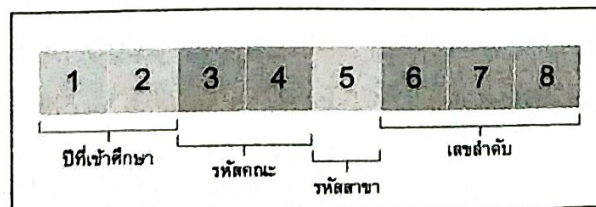


วิธีการลงรหัส (Coding Methods)

การลงรหัสให้กับข้อมูล จัดเป็นแนวทางหนึ่งที่ถูกพัฒนาขึ้นมาเพื่อลดจำนวนอินพุต ควบคุมข้อผิดพลาด และต้องการความรวดเร็วในการประมวลผล การลงรหัสสามารถใช้ตัวเลขหรือตัวอักษรเพียงไม่กี่ตัว เพื่ออ้างอิงรายละเอียดข้อมูลตามที่เราต้องการ โดยในที่นี้จะขอกล่าวถึงวิธีการลงรหัสด้วยวิธีดังต่อไปนี้

- รหัสเพื่อจำแนกหมวดหมู่ (Classification Codes)** เป็นวิธีการลงรหัสเพื่อจำแนกสิ่งบางสิ่ง ด้วยการแบ่งแยกกลุ่มอย่างชัดเจนเพื่อให้เป็นไปตามประเภทนั้นๆ ตัวอย่างเช่น ธนาคารแห่งประเทศไทย ได้กำหนด “รหัสสถาบันทางการเงิน” เพื่อจำแนกตามกลุ่มต่างๆ ดังต่อไปนี้
 - 1 = ธนาคารพาณิชย์จดทะเบียนในประเทศ
 - 2 = สาขานาคารในต่างประเทศ
 - 8 = บริษัทบริหารสินทรัพย์สถาบันการเงิน
 - 9 = บริษัทผู้ประกอบธุรกิจบัตรเครดิตที่มีใช้สถาบันการเงิน
- รหัสตามหน้าที่ (Function Code)** เป็นวิธีการกำหนดสถานะการทำงาน โดยนักวิเคราะห์ระบบ มักนำรหัสประเภทนี้มาใช้งานอยู่บ่อยๆ เพื่อสั่งให้ระบบประมวลผลตามสิ่งที่ต้องการ ตัวอย่างเช่น มีการออกแบบการประมวลผลเพิ่มข้อมูล และได้กำหนดโค้ดลงรหัสเพื่อทำหน้าที่ต่างๆ ดังต่อไปนี้คือ
 - A หรือ 1 เป็นการเพิ่มเรคอร์ดเข้าไปในแฟ้ม (Addition)
 - D หรือ 2 เป็นการลบเรคอร์ดออกจากแฟ้มข้อมูล (Deletions)
 - C หรือ 3 เป็นการแก้ไขเรคอร์ดข้อมูล (Change)
 - U หรือ 4 เป็นการอัปเดตข้อมูล (Update)
- รหัสจัดลำดับ (Sequence)** เป็นรหัสตัวเลขหรือตัวอักษรอย่างใดอย่างหนึ่ง ถูกกำหนดขึ้นมาเพื่อบอกลำดับของเหตุการณ์ที่เกิดขึ้น ตัวอย่างเช่น ลูกค้ายี่เข้ามาดำเนินธุรกรรมกับทางธนาคาร ระบบจะรันเลขที่อ้างอิงแล้วพิมพ์ลงในเอกสารฉบับนั้น ซึ่งหมายเลขอ้างอิงของเอกสารแต่ละฉบับจะถูกรันเรียงตามลำดับตัวเลข หรือบิลใบเสร็จรับเงินที่ได้พิมพ์ลำดับเลขที่เอาไว้ เป็นต้น
 - รหัสแบบซัพเซต (Subset Code)** บ่อยครั้งที่เดียวที่การอ้างอิงข้อมูลจากรหัส ต้องการรายละเอียดมากกว่าหนึ่งสิ่งด้วยกัน ตัวอย่างเช่น รหัสประจำตัวนักศึกษาที่ประกอบด้วยตัวเลข 8 หลักนั้น เลข 2 ตัวแรก แทนปีที่เข้าศึกษา, เลขสองตัวถัดไป แทนคณะ, เลขตัวถัดไป แทนสาขา และเลขอีก 3 ตัวถัดไป ใช้แทนลำดับเลขที่



