



การเขียนโค้ดและการทดสอบที่ได้รับการช่วยเหลือจาก AI: การประเมินประสิทธิภาพและคุณภาพของโค้ด

AI-Assisted Coding and Testing: A Performance and Code Quality Evaluation

จัดทำโดย

นาย กัมแพงเพชร สิงห์ชรณ รหัสนักศึกษา 653380120-2

นาย ณภัทร ประสงค์ดี รหัสนักศึกษา 653380128-6

นาย ธีรัช เจริญวารี รหัสนักศึกษา 653380268-0

อาจารย์ประจำวิชา

ผศ. ดร.ชิตสุรา สุ่มเล็ก

เอกสารนี้เป็นส่วนหนึ่งของวิชา การประกันคุณภาพซอฟต์แวร์

(Software Quality Assurance)

ประจำภาคเรียนที่ 1 ปีการศึกษา 2567

วิทยาลัยการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น

ชื่อหัวข้อโครงการภาษาไทย

: การเขียนโค้ดและการทดสอบที่ได้รับการช่วยเหลือจาก AI: การประเมินประสิทธิภาพและคุณภาพของโค้ด

ชื่อหัวข้อโครงการภาษาอังกฤษ

: AI-Assisted Coding and Testing: A Performance and Code Quality Evaluation

ชื่อผู้จัดทำโครงการ

: นาย กัมแพงเพชร สิงห์ธรรม รหัสนักศึกษา 653380120-2

นาย ณภัทร ประสงค์ดี รหัสนักศึกษา 653380128-6

นาย ธีรัช เจริญวารี รหัสนักศึกษา 653380268-0

อาจารย์ประจำวิชา

: ผศ. ดร.ชิตสุชา สุ่มเล็ก

สาขาวิชา

: วิทยาการคอมพิวเตอร์ คณะวิทยาลัยการคอมพิวเตอร์

ปีการศึกษา

: 2567

บทคัดย่อ

ในยุคปัจจุบัน AI กำลังเข้ามามีบทบาทสำคัญในการพัฒนาซอฟต์แวร์ โดยเฉพาะอย่างยิ่งในด้านการเขียนโค้ดและการทดสอบ ซึ่งเป็นที่คาดหวังว่าจะช่วยเพิ่มทั้งประสิทธิภาพและคุณภาพของซอฟต์แวร์ได้อย่างมาก

ผลงานวิจัยชนิดนี้จึงมุ่งเน้นไปที่การศึกษาผลกระทบของ AI ต่อกระบวนการพัฒนาซอฟต์แวร์ โดยเปรียบเทียบโค้ดที่เขียนด้วย ChatGPT4o, Gemini 1.5 Pro, Gemini 1.5 Flash, GitHub Copilot เพื่อวิเคราะห์ประสิทธิภาพและคุณภาพแต่ละ Generative AI

การวิจัยครอบคลุมการตรวจสอบคุณภาพของโค้ดในหลายมิติ เช่น การหาข้อผิดพลาด การปฏิบัติตามมาตรฐาน การเขียนโค้ด และการประเมินประสิทธิภาพการทำงานของโปรแกรม ผลการวิจัยจะช่วยให้เราเข้าใจถึงข้อดีและข้อจำกัดของการใช้ Generative AI ในการพัฒนาและทดสอบซอฟต์แวร์ ทั้งในเรื่องความเร็ว ความถูกต้อง และคุณภาพโดยรวม

ข้อมูลเชิงลึกจากการวิจัยนี้จะเป็นประโยชน์อย่างยิ่งต่อผู้พัฒนาและผู้จัดการซอฟต์แวร์ ในการตัดสินใจเลือกใช้เทคโนโลยี Generative AI ในกระบวนการพัฒนาซอฟต์แวร์ได้อย่างเหมาะสมและมีประสิทธิภาพมากขึ้น

การทดลอง และ ผลลัพธ์

1. Chat-GPT 4o

Prompt ที่ใช้ในการ Generate Code Python และ Generate Test ทั้งสามรอบ โดยใช้ Composite Design Pattern ใน การ Gen Code

Use the composite design pattern to Write code in Python programming language from the requirements below.

"ความต้องการระบบ ATM

1. การยืนยันตัวตนผู้ใช้

การป้อนข้อมูล: ผู้ใช้ต้องสามารถใส่บัตร ATM และรหัส PIN ได้

การตรวจสอบ: ระบบต้องตรวจสอบความถูกต้องของบัตรและรหัส PIN

การจำกัดความพยายาม: หากใส่ผิดเกินจำนวนครั้งที่กำหนด บัตรจะถูกยกยื่ด

2. การแสดงยอดเงินคงเหลือ

การเข้าถึงข้อมูล: ผู้ใช้สามารถเลือกดูยอดเงินคงเหลือในบัญชีได้

การแสดงผล: ระบบต้องแสดงยอดเงินคงเหลือปัจจุบัน

3. การถอนเงิน

การเลือกจำนวนเงิน: ผู้ใช้สามารถเลือกจำนวนเงินที่ต้องการถอนได้

การตรวจสอบยอดเงิน: ระบบต้องตรวจสอบว่ามียอดเงินเพียงพอสำหรับการถอน

การจ่ายเงิน: หากมียอดเงินเพียงพอ ระบบจะจ่ายเงินให้ผู้ใช้ และปรับปรุงยอดเงินคงเหลือ

การแจ้งเตือน: หากมียอดเงินไม่เพียงพอ ระบบจะแจ้งเตือนผู้ใช้

4. การฝากเงิน

การรับเงินสด: ผู้ใช้สามารถใส่เงินสดเข้าเครื่อง ATM ได้

การนับเงิน: ระบบต้องนับจำนวนเงินที่ฝากเข้ามา

การปรับปรุงยอดเงิน: ระบบต้องปรับปรุงยอดเงินคงเหลือในบัญชี

5. การเปลี่ยนรหัส PIN

การเข้าถึงฟังก์ชัน: ผู้ใช้สามารถเปลี่ยนรหัส PIN ของตนเองได้

การยืนยัน: ระบบต้องขอรหัส PIN เก่าและรหัส PIN ใหม่ 2 ครั้งเพื่อยืนยัน

การเปลี่ยนรหัส: หากรหัส PIN ใหม่ตรงกัน ระบบจะเปลี่ยนรหัส PIN ให้

6. การพิมพ์สลิป

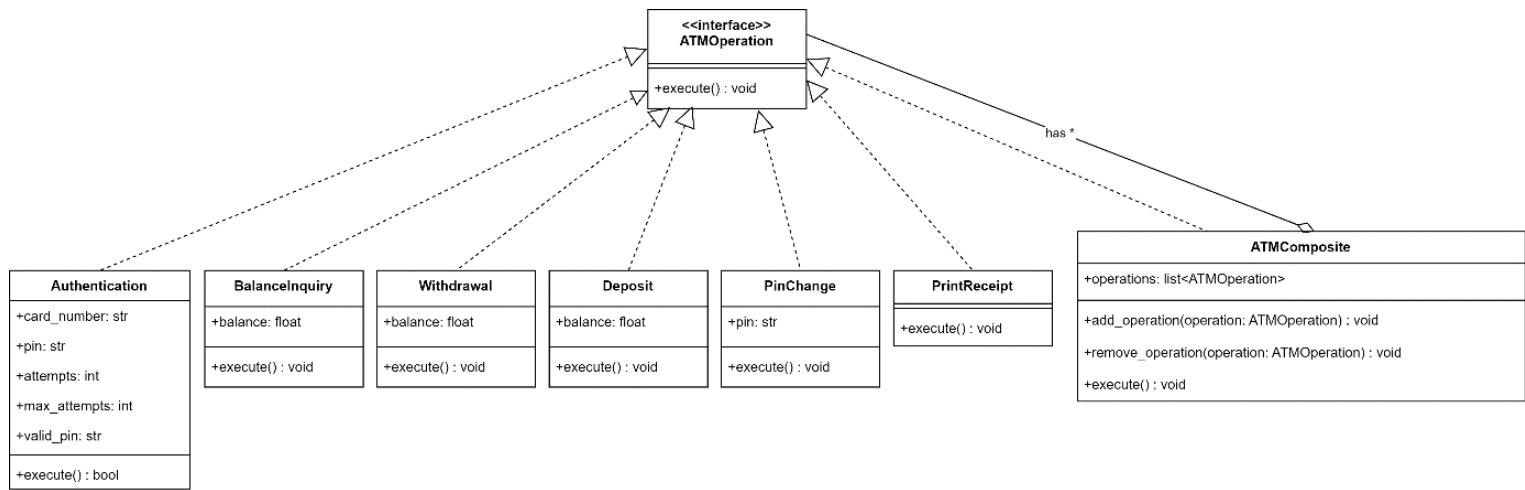
การเลือกพิมพ์: หลังจากทำการเสร็จสิ้น ผู้ใช้สามารถเลือกพิมพ์สลิปได้

เนื้อหาสลิป: สลิปต้องแสดงข้อมูลรายละเอียดของรายการ เช่น วันที่ เวลา จำนวนเงิน และยอดเงินคงเหลือ"

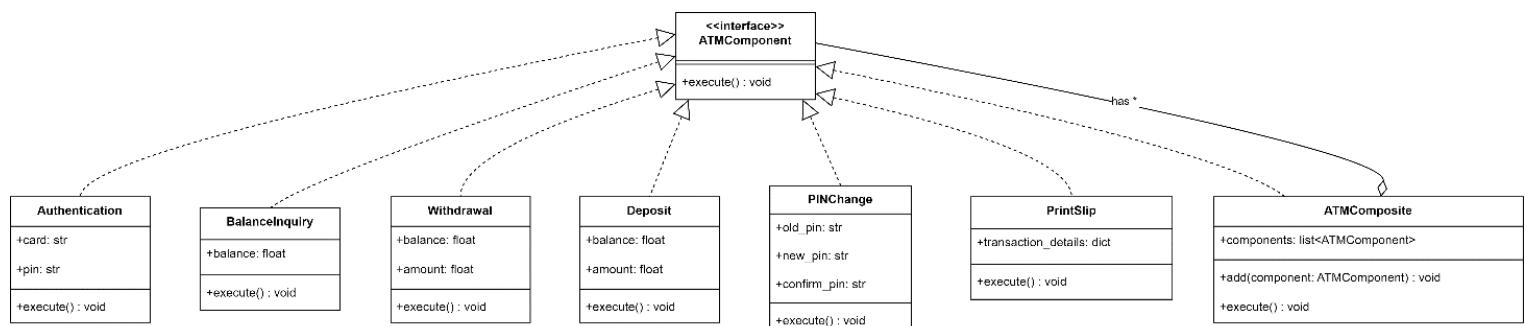
ใช้ Gen Test หลังจาก Gen code เสร็จ

Write a pytest to test your given code that has 100% statement coverage and 100% branch coverage.

