

การออกแบบยูสเซอร์อินเทอร์เฟซ (User Interface Design)

ยูสเซอร์อินเทอร์เฟซ ถือเป็นส่วนหนึ่งของระบบที่จัดเตรียมไว้เพื่อให้ผู้ใช้ได้โต้ตอบกับระบบผ่านทางหน้าจอ และใช้เป็นเครื่องมือนำทางเข้าสู่ระบบได้ในที่สุด พึงเข้าใจว่า ระบบสารสนเทศที่พัฒนาขึ้นมา นั้น มีจุดประสงค์คือ การประมวลผลข้อมูลเพื่อให้ได้มาซึ่งสารสนเทศตามที่ต้องการ ซึ่งเป็นกระบวนการทำงานในเชิง “คอมพิวเตอร์ศาสตร์” ในขณะเดียวกัน ยูสเซอร์อินเทอร์เฟซที่มนุษย์ใช้ติดต่อกับระบบ จัดเป็นแนวคิดเชิง “จิตวิทยา” มนุษย์มีความคิด มีความรู้สึกต่อการโต้ตอบกับสิ่งเร้าต่างๆ ดังนั้น การออกแบบยูสเซอร์อินเทอร์เฟซ จึงจำเป็นต้องนำทั้งศาสตร์ และศิลป์มาพิจารณาร่วมกัน เพื่อนำไปสู่การออกแบบที่ถูกต้องตามหลักการ เพื่อให้มนุษย์สามารถปฏิสัมพันธ์กับ เครื่องจักร (คอมพิวเตอร์) และสามารถสั่งงานเพื่อควบคุมเครื่องจักรได้อย่างสะดวก รวดเร็ว และถูกต้อง เพื่อนำไปสู่การประมวลผลข้อมูลที่มีประสิทธิภาพ และผู้ใช้รู้สึกพอใจและยอมรับในที่สุด

Helpful Tips

เคยสังเกตหรือไม่ว่า ระบบปฏิบัติการที่ใช้งานบนโทรศัพท์เคลื่อนที่หรือสมาร์ทโฟนแต่ละยี่ห้อแต่ละรุ่นนั้น จะมีการใช้งานยากง่ายแตกต่างกันไป บางรุ่นใช้งานง่าย ทำให้ผู้ใช้สามารถโต้ตอบกับระบบและแสดงผลในสิ่งที่ต้องการได้อย่างรวดเร็ว ในขณะที่บางรุ่น กว่าที่จะเข้าถึงในสิ่งที่ต้องการ กลับยุ่งยากกว่า ใช้ภาษาที่เข้าใจยาก สิ่งเหล่านี้ล้วนเกี่ยวข้องกับหลักการออกแบบยูสเซอร์อินเทอร์เฟซทั้งสิ้น ดังนั้น ระบบปฏิบัติการบนสมาร์ทโฟนที่มีระบบอินเทอร์เฟซที่ดี ผู้ใช้ส่วนใหญ่ล้วนพึงพอใจ เพราะช่วยให้พวกเขาสามารถเข้าถึงในสิ่งที่ต้องการได้ง่ายและสะดวกขึ้น ช่วยลดภาระให้กับพวกเขา ก่อให้เกิดความรู้สึกที่ดี ทำงานองเดียวกันกับระบบสารสนเทศ ที่จำเป็นต้องออกแบบยูสเซอร์อินเทอร์เฟซที่ดี เพื่อให้ผู้ใช้สามารถเข้าถึงเพื่อควบคุมและโต้ตอบกับระบบได้อย่างสะดวก ใช้งานง่าย เพื่อนำไปสู่การประมวลผลที่ถูกต้อง ตรงตามความต้องการ