- รหัสช่วยจำ (Mnemonic Codes) เป็นวิธีการลงรหัสด้วยการใช้ชื่อย่อของสิ่งๆ หนึ่ง เพื่อช่วยให้นึกถึงคำเต็มของสิ่งๆ นั้นได้ทันที ซึ่งจัดเป็นวิธีลงรหัสที่ได้รับความนิยมเช่นกัน ตัวอย่างเช่น

CS = Computer Science

IT = Information Technology

วิธีการป้อนข้อมูล (Data Entry Methods)

ในการป้อนข้อมูลเข้าสู่ระบบให้มีประสิทธิภาพ นักวิเคราะห์ระบบจะต้องศึกษากระบวนการทางธุรกิจของหน่วยธุรกิจนั้นๆ ก่อน ครั้นได้ทราบถึงทราบแผนกชั้นและการดำเนินงานทางธุรกิจแล้ว ก็จะกำหนดวิธีการนำข้อมูลเข้าสู่ระบบ ซึ่งโดยปกติ เรื่องแรกที่ต้องตัดสินใจให้ได้ก่อนคือ จะใช้วิธีการนำเข้าข้อมูลแบบแบตช์หรือแบบออนไลน์

1. การป้อนข้อมูลแบบแบตช์ (Batch Input) วิธีนี้จะรวบรวมข้อมูลไว้เป็นชุดหรือเป็นกอง ตามกำหนดรอบระยะเวลาหนึ่งๆ เช่น ทุกวัน (หลังเลิกงาน) ทุกสัปดาห์ หรือทุกเดือน นั้นหมายความว่า ผู้ป้อนข้อมูลสามารถกำหนดเวลาได้ว่าจะดำเนินการเมื่อไร ตัวอย่างเช่น เจ้าหน้าที่แผนกบุคลากรได้รวบรวมบัตรลงเวลาทำงานของพนักงานทุกคน มาป้อนเข้าสู่ระบบเป็นประจำทุกสัปดาห์ หรืออาจารย์ผู้สอนได้รวบรวมคะแนนสอบของนักศึกษาในแต่ละภาคการศึกษา เพื่อรอป้อนเข้าสู่ระบบประเมินผล

2. การป้อนข้อมูลแบบออนไลน์ (Online Input) แม้ว่าการป้อนข้อมูลแบบแบตช์ อาจใช้งานได้ดีกับงานเฉพาะกิจนั้นๆ แต่บางกิจกรรมในระบบธุรกิจไม่สามารถรอได้ จึงจำเป็นต้องป้อนข้อมูลแบบออนไลน์เพื่อให้ระบบประมวลผลและแสดงผลลัพธ์โดยทันที ที่สำคัญ หลายๆ องค์กรได้หันมาใช้วิธี การส่งข้อมูลเข้าสู่ระบบโดยตรง (Source Data Automation) ผ่านเครือข่ายการสื่อสารที่มีประสิทธิภาพสูง เช่น