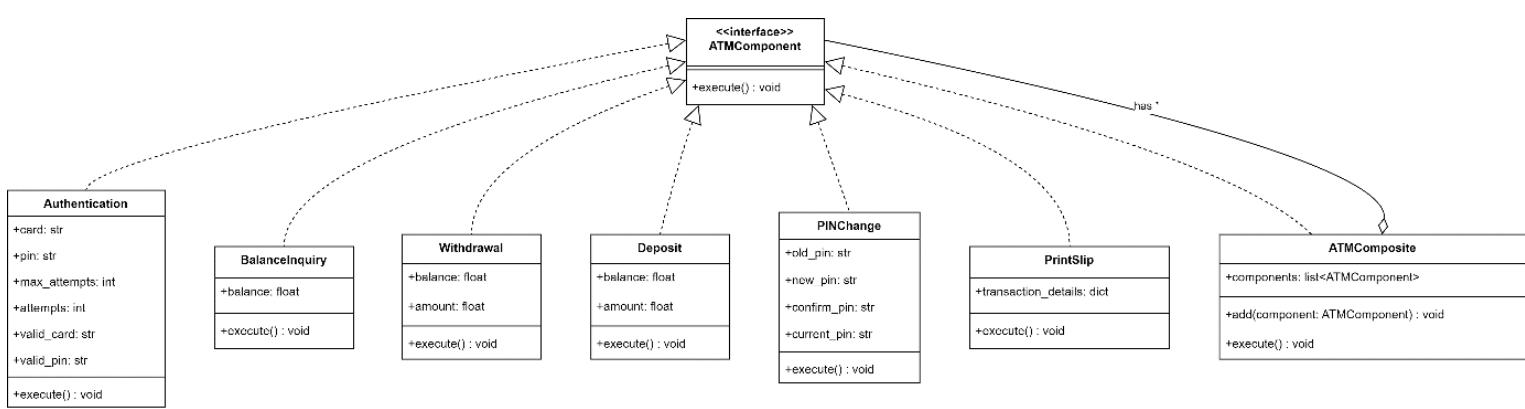
ผลลัพธ์ ในรอบที่ 1



ผลลัพธ์ ในรอบที่ 2



ผลลัพธ์ ในรอบที่ 3



Prompt ที่ใช้ในการ Generate Code Python และ Generate Test ทั้งสามรอบ โดยใช้ Visitor Design Pattern ใน การ Gen Code

Use the visitor design pattern to Write code in Python programming language from the requirements below.

"ความต้องการระบบตู้ ATM

1. การยืนยันตัวตนผู้ใช้

การป้อนข้อมูล: ผู้ใช้ต้องสามารถใส่บัตร ATM และรหัส PIN ได้

การตรวจสอบ: ระบบต้องตรวจสอบความถูกต้องของบัตรและรหัส PIN

การจำกัดความพยายาม: หากใส่ผิดเกินจำนวนครั้งที่กำหนด บัตรจะถูกยกยื่น

2. การแสดงยอดเงินคงเหลือ

การเข้าถึงข้อมูล: ผู้ใช้สามารถเลือกดูยอดเงินคงเหลือในบัญชีได้

การแสดงผล: ระบบต้องแสดงยอดเงินคงเหลือปัจจุบัน

3. การถอนเงิน

การเลือกจำนวนเงิน: ผู้ใช้สามารถเลือกจำนวนเงินที่ต้องการถอนได้

การตรวจสอบยอดเงิน: ระบบต้องตรวจสอบว่ามียอดเงินเพียงพอสำหรับการถอน

การจ่ายเงิน: หากมียอดเงินเพียงพอ ระบบจะจ่ายเงินให้ผู้ใช้ และปรับปรุงยอดเงินคงเหลือ

การแจ้งเตือน: หากมียอดเงินไม่เพียงพอ ระบบจะแจ้งเตือนผู้ใช้

4. การฝากเงิน

การรับเงินสด: ผู้ใช้สามารถใส่เงินสดเข้าเครื่อง ATM ได้

การนับเงิน: ระบบต้องนับจำนวนเงินที่ฝากเข้ามา

การปรับปรุงยอดเงิน: ระบบต้องปรับปรุงยอดเงินคงเหลือในบัญชี

5. การเปลี่ยนรหัส PIN

การเข้าถึงพังก์ชัน: ผู้ใช้สามารถเปลี่ยนรหัส PIN ของตนเองได้

การยืนยัน: ระบบต้องขอรหัส PIN เก่าและรหัส PIN ใหม่ 2 ครั้งเพื่อยืนยัน

การเปลี่ยนรหัส: หากรหัส PIN ใหม่ตรงกัน ระบบจะเปลี่ยนรหัส PIN ให้

6. การพิมพ์สลิป

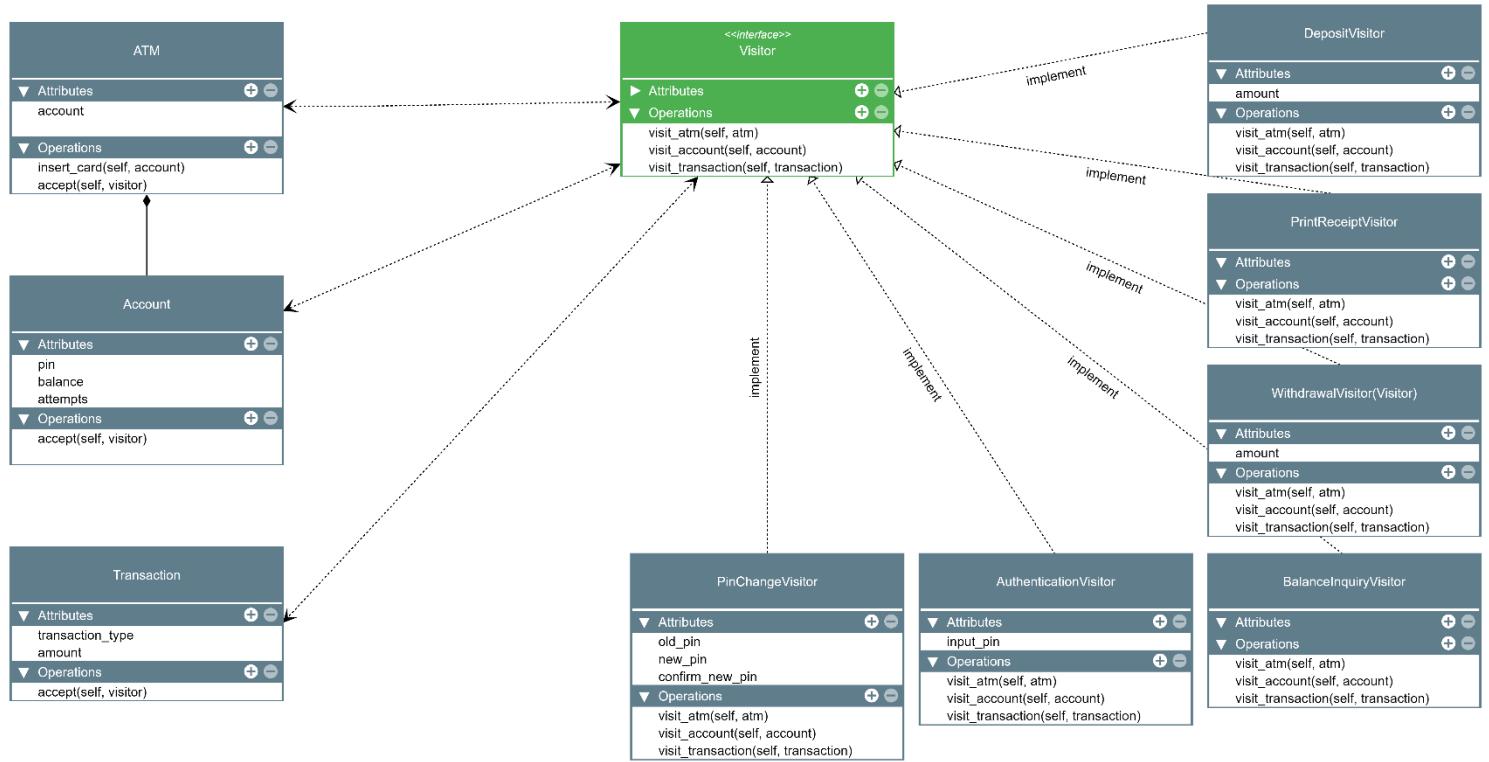
การเลือกพิมพ์: หลังจากทำการเสร็จสิ้น ผู้ใช้สามารถเลือกพิมพ์สลิปได้

เนื้อหาสลิป: สลิปต้องแสดงข้อมูลรายละเอียดของรายการ เช่น วันที่ เวลา จำนวนเงิน และยอดเงินคงเหลือ"

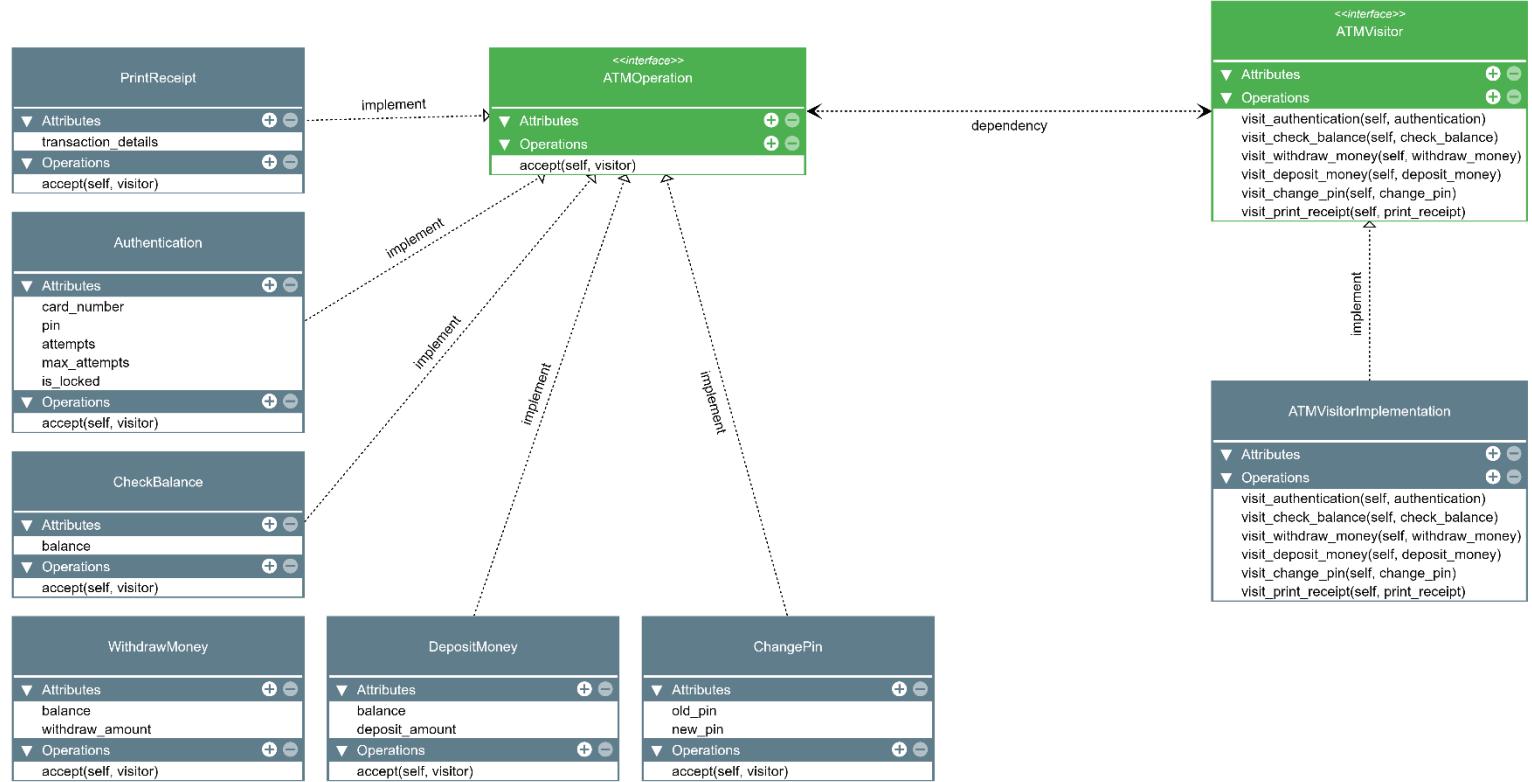
ใช้ GenTest หลังจาก Gen code เสร็จ

Write pytest to test your given code that has 100% statement coverage and 100% branch coverage.

