ไม่จำเป็นต้องมีความรู้ทางด้าน HTML ทำให้การเพิ่มเติมหรือเปลี่ยนแปลงเป็นไปอย่างรวดเร็ว ไม่ยุ่งยาก

Lab Architectural Patterns/Styles

Instructions

-Please choose 2 projects from the following open-source software projects in the list below and write a (very) brief report for each project. Your report must explain what is the purpose of the project, its architectural patterns/styles, as well as 3 quality attribute scenarios. The quality attributes must be aligned with the quality attribute advocated/promoted by project developers/maintainers. Please include in your report references to the sources where the relevant information can be found.

Audacity [<https://www.audacityteam.org/>]

gpsd [<https://gpsd.gitlab.io/gpsd/>]

matplotlib [<https://matplotlib.org/>]

Selenium WebDriver [<https://www.selenium.dev/>]

Zotonic [<https://zotonic.com/>]

-Please choose 1 project from the following open-source software projects in the list below and write a (very) brief report for a project. Your report must explain what is the purpose of the project, its architectural patterns/styles, as well as 3 quality attribute scenarios. The quality attributes must be aligned with the quality attribute advocated/promoted by project developers/maintainers. Please include in your report references to the sources where the relevant information can be found.

Jitsi [<https://jitsi.org/>]

Joomla [<https://www.joomla.org/>]

Kill Bill [<https://killbill.io/>]

ONOS [<https://opennetworking.org/onos/>]

Yesod [<https://www.yesodweb.com/>]

Hint: Apart from the provided websites, you should also search for more information from other sites.

IMPORTANT: Upload or Commit your report on GitHub and submit a link to your file(s). Your file(s) can be in any format: docx, pdf, txt, etc. Please named the file with your ID followed by the word "Architecture". For example, "48010522 Architecture.xxx"