- ธุรกิจขายหน้าร้าน (Point-Of-Sale : POS) นำเครื่องอ่านบาร์โค้ดอ่านป้ายรหัสสินค้าเพื่อป้อนรหัสสินค้าเข้าสู่ระบบโดยตรง และใช้เครื่องอ่านแถบแม่เหล็ก เพื่อป้อนข้อมูลบัตรเครดิตเข้าสู่ระบบแบบอัตโนมัติ ซึ่งกระบวนการดังกล่าว ช่วยให้การดำเนินงานรวดเร็วยิ่งขึ้น
- เครื่องบริการเงินด่วน (Automatic Teller Machines : ATM) จะมีช่องสอดบัตร เพื่ออ่านแถบแม่เหล็กหรือชิปบนบัตรเอทีเอ็ม
- คนงานในโรงงานใช้บัตรประจำตัวแบบบาร์โค้ด สแกนผ่านเครื่องอ่านบัตรเพื่อบันทึกเวลาเข้างานและเวลาเลิกงาน เพื่อให้ทางบริษัทสามารถติดตามต้นทุนการผลิตได้อย่างถูกต้อง
- คลังสินค้าได้ใช้ระบบ RFID ในการตรวจสอบความเคลื่อนไหวของสินค้าภายในคลัง โดย RFID จะใช้คลื่นวิทยุแผ่รัศมีเป็นวงกว้างเพื่ออ่าน/บันทึกข้อมูลจากแผ่นป้ายที่ติดกับตัวสินค้า โดยไม่จำเป็นต้องสัมผัสกับตัวสินค้าโดยตรง

การตรวจสอบความถูกต้องจากการป้อนข้อมูล (Input Validation)

จุดมุ่งหมายของการออกแบบอินพุตก็คือ เพื่อลดข้อผิดพลาดที่เกิดขึ้นในระหว่างการป้อนข้อมูล แต่อย่างไรก็ตาม นักวิเคราะห์ระบบจะต้องคิดอยู่เสมอว่า ข้อผิดพลาดเหล่านี้สามารถเกิดขึ้นได้ จึงต้องพยายามค้นหาให้พบในระหว่างการป้อนข้อมูลและแก้ไขก่อนที่จะถูกจัดเก็บหรือประมวลผลข้อมูล มิฉะนั้นจะกลายเป็นเรื่องที่ยุ่งยากมาก โดยเฉพาะการแก้ไขข้อมูลที่ผิดพลาดภายหลังจากถูกบันทึกลงไปในระบบแล้ว

การตรวจสอบความถูกต้องของข้อมูลนำเข้า จะทำการตรวจสอบข้อมูลที่ป้อนเข้ามาและจะปฏิเสธรายการที่ป้อนผิดพลาดหรือไม่ตรงตามเงื่อนไข ซึ่งมีอยู่หลายวิธีด้วยกันคือ

1. **การตรวจสอบลำดับ (Sequence Check)** เป็นวิธีการตรวจสอบความถูกต้องของข้อมูล โดยทดสอบว่ากลุ่มของข้อมูลที่ป้อนเข้าสู่ระบบนั้น มีการเรียงลำดับเลขที่ตามเอกสารหรือไม่ ตัวอย่างเช่น หมายเลขสั่งซื้อที่ได้ป้อนเข้าสู่ระบบ หากปรากฏว่าลำดับเลขที่ที่ป้อนเข้าไป ไม่ได้เรียงต่อจากลำดับก่อนหน้า นั้นหมายความว่าเกิดการลำดับข้อมูลผิดพลาดแล้ว ระบบก็จะแจ้งข้อผิดพลาดเตือนออกมาให้ทราบ

2. **การตรวจสอบว่ามีการป้อนข้อมูลหรือไม่ (Existence Checks)** เป็นวิธีตรวจสอบโดยจะไม่ยอมให้ผู้ใช้สั่งบันทึกข้อมูลได้ จนกว่าจะป้อนข้อมูลบางฟิลด์ให้ครบถ้วน เช่น หากผู้ใช้ไม่ได้ป้อนหมายเลขประจำตัวบัตรประชาชนของลูกค้า ระบบจะไม่ยอมให้บันทึกข้อมูล จนกว่าจะกรอกลงไปให้ถูกต้อง

3. **การตรวจสอบชนิดข้อมูล (Data Type Check)** เป็นวิธีตรวจสอบว่าข้อมูลที่ป้อนเข้าไปตรงกับชนิดข้อมูลที่ต้องการหรือไม่ เช่น ข้อมูลที่ป้อนต้องเป็นค่าตัวเลข หรือเป็นตัวอักษร A ถึง Z เท่านั้น หากป้อนตัวอักษรอื่นๆ นอกเหนือจากที่กำหนด ระบบจะปฏิเสธโดยทันที