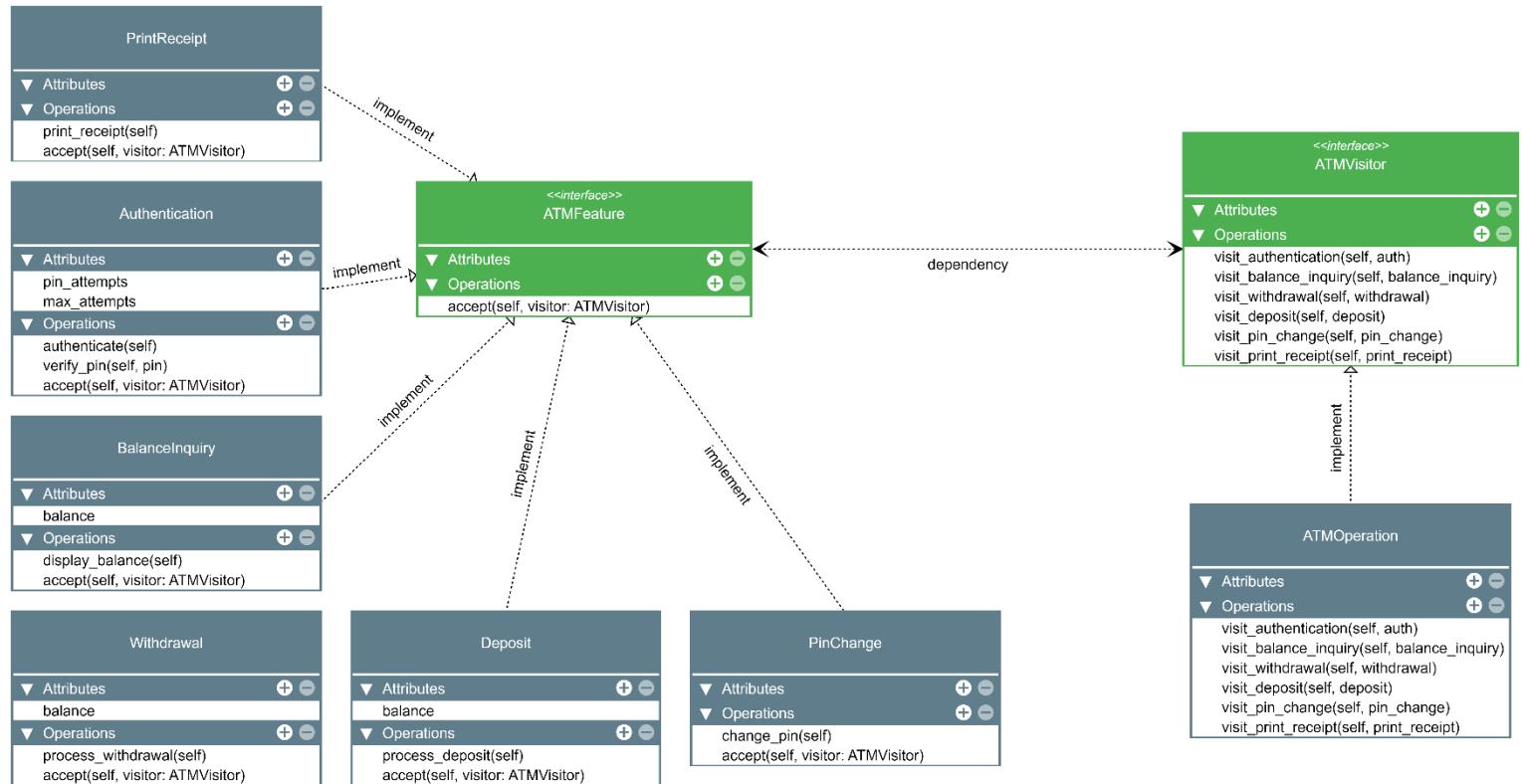
ผลลัพธ์ ในรอบที่ 1



ผลลัพธ์ ในรอบที่ 2



ผลลัพธ์ ในรอบที่ 3



วิเคราะห์ผลการทดลองการ Generate Code In Python Programming Language by Chat-GPT 4o

จากผลลัพธ์ พบร่วมกับ Chat-GPT 4o สามารถสร้างโค้ดถูกต้องตาม requirement แต่รายละเอียดการทำงานของฟังก์ชันนั้นจะไม่ชัดเจนเป็นเพียงแค่ตัวอย่างว่าเมื่อฟังก์ชันได้รับ input เข้ามาแล้วหรือถูกเรียกใช้ ฟังก์ชันจะตอบสนองและส่งคืนค่ากลับหรือแสดงผลอย่างไร ตาม requirement ที่ระบุ

Chat-GPT 4o สร้างโค้ดขึ้นโดยใช้ภาษาโปรแกรมและ Design pattern ที่ระบุได้ถูกต้องทั้ง Composite Design Pattern และ Visitor Design แต่มีผิดพลาดอยู่ 1 ผลลัพธ์ นั่นคือ

ผลลัพธ์ในรอบที่ 1 Visitor Design Pattern ที่สร้างโค้ดผิดจากข้อกำหนดของ Visitor Design Pattern เนื่องจากไม่มีการสร้าง interface Element แต่เป็นการสร้าง Concrete Element Class (ATM Account Transaction) ขึ้นมาเรียกใช้ Visitor โดยตรง ซึ่งถือว่าผิดจากข้อกำหนดของ Visitor Design Pattern

Prompt ที่ใช้ในการ Generate Code Java และ Generate Test ทั้งสามรอบ โดยใช้ Composite Design Pattern ในการ Gen Code

Use the composite design pattern to Write code in Java programming language from the requirements below.

"ความต้องการระบบ ATM

1. การยืนยันตัวตนผู้ใช้

การป้อนข้อมูล: ผู้ใช้ต้องสามารถใส่บัตร ATM และรหัส PIN ได้

การตรวจสอบ: ระบบต้องตรวจสอบความถูกต้องของบัตรและรหัส PIN

การจำกัดความพยายาม: หากใส่ผิดเกินจำนวนครั้งที่กำหนด บัตรจะถูกยกยื่ด

2. การแสดงยอดเงินคงเหลือ

การเข้าถึงข้อมูล: ผู้ใช้สามารถเลือกดูยอดเงินคงเหลือในบัญชีได้

การแสดงผล: ระบบต้องแสดงยอดเงินคงเหลือปัจจุบัน

3. การถอนเงิน

การเลือกจำนวนเงิน: ผู้ใช้สามารถเลือกจำนวนเงินที่ต้องการถอนได้

การตรวจสอบยอดเงิน: ระบบต้องตรวจสอบว่ามียอดเงินเพียงพอสำหรับการถอน

การจ่ายเงิน: หากมียอดเงินเพียงพอ ระบบจะจ่ายเงินให้ผู้ใช้ และปรับปรุงยอดเงินคงเหลือ

การแจ้งเตือน: หากมียอดเงินไม่เพียงพอ ระบบจะแจ้งเตือนผู้ใช้

4. การฝากเงิน

การรับเงินสด: ผู้ใช้สามารถใส่เงินสดเข้าเครื่อง ATM ได้

การนับเงิน: ระบบต้องนับจำนวนเงินที่ฝากเข้ามา

การปรับปรุงยอดเงิน: ระบบต้องปรับปรุงยอดเงินคงเหลือในบัญชี

5. การเปลี่ยนรหัส PIN

การเข้าถึงฟังก์ชัน: ผู้ใช้สามารถเปลี่ยนรหัส PIN ของตนเองได้

การยืนยัน: ระบบต้องขอรหัส PIN เก่าและรหัส PIN ใหม่ 2 ครั้งเพื่อยืนยัน

การเปลี่ยนรหัส: หากรหัส PIN ใหม่ตรงกัน ระบบจะเปลี่ยนรหัส PIN ให้

6. การพิมพ์สลิป

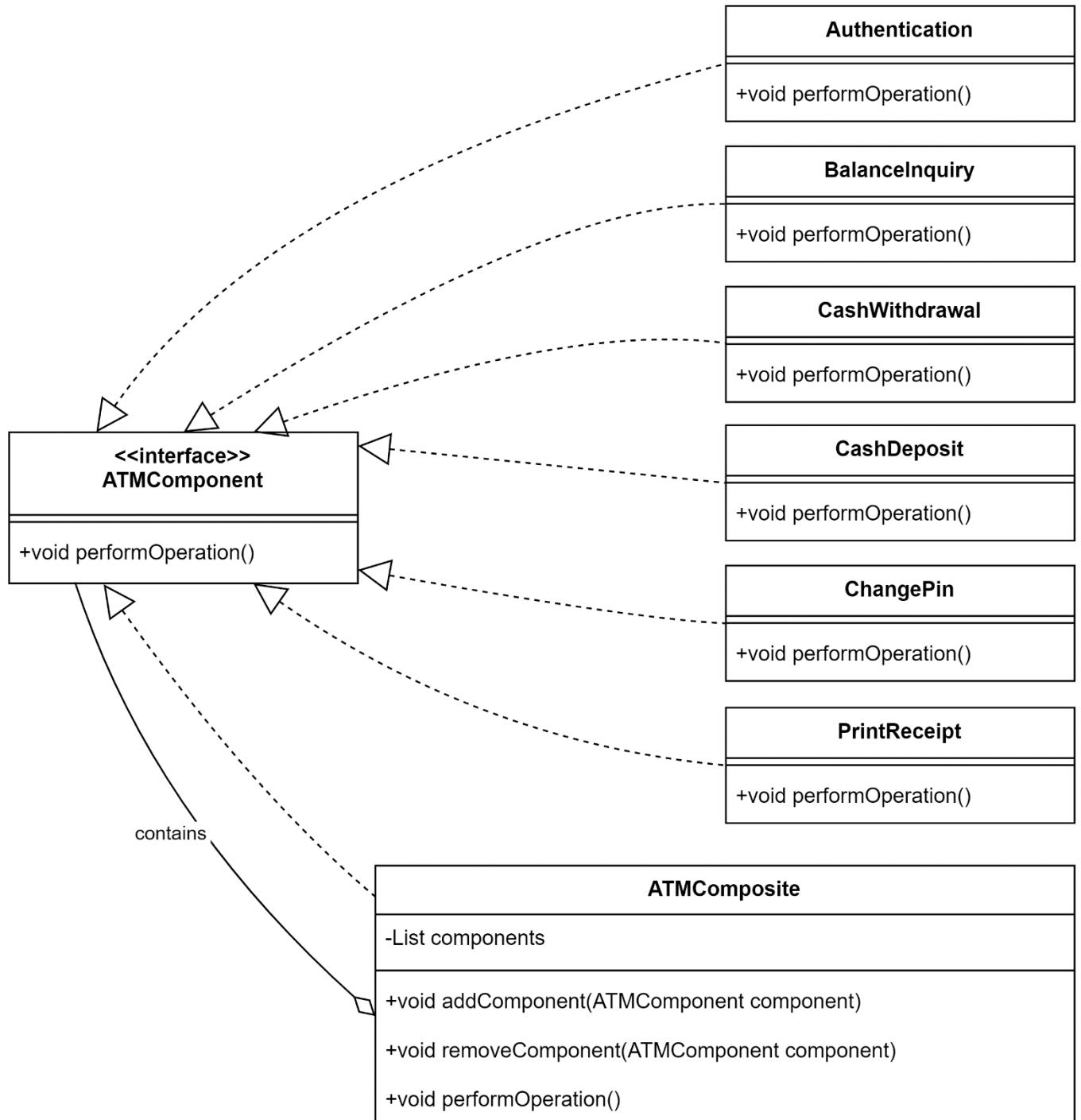
การเลือกพิมพ์: หลังจากทำการเสร็จสิ้น ผู้ใช้สามารถเลือกพิมพ์สลิปได้