* จุดประสงค์ของการออกแบบยูสเซอร์อินเตอร์เฟซ

1. เพื่อบอกระบบให้รู้ว่าได้ทำกิจกรรมอะไรลงไป ตัวอย่างเช่น ผู้ใช้ได้เลือกกิจกรรมการประมวลผล ไม่จะเป็นการเคาะปุ่ม Enter, การคลิกปุ่มเพื่อแก้ไขข้อมูล หรือดึงข้อมูลออกมาใช้งาน
2. เพื่ออำนวยความสะดวกต่อการใช้งาน โดยตัวระบบจะเตรียมงานบริการแก่ผู้ใช้เพื่อให้การดำเนินงานด้านการประมวลผลใดๆ สามารถลุล่วงไปอย่างมีประสิทธิภาพและประสิทธิผล
3. หลีกเลี่ยงข้อผิดพลาดที่เกิดจากผู้ใช้ เป็นการป้องกันความผิดพลาด หรือการขัดจังหวะใดๆ ที่อาจเกิดจากความตั้งใจของผู้ใช้หรือไม่ก็ตาม เพื่อให้ระบบไม่สามารถประมวลผลได้ตามความคาดหวัง ซึ่งสิ่งเหล่านั้นอาจส่งผลกระทบต่อความเสียหายต่อข้อมูลในระบบได้

หลักการเบื้องต้นของการออกแบบโดยคำนึงถึงผู้ใช้

หลักการออกแบบเพื่อคำนึงถึงผู้ใช้ จะมีหลักการพื้นฐาน 8 ประการด้วยกัน คือ

1. ทำความเข้าใจเกี่ยวกับฟังก์ชันการทำงานทางธุรกิจต่างๆ ให้ดี
2. นำอินเตอร์เฟซแบบ GUI มาใช้
3. รู้ระดับความสามารถและประสบการณ์ของผู้ใช้ระบบ
4. ต้องคิดว่าตนเอง เสมือนเป็นผู้ใช้คนหนึ่ง
5. นำต้นแบบมาใช้ให้เกิดประโยชน์
6. ออกแบบอินเตอร์เฟซด้วยความเข้าใจ
7. นำความคิดเห็นที่ได้มาปรับใช้ เพื่อปรับปรุงให้ดียิ่งขึ้น
8. จัดทำเอกสารการออกแบบอินเตอร์เฟซ

* คำแนะนำเกี่ยวกับการออกแบบยูสเซอร์อินเตอร์เฟซที่ดี

จากเนื้อหาที่ผ่านมา ได้กล่าวถึงรายละเอียดในเรื่องของการออกแบบเอาต์พุตและอินพุตมาบ้างแล้ว ดังนั้นในที่นี้เราจะรวมเรื่องของการออกแบบเอาต์พุตและอินพุตเข้าด้วยกันเพื่อนำมาเป็นส่วนติดต่อกับผู้ใช้ เนื่องจากทั้งเอาต์พุตและอินพุต ถือเป็นส่วนหนึ่งของการอินเตอร์เฟซนั่นเอง

ต่อไปนี้เป็นคำแนะนำเกี่ยวกับการออกแบบยูสเซอร์อินเตอร์เฟซที่ดี

1. ผู้ใช้จะต้องรู้เสมอว่า สิ่งที่กำลังโต้ตอบกับระบบอยู่นั้น คืออะไร และจะต้องดำเนินการต่อไปอย่างไร ดังนั้นระบบที่ดีควรมีการจัดเตรียมคำแนะนำให้แก่ผู้รับทราบว่าจะต้องดำเนินการต่อไปอย่างไร เช่น ใ้บันทึกข้อมูล หรือให้กำหนดทางเลือก ซึ่งถือเป็นการส่งผลป้อนกลับมายังผู้ใช้ให้รับทราบ ดังรายละเอียดต่อไปนี้

- บอกให้ผู้รับทราบว่าต้องทำอะไร ด้วยการใช้ข้อความง่ายๆ เพื่อให้ผู้รับทราบว่าตนต้องทำอะไรต่อไป ตัวอย่างเช่น

Please type data

Select one or more options

- บอกให้ผู้รับทราบว่า ข้อมูลที่ป้อนเข้าไบนั้นถูกต้อง เช่น กรณีป้อนข้อมูลฟิลด์หนึ่งเสร็จแล้ว ตำแหน่งเคอร์เซอร์ก็จะกระโดดไปยังฟิลด์ถัดไป ซึ่งถือเป็นการตอบรับโดยอัตโนมัติว่าฟิลด์ที่บันทึกไปก่อนหน้านี้ นั้นถูกต้อง ไม่มีข้อผิดพลาด หรืออาจมีข้อความสั้นๆ เพื่อยืนยันถึงความถูกต้องก็ได้ ดังตัวอย่างเช่น