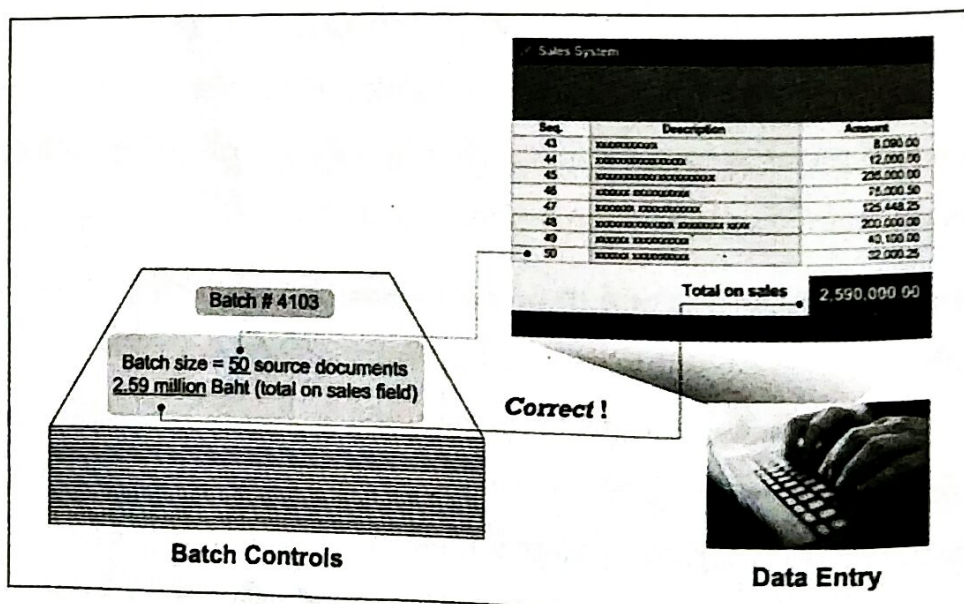
4. **การตรวจสอบช่วงข้อมูล (Range Check)** เป็นการตรวจสอบช่วงของค่าตัวเลข ซึ่งจะต้องเป็นไปตามค่าที่อยู่ในช่วงระหว่างค่าต่ำสุดและค่าสูงสุด เช่น จำนวนชั่วโมงการทำงานของพนักงานในแต่ละวัน ค่าที่เป็นไปได้จะต้องอยู่ระหว่าง 0 จนถึง 24 เท่านั้น ดังนั้น หากมีการป้อนค่าต่ำกว่า 0 หรือมากกว่า 24 ระบบก็จะไม่ยอมรับค่าตัวเลขดังกล่าว ต้องกลับไปป้อนค่าใหม่จนกว่าจะถูกต้อง อย่างไรก็ตาม หากการตรวจสอบช่วงข้อมูลเป็นไปในรูปแบบของการกำหนดค่าสูงสุดของข้อมูลนำเข้า เช่น ระดับอุณหภูมิ วงเงินอนุมัติ จำนวนหน่วยสั่งซื้อ เราจะเรียกการตรวจสอบเหล่านี้ว่า การตรวจสอบวงจำกัด (Limit Check) / 2

5. **การตรวจสอบความสมเหตุสมผล (Reasonableness Check)** ค่าต่างๆ ที่ผู้ใช้ป้อนเข้าสู่ระบบ ในบางครั้งอาจมีเรื่องให้น่าชวนสงสัยอยู่ไม่น้อย แต่ไม่จำเป็นต้องหมายความว่า ข้อมูลที่ป้อนเข้าป้อนนั้นจะผิดเสมอไป ตัวอย่างเช่น สมมติว่า จำนวนชั่วโมงการทำงานของพนักงานในแต่ละวัน มีค่าที่เป็นไปได้คือ 0 ถึง 24 แต่ถ้าบังเอิญมีการป้อนค่า 24 เข้าไป ก็ยังถือว่าผ่านเกณฑ์การตรวจสอบ แล้วจะมีความเป็นไปได้หรือไม่ ที่พนักงานจะทำงานตลอด 24 ชั่วโมงในหนึ่งวัน โดยไม่มีการพัก สิ่งเหล่านี้ก็คือความสมเหตุสมผลนั่นเอง