เนื้อหาสลิป: สลิปต้องแสดงข้อมูลรายละเอียดของรายการ เช่น วันที่ เวลา จำนวนเงิน และยอดเงินคงเหลือ"

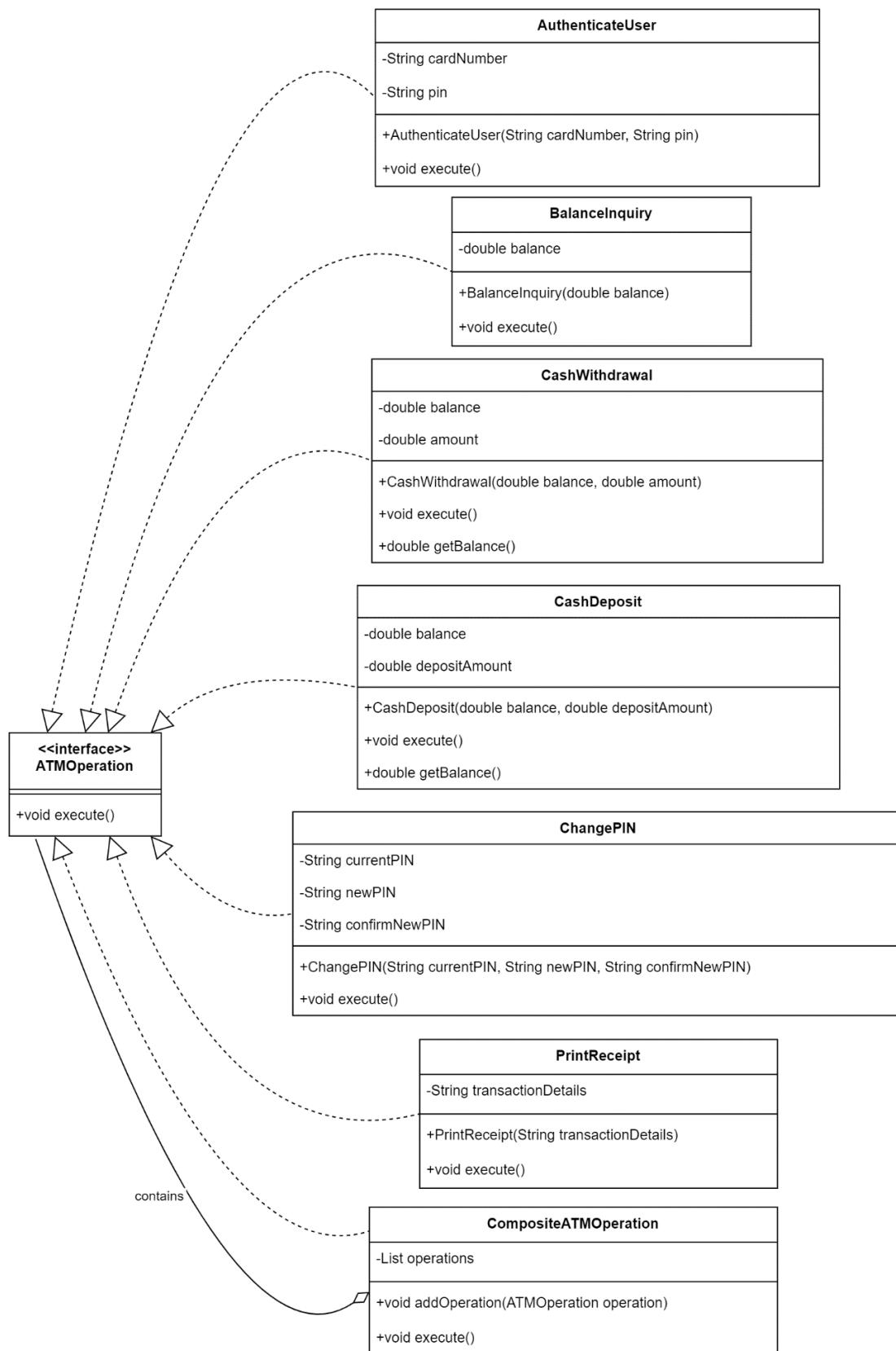
ใช้ GenTest หลังจาก Gen code เสร็จ

Write a JUnit to test your given code that has 100% statement coverage and 100% branch coverage.

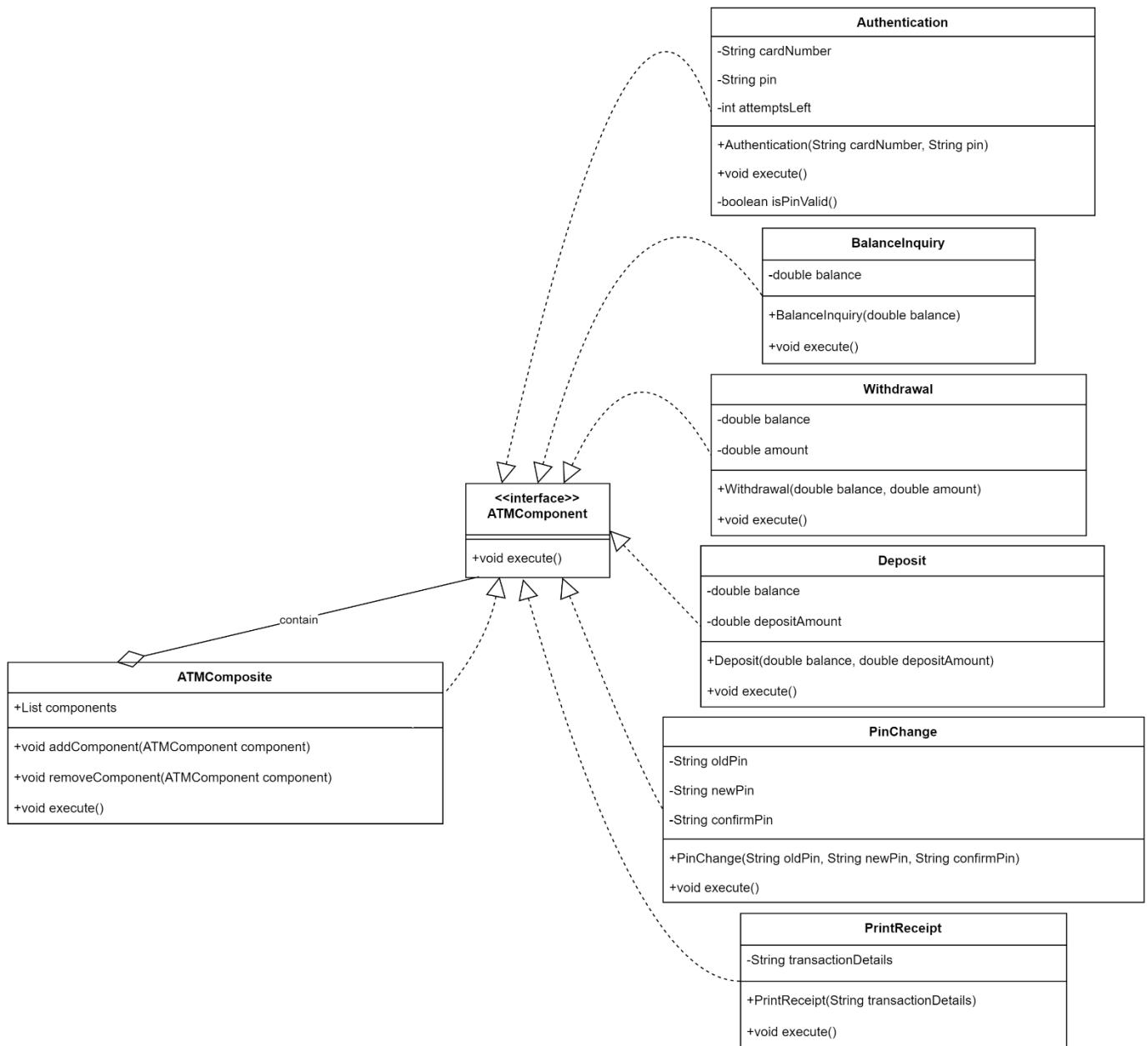
ผลลัพธ์ในรอบที่ 1



ผลลัพธ์ในรอบที่ 2



ผลลัพธ์ในรอบที่ 3



Prompt ที่ใช้ในการ Generate Code Java และ Generate Test ทั้งสามรอบ โดยใช้ Visitor Design Pattern ในการ Gen Code

Use the visitor design pattern to Write code in Java programming language from the requirements below.

"ความต้องการระบบ ATM

1. การยืนยันตัวตนผู้ใช้

การป้อนข้อมูล: ผู้ใช้ต้องสามารถใส่บัตร ATM และรหัส PIN ได้

การตรวจสอบ: ระบบต้องตรวจสอบความถูกต้องของบัตรและรหัส PIN

การจำกัดความพยายาม: หากใส่ผิดเกินจำนวนครั้งที่กำหนด บัตรจะถูกยึด

2. การแสดงยอดเงินคงเหลือ

การเข้าถึงข้อมูล: ผู้ใช้สามารถเลือกดูยอดเงินคงเหลือในบัญชีได้

การแสดงผล: ระบบต้องแสดงยอดเงินคงเหลือปัจจุบัน

3. การถอนเงิน

การเลือกจำนวนเงิน: ผู้ใช้สามารถเลือกจำนวนเงินที่ต้องการถอนได้

การตรวจสอบยอดเงิน: ระบบต้องตรวจสอบว่ามียอดเงินเพียงพอสำหรับการถอน

การจ่ายเงิน: หากมียอดเงินเพียงพอ ระบบจะจ่ายเงินให้ผู้ใช้ และปรับปรุงยอดเงินคงเหลือ