Data OK

- บอกให้ผู้รับทราบว่า ข้อมูลที่ป้อนเข้าไบนั้นไม่ถูกต้อง อันเนื่องจากการป้อนข้อมูลไม่ตรงตามรูปแบบที่กำหนดไว้ ดังนั้น ควรแจ้งรายละเอียดหรืออธิบายให้รู้ว่า ข้อมูลที่ป้อนไบนั้น ที่ถูกต้องควรป้อนในรูปแบบใด ตัวอย่างเช่น

Incorrect date!

Please type date format mm/dd/yyyy

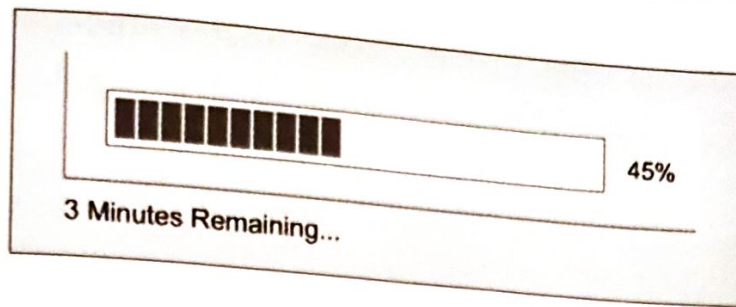
Digit must between 1 - 100

- บอกให้ผู้รับทราบว่า ระบบต้องใช้เวลาในการประมวลผลสักครู่หนึ่ง โดยเฉพาะในบางกิจกรรม จำเป็นต้องใช้เวลาในการประมวลผล ดังนั้น หากระบบไม่มีข้อความใดๆ ปรากฏบนจอภาพให้ผู้รับทราบเลย จอภาพแลดูนิ่งๆ ไม่มีอะไรเกิดขึ้น ทำให้ผู้ใช้เกิดความเข้าใจผิดว่าระบบเกิดข้อขัดข้องหรือเครื่องแสกก็ไม่ได้ ดังนั้น ถ้ากิจกรรมใดๆ ต้องใช้เวลาในการประมวลผลนาน เช่น การจัดเรียงลำดับข้อมูล การจัดเรียงดัชนี หรือการคำนวณตัวเลขที่มีความซับซ้อน ก็ควรมีข้อความแจ้งให้ผู้ใช้ทราบด้วย ตัวอย่างเช่น

Please wait....

This may take a few minutes...

อย่างไรก็ตาม กรณีเป็นอินเตอร์เฟซแบบ GUI เราสามารถนำ Progress Bar มาใช้ ซึ่งจัดเป็นวิธีหนึ่งที่นิยมนำมาใช้ เพราะทำให้ผู้ใช้งานได้เห็นความคืบหน้าของงาน ซึ่งอาจจะระบุเป็นเวลาทั้งหมดที่ต้องใช้ในการประมวลผล (Total Time) พร้อมเวลาที่ผ่านไป (Elapsed Time) หรืออาจแสดงเพียง Remaining Time ว่าคงเหลือเวลาอีกเท่าไร จึงจะประมวลผลเสร็จ



- บอกให้ผู้ใช้ทราบว่า งานที่ส่งไปประมวลผลนั้นเสร็จสมบูรณ์หรือไม่ ซึ่งข้อความเหล่านี้จัดเป็นสิ่งสำคัญทีเดียว เนื่องจากเป็นคำยืนยันถึงการดำเนินงานว่าเสร็จสมบูรณ์หรือไม่ ตัวอย่างเช่น