6. การตรวจสอบความถูกต้อง (Validity Check) เป็นการตรวจสอบข้อมูลที่ป้อนเข้าไป ว่าเป็นไปตามที่กำหนดไว้หรือไม่ เช่น รหัสเพศของลูกค้า จะต้องกรอกค่า F (Female) หรือ M (Male) เท่านั้น นอกจากนี้ยังรวมถึงอีกกรณีหนึ่ง เช่น การตรวจสอบความถูกต้องของรหัสลูกค้าที่สั่งซื้อสินค้า โดยรหัสลูกค้าดังกล่าวจะต้องได้รับการตรวจสอบในแฟ้มลูกค้าว่ามีรหัสนี้อยู่จริงหรือไม่ เพราะระบบจะต้องนำข้อมูลจากรหัสนี้ไปใช้อย่างถูกต้องกับแฟ้มข้อมูลอื่นๆ อีก เช่น แฟ้มรายการสั่งซื้อ หรือแฟ้มจัดส่งสินค้า ซึ่งวิธีนี้สามารถเรียกได้อีกวิธีหนึ่งว่า การตรวจสอบความคงสภาพของการอ้างอิง (Referential Integrity)

7. การตรวจสอบความสัมพันธ์ของข้อมูล (Combination Check) เป็นวิธีการตรวจสอบที่จะดำเนินการกับฟิลด์ข้อมูลอย่างน้อย 2 ฟิลด์ด้วยกัน เพื่อให้เกิดความมั่นใจว่า ข้อมูลทั้งสองจะมีความสอดคล้องกันหรือไม่ ตัวอย่างเช่น นักศึกษาที่สังกัดคณะวิทยาศาสตร์ ก็จะต้องเลือกสาขาในสังกัดคณะวิทยาศาสตร์เท่านั้น จะไปเลือกสาขาที่สังกัดในคณะอื่นๆ คงเป็นไปได้

8. การคุมชุดเอกสาร (Batch Controls) เป็นวิธีการตรวจสอบความถูกต้องของข้อมูล ที่เหมาะกับการประมวลผลข้อมูลแบบแบตช์ วิธีการก็คือ จะเตรียมใบปะหน้าหรือเอกสารใบหนึ่ง ที่นำมาใช้คุมชุดเอกสารในแต่ละกอง เพื่อความถูกต้องและความครบถ้วนในการบันทึกข้อมูล ตัวอย่างเช่น มีกองเอกสารเกี่ยวกับการขายสินค้าอยู่กองหนึ่ง โดยสมมุติว่ากองเอกสารดังกล่าวมีการกำกับหมายเลขแบตช์เอาไว้คือ Batch #4103 มีเอกสารจำนวน 50 ฉบับ และมียอดขายรวมทั้งสิ้น 2.59 ล้านบาท จากนั้นก็ให้เขียนข้อมูลเหล่านี้ลงไปในใบปะหน้าว่า "Batch #4103 มีจำนวนเอกสารอยู่ 50 ฉบับ และมียอดขายรวมทั้งสิ้น 2.59 ล้านบาท" ซึ่งใบปะหน้าใบนี้ก็คือ Batch Control นั่นเอง ต่อมาได้มอบหมายให้พนักงานนำเอกสารกองนี้ไปคีย์เข้าระบบ ครั้นเมื่อพนักงานป้อนข้อมูลเสร็จเรียบร้อยแล้ว รายการข้อมูลที่ป้อนเข้าไปจะต้องมีครบ 50 รายการ และต้องมียอดขายรวม 2.59 ล้านบาท ถ้าข้อมูลในระบบกับ Batch Control ตรงกัน จึงถือว่าข้อมูลที่กรอกเข้าไปถูกต้อง ครบถ้วน แต่ถ้ายอดไม่ตรงกัน ก็หมายความว่าเกิดข้อผิดพลาดขึ้นแล้ว