การแจ้งเตือน: หากมียอดเงินไม่เพียงพอ ระบบจะแจ้งเตือนผู้ใช้

4. การฝากเงิน

การรับเงินสด: ผู้ใช้สามารถใส่เงินสดเข้าเครื่อง ATM ได้

การนับเงิน: ระบบต้องนับจำนวนเงินที่ฝากเข้ามา

การปรับปรุงยอดเงิน: ระบบต้องปรับปรุงยอดเงินคงเหลือในบัญชี

5. การเปลี่ยนรหัส PIN

การเข้าถึงฟังก์ชัน: ผู้ใช้สามารถเปลี่ยนรหัส PIN ของตนเองได้

การยืนยัน: ระบบต้องขอรหัส PIN เก่าและรหัส PIN ใหม่ 2 ครั้งเพื่อยืนยัน

การเปลี่ยนรหัส: หากรหัส PIN ใหม่ตรงกัน ระบบจะเปลี่ยนรหัส PIN ให้

6. การพิมพ์สลิป

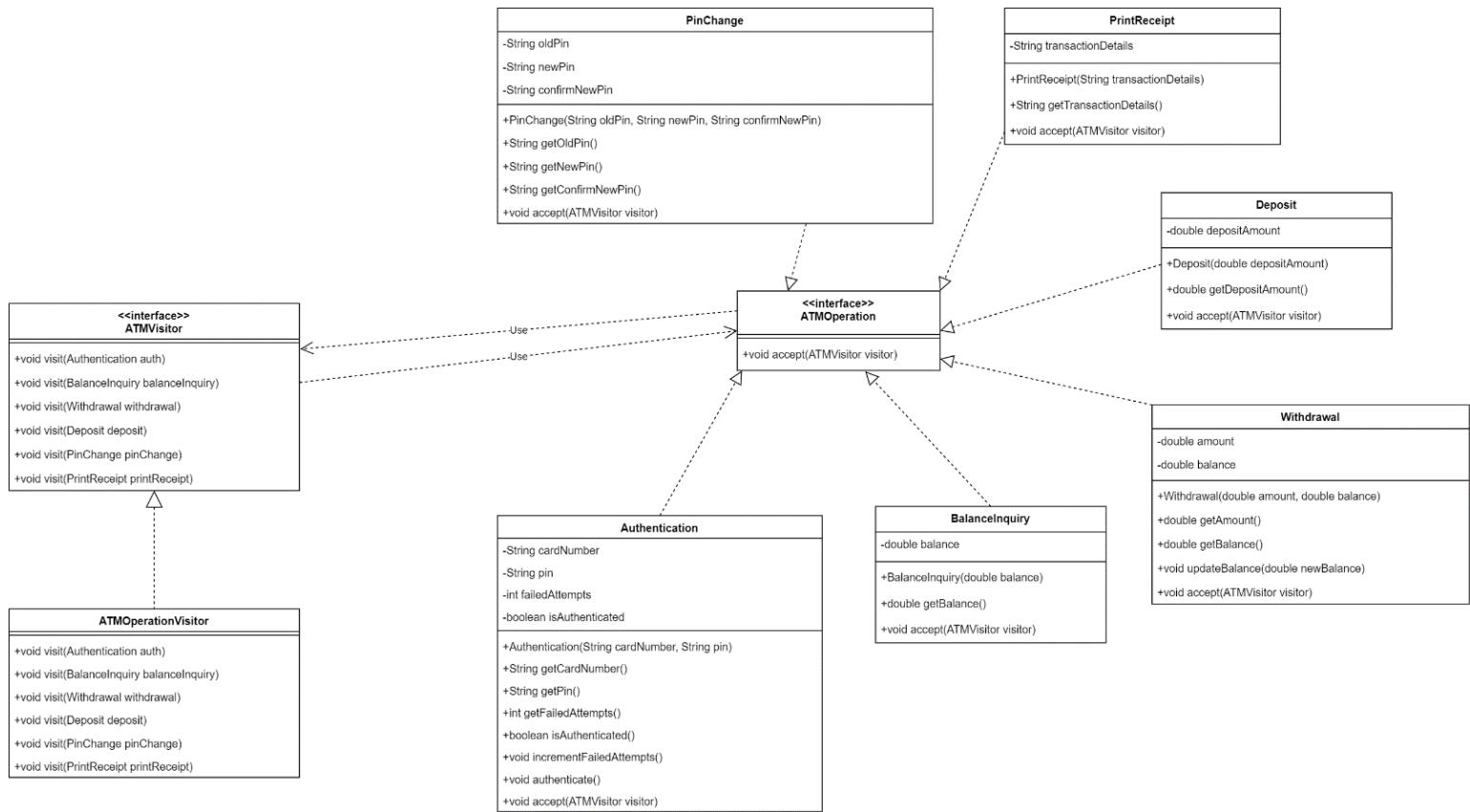
การเลือกพิมพ์: หลังจากทำการเสร็จสิ้น ผู้ใช้สามารถเลือกพิมพ์สลิปได้

เนื้อหาสลิป: สลิปต้องแสดงข้อมูลรายละเอียดของรายการ เช่น วันที่ เวลา จำนวนเงิน และยอดเงินคงเหลือ"

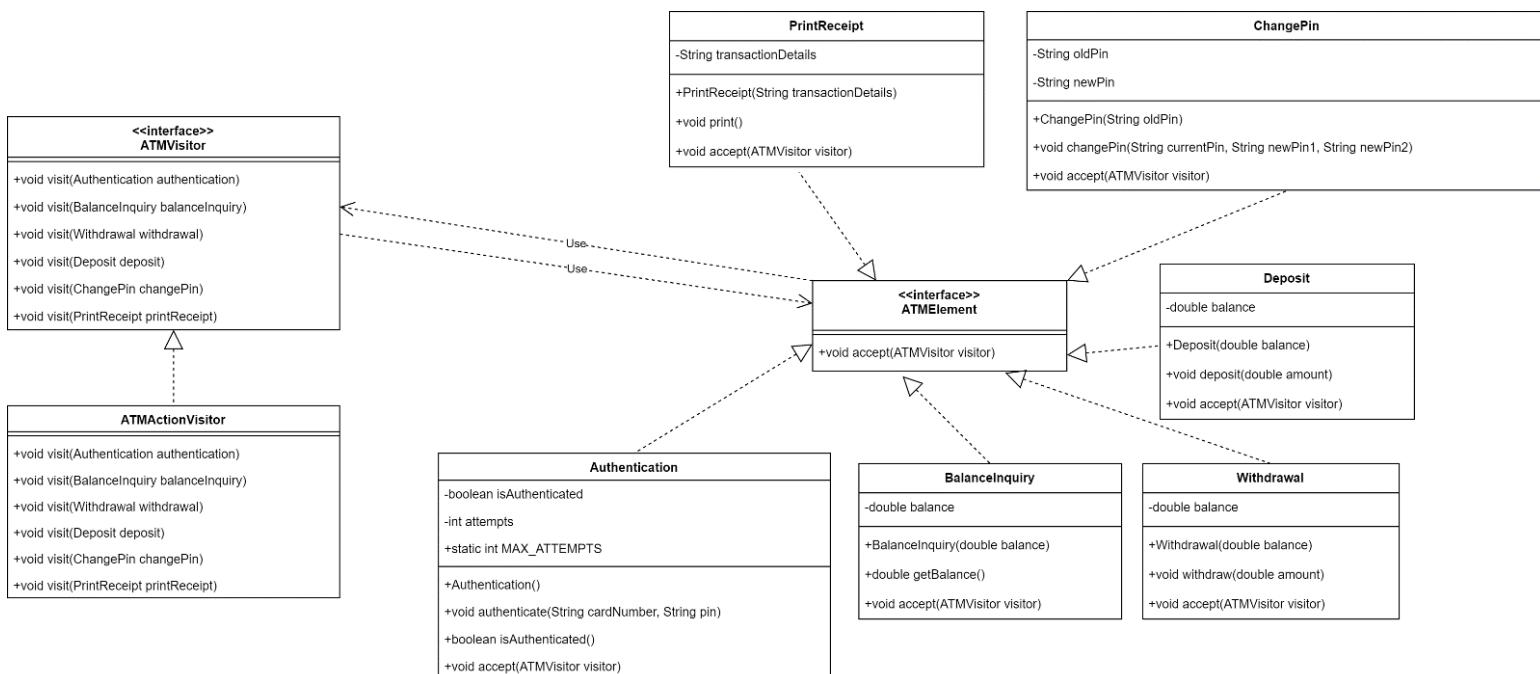
ใช้ GenTest หลังจาก Gen code เสร็จ

Write a JUnit to test your given code that has 100% statement coverage and 100% branch coverage.

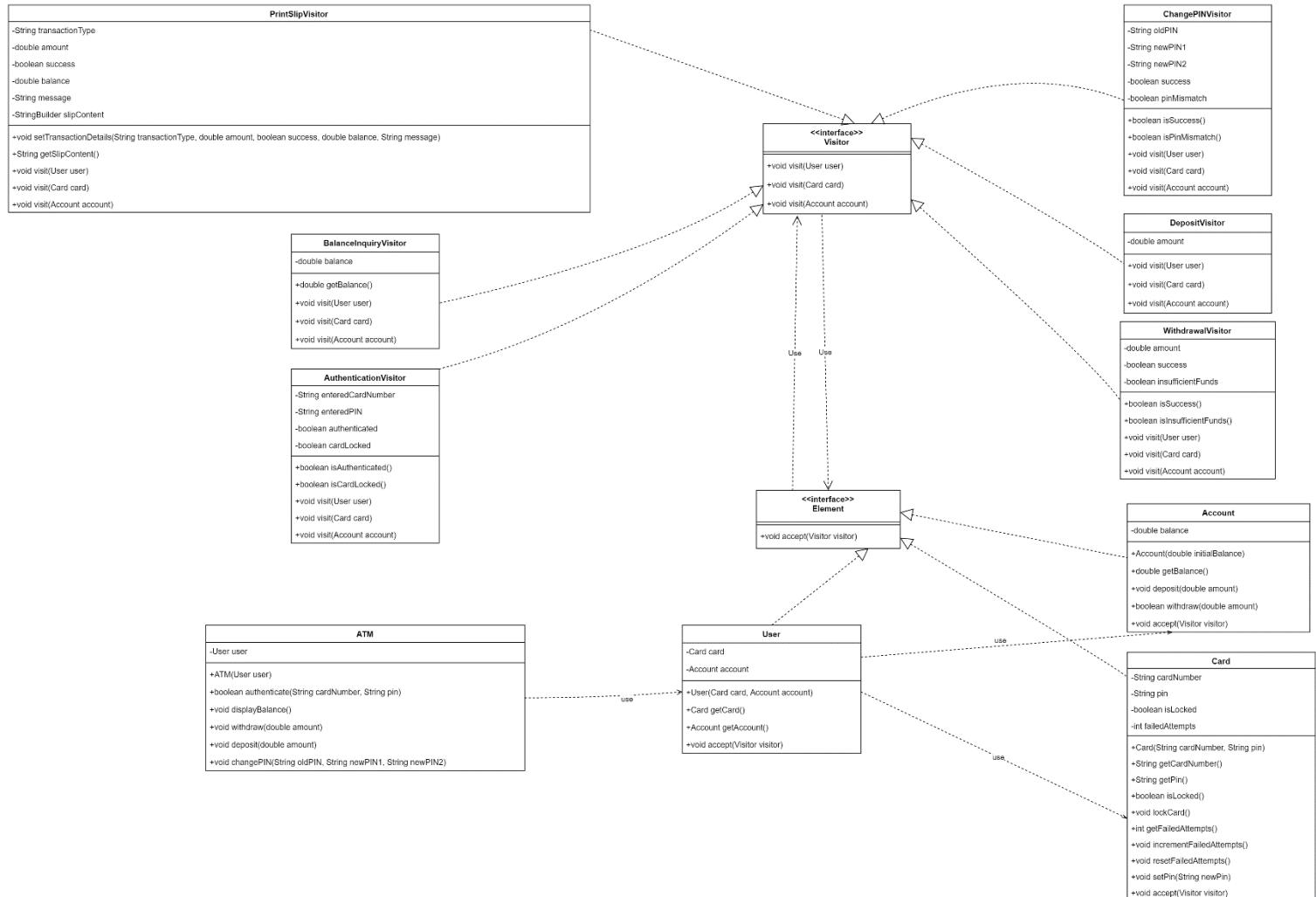
ผลลัพธ์ในรอบที่ 1



ผลลัพธ์ในรอบที่ 2



ผลลัพธ์ในรอบที่ 3



วิเคราะห์ผลการทดลอง Generate Code In Java Programming Language by Chat-GPT 4o

จากผลลัพธ์ พบร่วมกับ Chat-GPT 4o สามารถสร้างโค้ดถูกต้องตาม Requirement แต่รายละเอียดการทำงานของฟังก์ชันนั้นจะไม่ซับซ้อนเป็นเพียงแค่ตัวอย่างว่าเมื่อฟังก์ชันได้รับ input เข้ามาแล้วหรือถูกเรียกใช้ ฟังก์ชันจะตอบสนองและส่งคืนค่ากลับหรือแสดงผลอย่างไร ตาม Requirement ที่ระบุ

Chat-GPT 4o สร้างโค้ดขึ้นโดยใช้ภาษาโปรแกรมและ Design pattern ที่ระบุได้ถูกต้องทั้ง Composite Design Pattern และ Visitor Design

มีผลลัพธ์ในรอบที่ 3 Visitor Design Pattern ที่รูปแบบโค้ดดูซับซ้อนเพิ่มขึ้น

สรุปผลการทดลอง Generate Code by Chat-GPT 4o ทั้ง ภาษา Python และ Java

จากการทดลองส่วนใหญ่พบว่า Chat-GPT สามารถช่วยออกแบบระบบตาม Requirement และ Design Pattern ได้จริงแต่การทำงานของระบบจะไม่ซับซ้อนมากซึ่งในการ Generate Code ครั้งแรกนั้นอาจมีจุดผิดพลาดบางส่วนในการออกแบบตาม Design Pattern ที่กำหนดไว้ และ ความซับซ้อนของฟังก์ชันในการทำงานในระบบน้อยไป แต่ในรอบการทดลองครั้งถัดมาเกิดการแก้ไขให้สมบูรณ์ขึ้น

โดยสรุป Chat-GPT 4o สามารถเป็นผู้ช่วยในการสร้าง ออกแบบ หรือ นำร่องการพัฒนาโปรแกรมได้ในระดับหนึ่ง ซึ่งนักพัฒนานั้นควรจะมีทักษะเฉพาะทาง และตรวจสอบผลลัพธ์ที่ได้จาก Generative AI Chat-GPT อย่างละเอียด ก่อนนำไปพัฒนาต่อ หรือใช้งานจริงเพื่อให้เกิดความถูกต้องสมบูรณ์ของระบบ

2. Gemini 1.5 Flash / Gemini 1.5 Pro

Prompt ที่ใช้ในการ Generate Code Python และ Generate Test ทั้งสามรอบ โดยใช้ Composite Design Pattern

We are developing a fully functional ATM system using the Python Programming Language and implementing the Composite Design Pattern. The system should meet the following requirements:

#System Requirements:

1)User Authentication:

1.1)Input: The system must allow users to insert their ATM card and enter a PIN.