Printing completed

File has been deleted

Try again...or contact your network administrator

2. ควรจัดรูปแบบข้อมูลที่น่าเสนออยู่บนหน้าจอ ให้สามารถแสดงสารสนเทศได้หลายรูปแบบ เช่น นอกจากผู้ใช้จะรับรู้ถึงข้อมูลที่แสดงบนจอภาพแล้ว ในกรณีที่ต้องการดูรายละเอียดข้อมูลหรือสารสนเทศเฉพาะส่วน ระบบควรมี Pop-Up Dialogue Windows เพื่อให้ผู้ใช้งานสามารถชมเพื่อดูรายละเอียดเฉพาะส่วนที่ต้องการได้

3. ข้อความ คำอธิบาย หรือสารสนเทศที่แสดง จะต้องมีความยาวเพียงพอ ที่ผู้อ่านได้อ่านแล้วเกิดความเข้าใจในทันที และยอมรับ

4. การนำเสนอข้อความบนจอภาพ ควรใช้เทคนิคและเจดสีที่เหมาะสม เช่น การกำหนดข้อความเป็นแถบแสง (Reverse Video) การกำหนดสีเพื่อเน้นข้อความ (Highlighting) หรือการกำหนดให้ข้อความกระพริบ (Blinking) แต่ไม่ควรกำหนดมากเกินไปจนเกิดความจำเจ เพราะจะเป็นการรบกวนสายตาและก่อความรำคาญใจให้กับผู้ใช้ ดังนั้น ควรเลือกใช้กับข้อความสำคัญๆ ที่ต้องการเน้นจริงๆ เท่านั้น และจะต้องไม่มากเกินไปจนดูเลอะเทอะ แลดูไม่น่าเชื่อถือ

5. ในกรณีที่ระบบต้องกำหนดค่าเบื้องต้นให้กับตัวแปรหรือสภาพแวดล้อมต่างๆ เพื่อใช้งานเป็นจำนวนมาก “การกำหนดค่าปกติ (Default Value)” ให้กับตัวแปรเหล่านั้น จัดเป็นทางเลือกที่เหมาะสม เนื่องจากในบางระบบมีความจำเป็นต้องตั้งค่าต่างๆ ให้กับสภาพแวดล้อมในระบบมากมาย และถ้าค่าเหล่านั้น ผู้ใช้ต้องกำหนดเองทั้งหมด ก็คงเป็นเรื่องที่ยุ่งยากไม่น้อย ดังนั้น การกำหนดค่าเริ่มต้นให้กับระบบด้วยค่าดีฟอลต์ จึงเป็นทางออกที่ดีที่สุด ในขณะเดียวกัน หากผู้ใช้ที่มีความเชี่ยวชาญ ก็สามารถปรับแต่งค่าต่างๆ เพิ่มเติมได้ด้วยตนเอง

6. ในกรณีที่ผู้ใช้มีการสั่งรันงานใดๆ ที่อาจส่งผลกระทบต่อความเสียหายในข้อมูลหรือความปลอดภัยต่อระบบ เช่น การสั่งลบไฟล์ข้อมูล ก็ควรมีข้อความหรือไดอะล็อกซ์เพื่อยืนยันในสิ่งที่จะทำก่อน ตัวอย่างเช่น

Are you sure to delete this file? (Y/N)

7. ควรเอาใจใส่ต่อข้อผิดพลาดที่เกิดขึ้น ด้วยการไม่อนุญาตให้มีการประมวลผลใดๆ จนกว่าจะดำเนินการแก้ไขข้อผิดพลาดในข้อมูลให้ถูกต้องเสียก่อน ตัวอย่างเช่น หากยังไม่ได้ลงบัญชีบางรายการ ระบบจะไม่อนุญาตให้ปิดงบดุล เป็นต้น
8. ในกรณีที่ผู้ใช้ มีเจตนากระทำในบางสิ่ง ที่ส่งผลความเสียหายต่อระบบ วิธีล็อกแป้นพิมพ์โดยไม่ให้สามารถสั่งการหรือป้อนข้อมูลใดๆ ได้อีกต่อไป จัดเป็นวิธีการป้องกันที่ดี จากนั้นก็แสดงข้อความให้รับทราบ ดังตัวอย่างเช่น

The keyboard should be locked to prevent any further input and an instruction to call the technical support should be displayed.