รูปที่ 8.6 การตรวจสอบความถูกต้องของข้อมูลที่ป้อนเข้าไป ด้วยวิธีคุมชุดเอกสาร (Batch Controls)

9. การประมวลผลซ้ำ (Duplicate Processing) มีความเป็นไปได้ว่า บางระบบงานที่มีความอ่อนไหวมาก ๆ จำเป็นต้องประมวลผลข้อมูลซ้ำมากกว่าหนึ่งครั้ง ด้วยอุปกรณ์ที่แตกต่างกัน หรือใช้รูปแบบที่แตกต่างกัน แล้วนำผลลัพธ์ที่ได้มาทำการเปรียบเทียบเพื่อบรรลุข้อตกลงและความถูกต้อง โครงการอวกาศของประเทศสหรัฐอเมริกา มีการนำคอมพิวเตอร์หลายเครื่อง หลายระบบ มาประมวลผลข้อมูลที่เหมือนกัน แล้วนำมาเปรียบเทียบผลลัพธ์ ซึ่งการประมวลผลซ้ำเป็นการสร้างความแน่ใจในผลลัพธ์ว่า จะต้องมีความถูกต้องสูงสุด

การแก้ไขข้อมูลจากการทำธุรกรรม

ความจริงแล้วมีอยู่ 3 วิธีด้วยกัน ที่เราสามารถนำมาใช้เพื่อแก้ไขข้อมูลจากการทำธุรกรรมที่ผิดพลาด ให้กลายเป็นข้อมูลที่ถูกต้อง วิธีแรกก็คือ การแก้ไขด้วยตนเอง ส่วนอีก 2 วิธี ที่เราจะกล่าวถึงก็คือ การแก้ไขแบบอัตโนมัติ และวิธีการแก้ไขด้วยเลขตรวจสอบ

1. การแก้ไขแบบอัตโนมัติ (Automatic Correction) ในบางครั้ง นักวิเคราะห์ระบบต้องการให้เขียนโปรแกรมเพื่อแก้ไขข้อผิดพลาดแบบอัตโนมัติ หากข้อมูลที่ป้อนเข้าไปนั้นไม่ถูกต้อง ตัวอย่างเช่น ค่าตัวเลขที่ป้อนเข้าไป จะต้องเป็นค่าติดลบเท่านั้น ดังนั้น หากผู้ใช้กรอกค่าบวกเข้าไป โปรแกรมก็จะปรับเป็นค่าติดลบให้โดยอัตโนมัติ หรืออีกกรณีหนึ่ง เช่น หมายเลขอ้างอิง ที่ถูกกำหนดให้เป็นตัวเลข 9 หลัก แต่ถ้าผู้ใช้กรอกไม่ครบ 9 หลัก โปรแกรมก็จะเติมเลขศูนย์ลงไปยังหน้าจนครบ 9 หลักให้โดยอัตโนมัติ ตัวอย่างเช่น ได้กรอกหมายเลข 1525 ลงไป ก็จะถูกแก้ไขเป็น 000001525 เป็นต้น

2. การแก้ไขด้วยเลขตรวจสอบ (Check Digits) บ่อยครั้งที่เดียวที่ผู้ป้อนข้อมูลได้ป้อนรหัสผิดพลาดไป แทนที่จะดึงข้อมูลที่ถูกต้องขึ้นมา กลับกลายเป็นไม่พบข้อมูล ดังนั้น วิธีแก้ไขข้อผิดพลาดสำหรับในกรณีนี้ก็คือ การนำเลขตรวจสอบเข้ามาช่วย วิธีนี้จะเพิ่มบิตพิเศษจำนวน 1 บิตเข้าไปยังหลักสุดท้ายของชุดเลขรหัสเดิม ซึ่งบิตพิเศษนี้จะได้มาจากการคำนวณ โดยมีหลักการคำนวณดังนี้