1.2)Verification: The system must verify the correctness of the card and PIN.

1.3)Attempt Limitation: If the user enters the wrong PIN more than the allowed number of times, the card should be retained.

2)Balance Display:

2.1)Data Access: The user can view their account balance.

2.2)Display: The system must display the current balance.

3)Cash Withdrawal:

3.1)Amount Selection: The user can select the amount they wish to withdraw.

3.2)Balance Verification: The system must verify if there are sufficient funds for the withdrawal.

3.3)Dispensing Cash: If sufficient funds are available, the system will dispense the cash and update the balance.

3.4)Insufficient Funds Alert: If there are insufficient funds, the system will notify the user.

4)Deposit:

4.1)Cash Reception: The user can deposit cash into the ATM.

4.2)Money Counting: The system must count the deposited amount.

4.3)Balance Update: The system must update the account balance accordingly.

5)PIN Change:

5.1)Function Access: The user can change their PIN.

5.2)Confirmation: The system must request the old PIN and the new PIN (entered twice) for confirmation.

5.3)PIN Update: If the new PIN matches the confirmation, the system will update the user's PIN.

6)Receipt Printing:

6.1)Print Option: After completing a transaction, the user can choose to print a receipt.

6.2)Receipt Content: The receipt must show transaction details, such as date, time, amount, and the remaining balance.

#Testing Requirements:

Testing with pytest:

Develop unit tests to achieve 100% statement coverage and 100% branch coverage.

#Deliverables:

code.py: The full implementation of the ATM system.

unit_test.py: The pytest script with complete coverage tests.

Prompt ที่ใช้ในการ Generate Code Python และ Generate Test ทั้งสามรอบ โดยใช้ Visitor Design Pattern

We are developing a fully functional ATM system using the Python Programming Language and implementing the Visitor Design Pattern. The system should meet the following requirements:

#System Requirements:

1) User Authentication:

1.1) Input: The system must allow users to insert their ATM card and enter a PIN.

1.2) Verification: The system must verify the correctness of the card and PIN.

1.3) Attempt Limitation: If the user enters the wrong PIN more than the allowed number of times, the card should be retained.

2) Balance Display:

2.1) Data Access: The user can view their account balance.

2.2) Display: The system must display the current balance.

3) Cash Withdrawal:

3.1) Amount Selection: The user can select the amount they wish to withdraw.

3.2) Balance Verification: The system must verify if there are sufficient funds for the withdrawal.

3.3) Dispensing Cash: If sufficient funds are available, the system will dispense the cash and update the balance.

3.4) Insufficient Funds Alert: If there are insufficient funds, the system will notify the user.

4) Deposit:

4.1) Cash Reception: The user can deposit cash into the ATM.

4.2) Money Counting: The system must count the deposited amount.

4.3) Balance Update: The system must update the account balance accordingly.

5) PIN Change:

5.1) Function Access: The user can change their PIN.

5.2) Confirmation: The system must request the old PIN and the new PIN (entered twice) for confirmation.

5.3) PIN Update: If the new PIN matches the confirmation, the system will update the user's PIN.

6) Receipt Printing:

6.1) Print Option: After completing a transaction, the user can choose to print a receipt.

6.2) Receipt Content: The receipt must show transaction details, such as date, time, amount, and the remaining balance.

#Testing Requirements:

Testing with pytest:

Develop unit tests to achieve 100% statement coverage and 100% branch coverage.

#Deliverables:

code.py: The full implementation of the ATM system.

unit_test.py: The pytest script with complete coverage tests.

Prompt ที่ใช้ในการ Generate Code Java และ Generate Test ทั้งสามรอบ โดยใช้ Composite Design Pattern

We are developing a fully functional ATM system using the Java Programming Language and implementing the Composite Design Pattern. The system should meet the following requirements:

#System Requirements:

1) User Authentication:

1.1) Input: The system must allow users to insert their ATM card and enter a PIN.

1.2) Verification: The system must verify the correctness of the card and PIN.

1.3) Attempt Limitation: If the user enters the wrong PIN more than the allowed number of times, the card should be retained.

2) Balance Display:

2.1) Data Access: The user can view their account balance.

2.2) Display: The system must display the current balance.

3) Cash Withdrawal:

3.1) Amount Selection: The user can select the amount they wish to withdraw.

3.2) Balance Verification: The system must verify if there are sufficient funds for the withdrawal.

3.3) Dispensing Cash: If sufficient funds are available, the system will dispense the cash and update the balance.

3.4) Insufficient Funds Alert: If there are insufficient funds, the system will notify the user.

4) Deposit:

4.1) Cash Reception: The user can deposit cash into the ATM.

4.2) Money Counting: The system must count the deposited amount.

4.3) Balance Update: The system must update the account balance accordingly.

5) PIN Change:

5.1) Function Access: The user can change their PIN.

5.2) Confirmation: The system must request the old PIN and the new PIN (entered twice) for confirmation.

5.3) PIN Update: If the new PIN matches the confirmation, the system will update the user's PIN.

6) Receipt Printing:

6.1) Print Option: After completing a transaction, the user can choose to print a receipt.

6.2) Receipt Content: The receipt must show transaction details, such as date, time, amount, and the remaining balance.

#Testing Requirements:

Testing with JUnit5:

Develop unit tests to achieve 100% statement coverage and 100% branch coverage.

#Deliverables:

code : The full implementation of the ATM system.

unit_test : The JUnit5 script with complete coverage tests.

Prompt ที่ใช้ในการ Generate Code Java และ Generate Test ห้องสมุด โดยใช้ Visitor Design Pattern

We are developing a fully functional ATM system using the Java Programming Language and implementing the Visitor Design Pattern. The system should meet the following requirements:

#System Requirements:

1)User Authentication:

1.1)Input: The system must allow users to insert their ATM card and enter a PIN.

1.2)Verification: The system must verify the correctness of the card and PIN.

1.3)Attempt Limitation: If the user enters the wrong PIN more than the allowed number of times, the card should be retained.

2)Balance Display:

2.1)Data Access: The user can view their account balance.

2.2)Display: The system must display the current balance.

3)Cash Withdrawal:

3.1)Amount Selection: The user can select the amount they wish to withdraw.

3.2)Balance Verification: The system must verify if there are sufficient funds for the withdrawal.

3.3)Dispensing Cash: If sufficient funds are available, the system will dispense the cash and update the balance.

3.4)Insufficient Funds Alert: If there are insufficient funds, the system will notify the user.

4)Deposit:

4.1)Cash Reception: The user can deposit cash into the ATM.

4.2)Money Counting: The system must count the deposited amount.

4.3)Balance Update: The system must update the account balance accordingly.

5)PIN Change:

5.1)Function Access: The user can change their PIN.

5.2)Confirmation: The system must request the old PIN and the new PIN (entered twice) for confirmation.

5.3)PIN Update: If the new PIN matches the confirmation, the system will update the user's PIN.

6)Receipt Printing:

6.1)Print Option: After completing a transaction, the user can choose to print a receipt.

6.2)Receipt Content: The receipt must show transaction details, such as date, time, amount, and the remaining balance.

#Testing Requirements:

Testing with JUnit5:

Develop unit tests to achieve 100% statement coverage and 100% branch coverage.

#Deliverables:

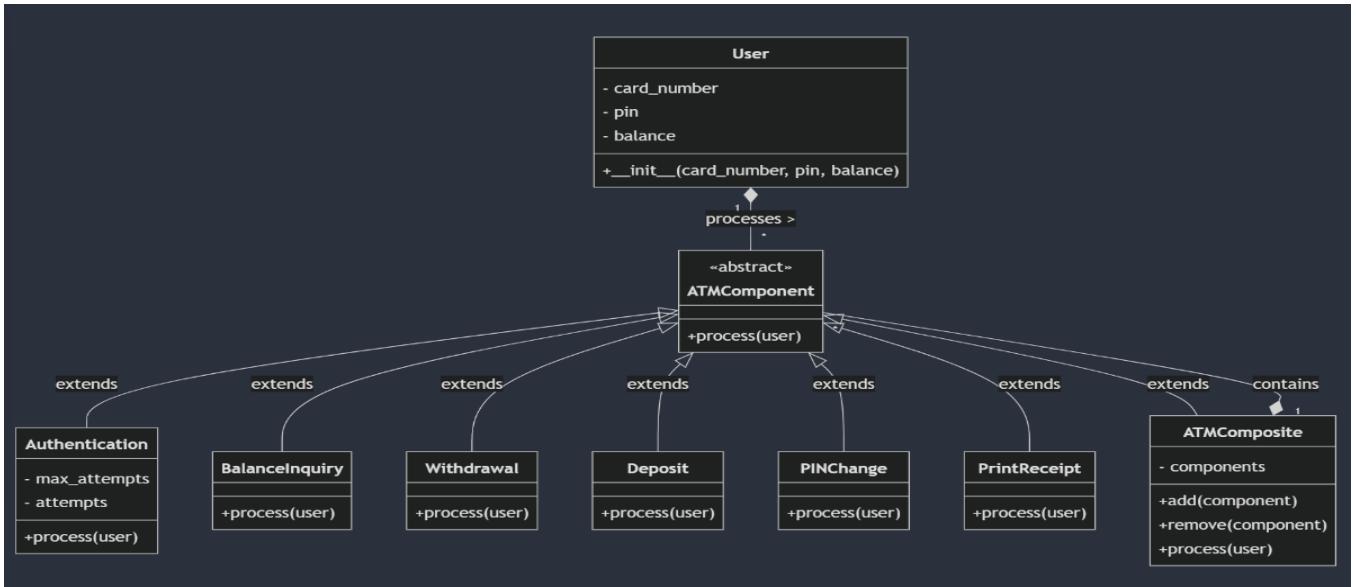
code : The full implementation of the ATM system.

unit_test : The JUnit5 script with complete coverage tests.