Customer number :	2 4 5 8
Weights :	5 4 3 2
Multiply numbers X weights :	10 16 15 16
Sum results :	$10 + 16 + 15 + 16 = 57$
Divide by modulus number :	$57/11 = 5$ with remainder of 2
Subtract remainder from modulus number :	$11 - 2 = 9$
Add check digit to original number :	→ 24589

- นำรหัสลูกค้า (2458) มาถ่วงน้ำหนักในแต่ละหลัก ซึ่งในที่นี้จะถ่วงน้ำหนักโดยเริ่มจากหลักหน่วยให้หาค่าเริ่มต้นเป็น 2 และเพิ่มขึ้นทีละหนึ่งในหลักถัดไป (จากขวาไปซ้าย)
- นำเลขรหัสลูกค้าแต่ละหลัก คูณกับค่าถ่วงน้ำหนัก
- นำผลคูณที่ได้ในแต่ละหลักมารวมกัน ซึ่งมีค่าเท่ากับ 57
- นำค่า 57 มาโมดูลัสด้วย 11 ผลลัพธ์ที่ได้คือ 2 (หารเอาเศษ)
- นำเลขโมดูลัส มาหักลบจากเศษที่ได้ (11 - 2) ผลที่ได้คือ 9
- นำบิตพิเศษหมายเลข 9 เพิ่มต่อท้ายรหัสลูกค้า ซึ่งจะกลายเป็นเลขรหัส 24589

คราวนี้ลองมาดูกันว่า บิตพิเศษที่ใช้เทคนิค Check Digit จะช่วยตรวจสอบว่ากรอกรหัสลูกค้าหรือผิดได้อย่างไร โดยสมมติว่า ผู้ใช้ได้กรอกรหัสลูกค้าหมายเลข 24539 ลงไป ตัวโปรแกรมก็จะนำเลขรหัสมาตรวจสอบในลักษณะเดียวกัน ดังนี้

Customer number :

2 4 5 3

Weights :

5 4 3 2

Multiply numbers X weights :

10 16 15 6

Sum results :

10 + 16 + 15 + 6 = 47

Divide by modulus number :

47/11 = 4 with remainder of 3

Subtract remainder from modulus number :

11 - 3 = 8

Add check digit to original number : —————→ 24538

ผลปรากฏว่า เลขตรวจสอบที่คำนวณได้คือ 8 นั้นหมายความว่า รหัสลูกค้าที่ถูกต้องควรจะเป็น 24538 มิใช่ 24539 ดังนั้น ค่าที่ป้อนเข้าไปเกิดข้อผิดพลาดแล้ว ซึ่งตัวโปรแกรมอาจปรับปรุงเลขรหัสที่ถูกต้องให้ แล้วแสดงข้อความเตือนแจ้งให้ผู้ใช้ทราบ ว่า “รหัสลูกค้า 24539 ผิด ที่ถูกต้องคือรหัส 24538 ใช่หรือไม่?”

การออกแบบหน้าจอและการควบคุมการป้อนข้อมูล

เราได้เรียนรู้เทคนิคการออกแบบอินพุตมาแล้ว โดยเฉพาะการออกแบบฟอร์มเอกสารต้นฉบับ ครั้นเมื่อเข้าสู่ขั้นตอนการออกแบบหน้าจอเพื่ออินพุตข้อมูล จึงจำเป็นต้องออกแบบหน้าจอให้สอดคล้องกับแบบฟอร์มเอกสารต้นฉบับด้วย

ความเป็นจริงแล้ว การออกแบบหน้านั้น คล้ายคลึงกับการออกแบบรายงาน แต่จะแตกต่างกันก็ตรงที่หน้าจอภาพจะมีพื้นที่จำกัด ตัวอย่างเช่น หน้าจอภาพแบบเท็กซ์โหมด จะมี 25 แถว แต่ละแถวมี 80 คอลัมน์ โดยรูปที่ 8.7 เป็นแบบฟอร์มที่นำมาใช้ออกแบบหน้าจอ ในขณะที่รูปที่ 8.8 เป็นมาตรฐานการแบ่งสัดส่วนพื้นที่บนหน้าจอแบบเท็กซ์โหมด