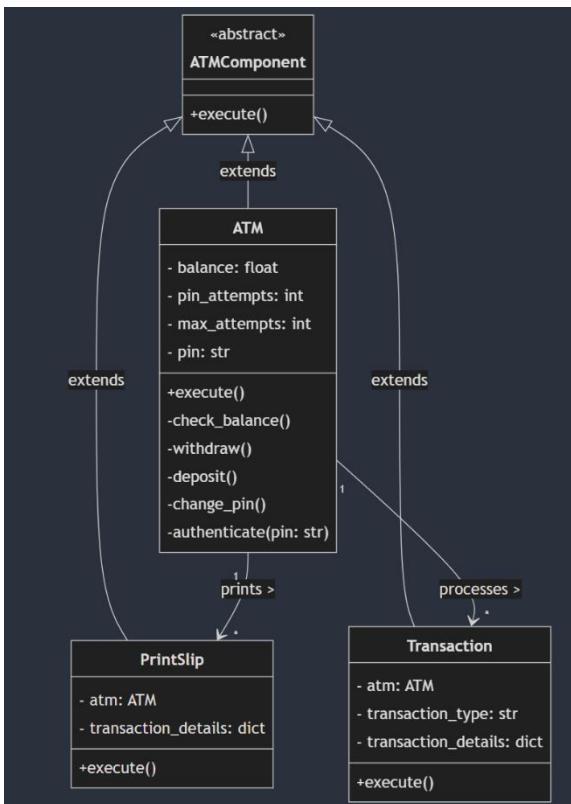
ผลลัพธ์ของ Python // Composite Design Pattern

โดย Gemini 1.5 Flash

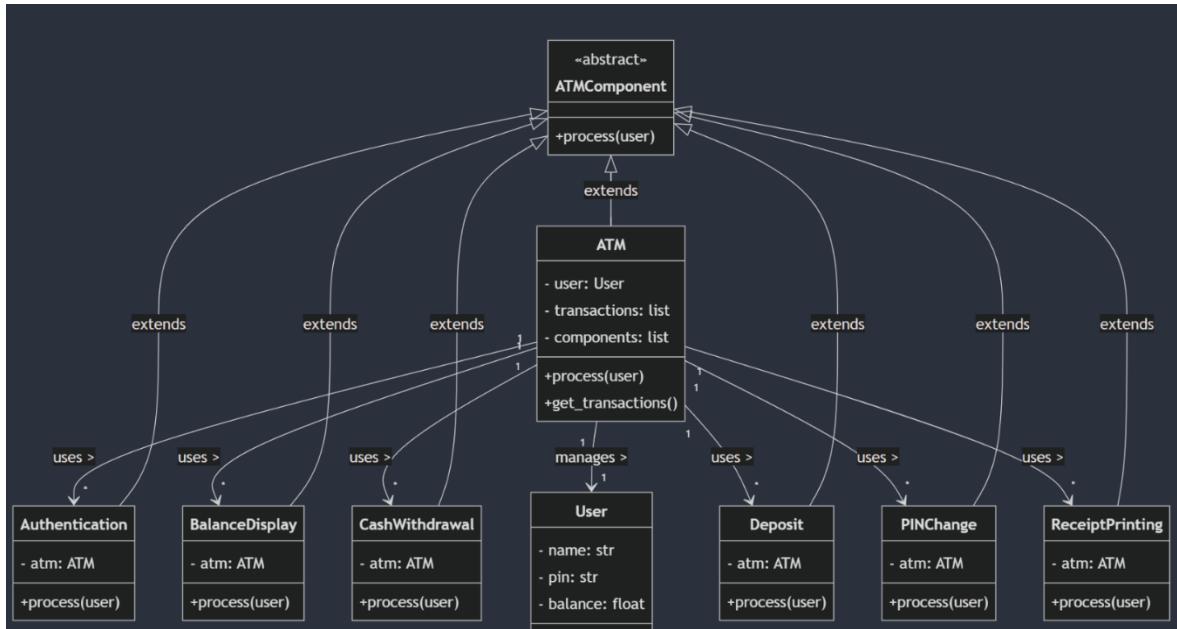
รอบที่ 1



รอบที่ 2

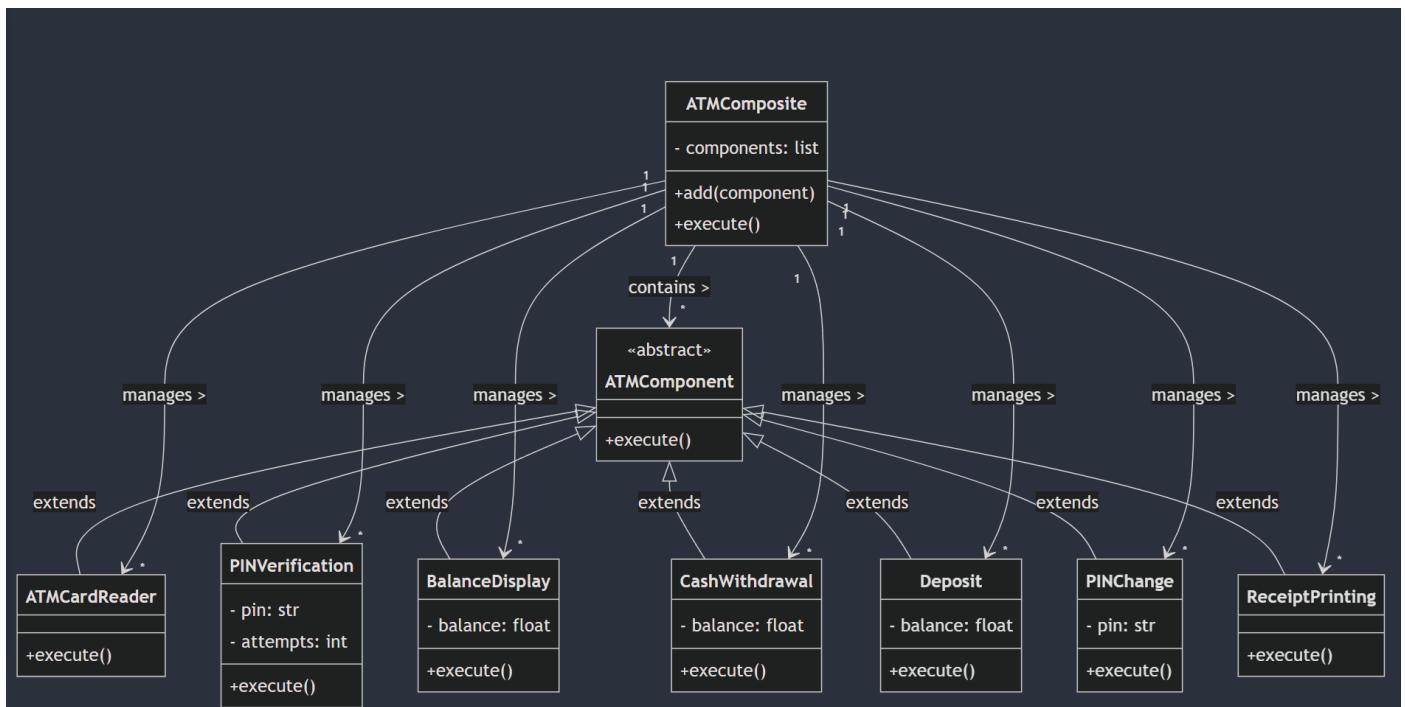


ຮອບທີ 3

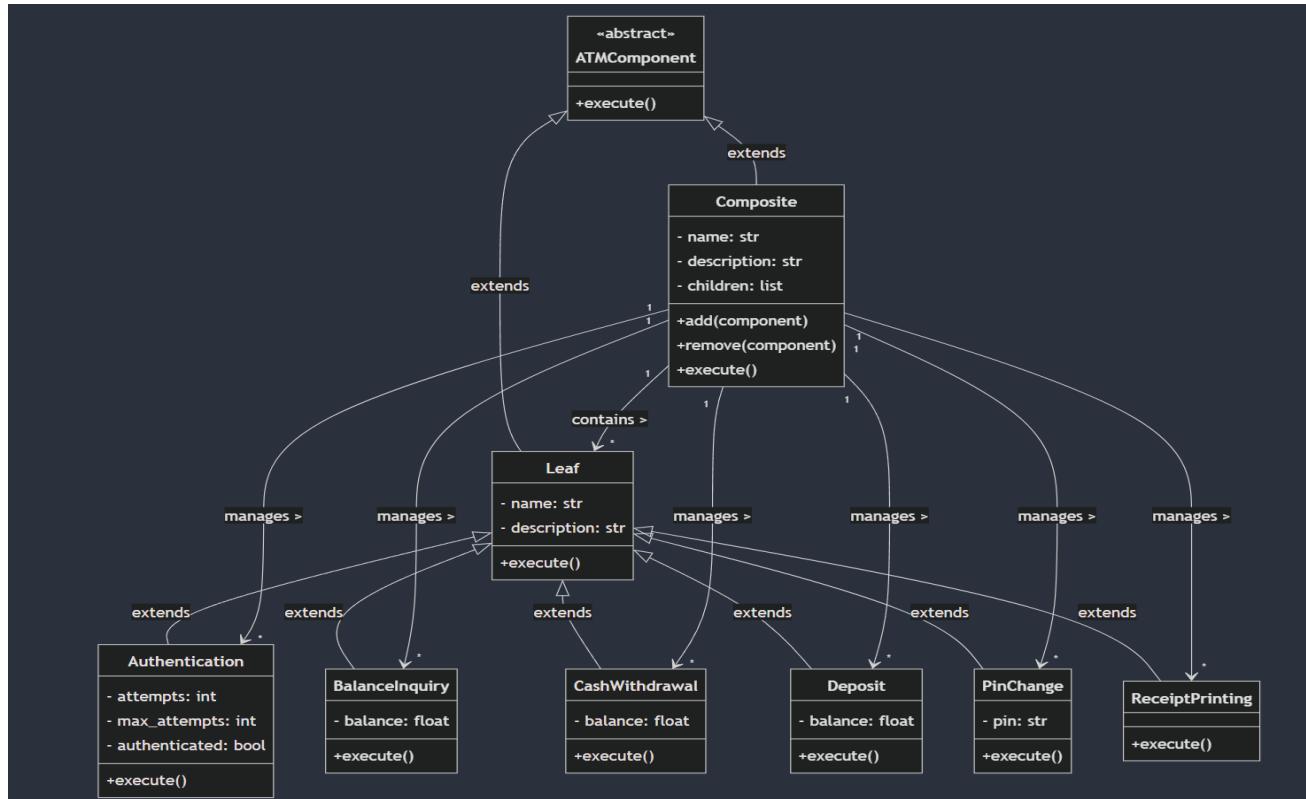


ໂດຍ Gemini 1.5 Pro

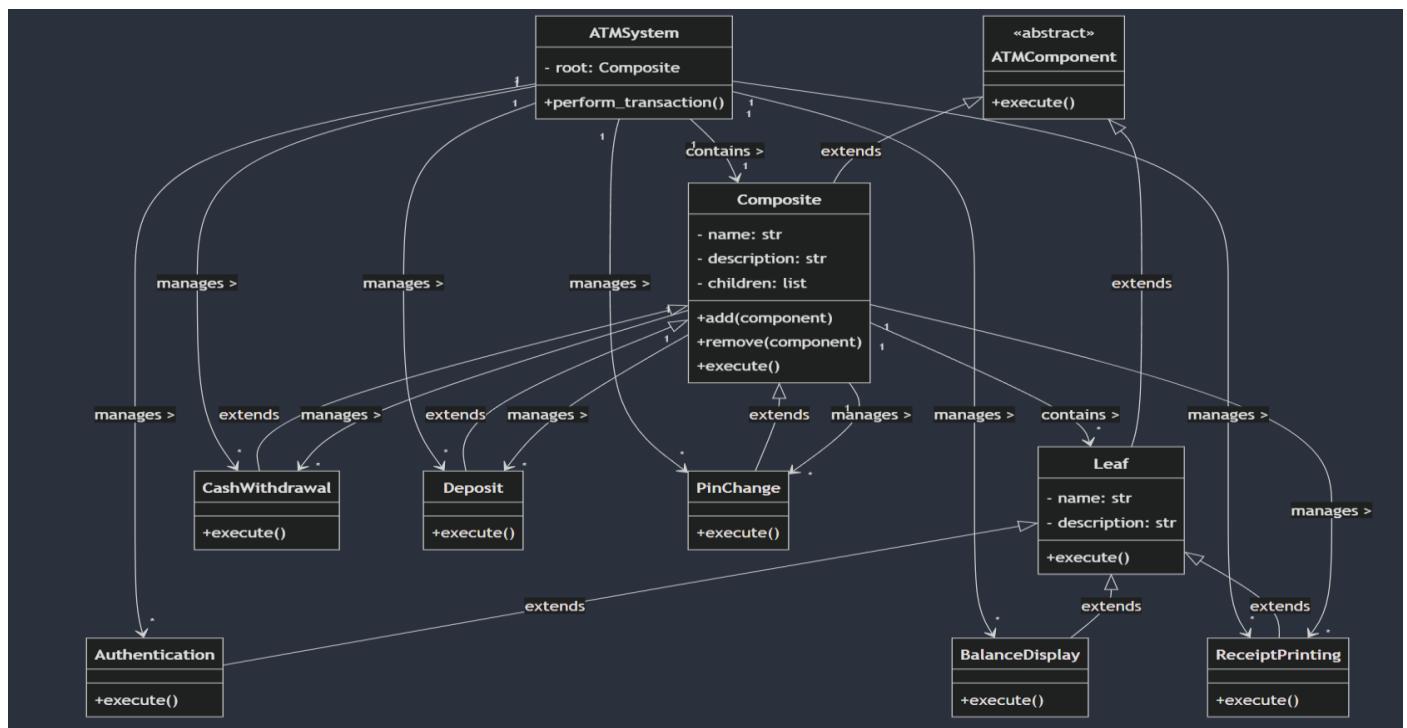
ຮອບທີ 1



รอบที่ 2



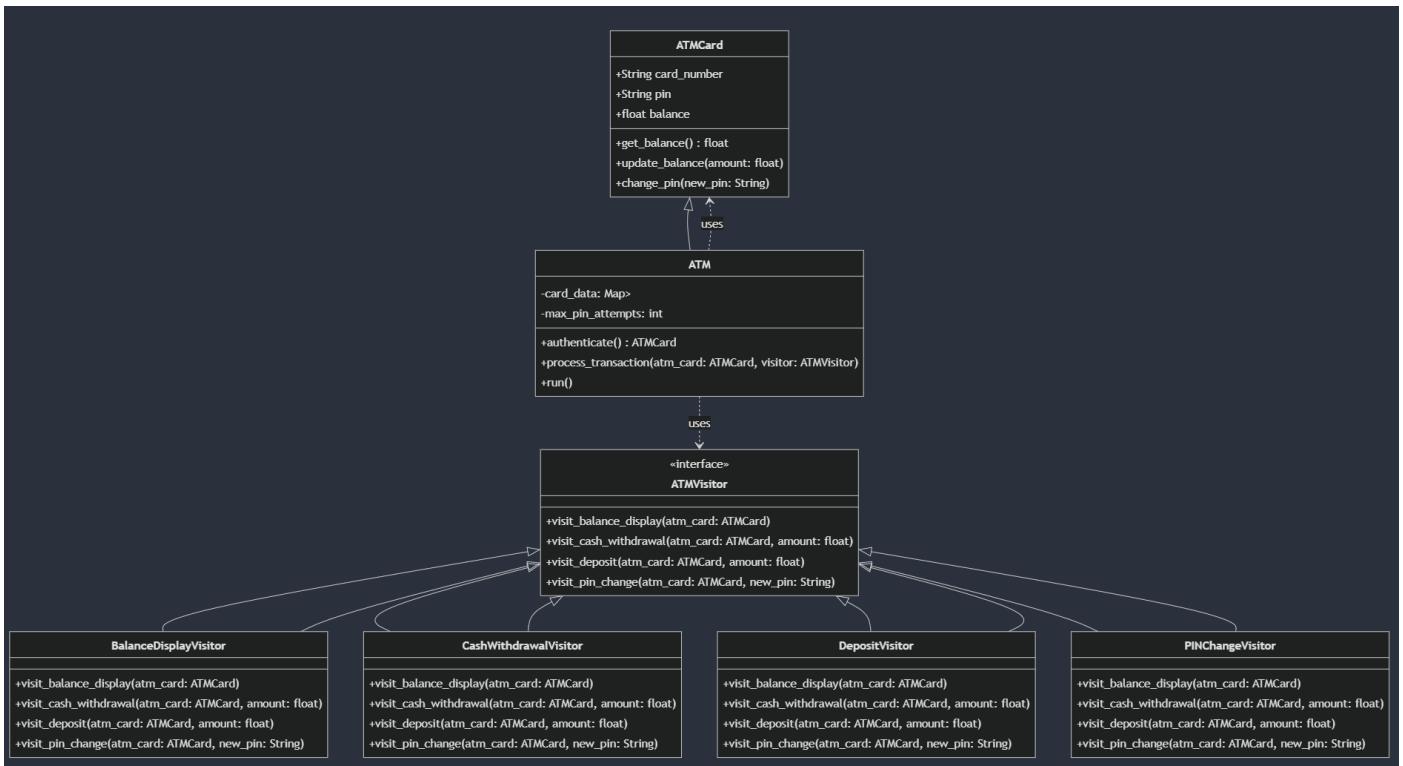
รอบที่ 3



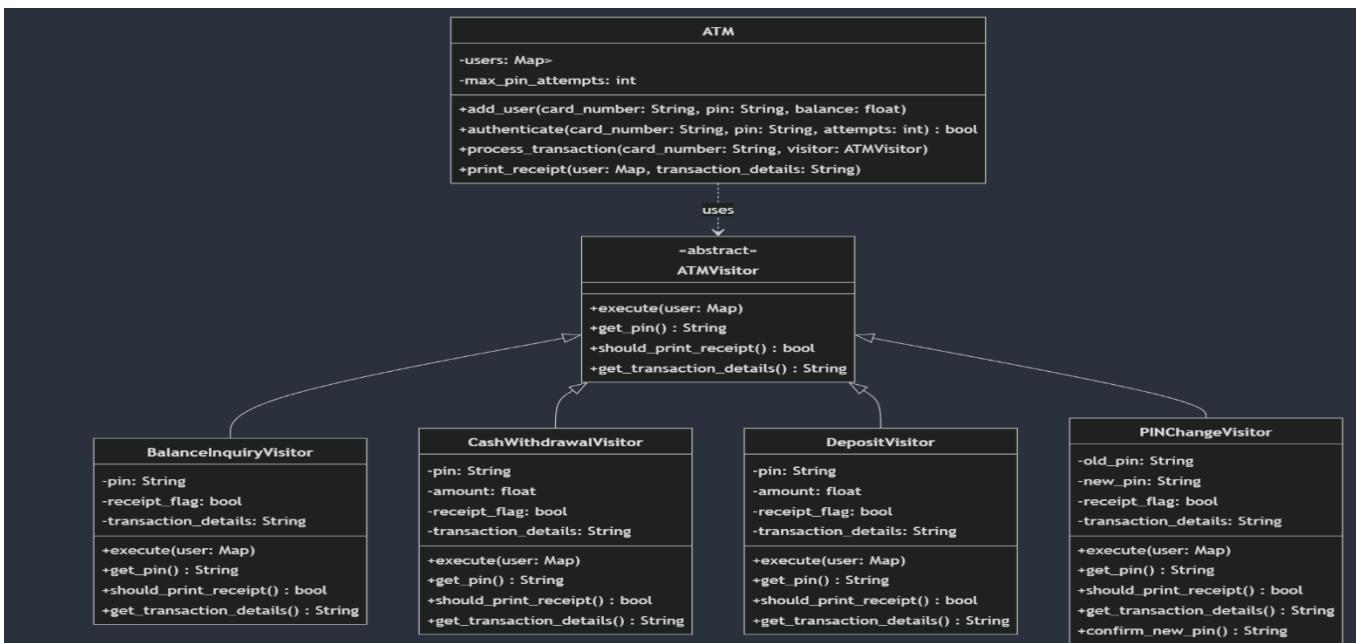
ผลลัพธ์ของ Python // Visitor Design Pattern

โดยใช้ Gemini 1.5 Flash

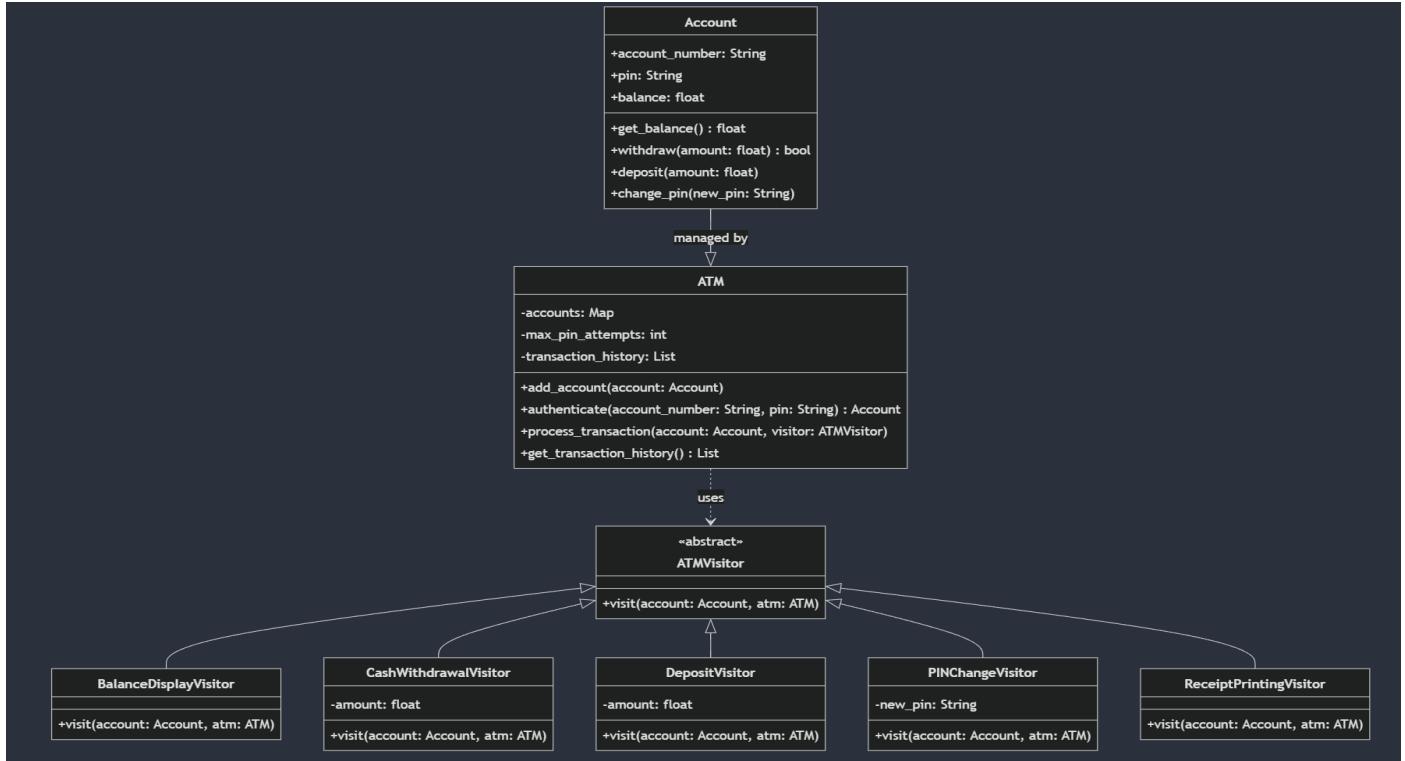
รอบที่ 1



รอบที่ 2

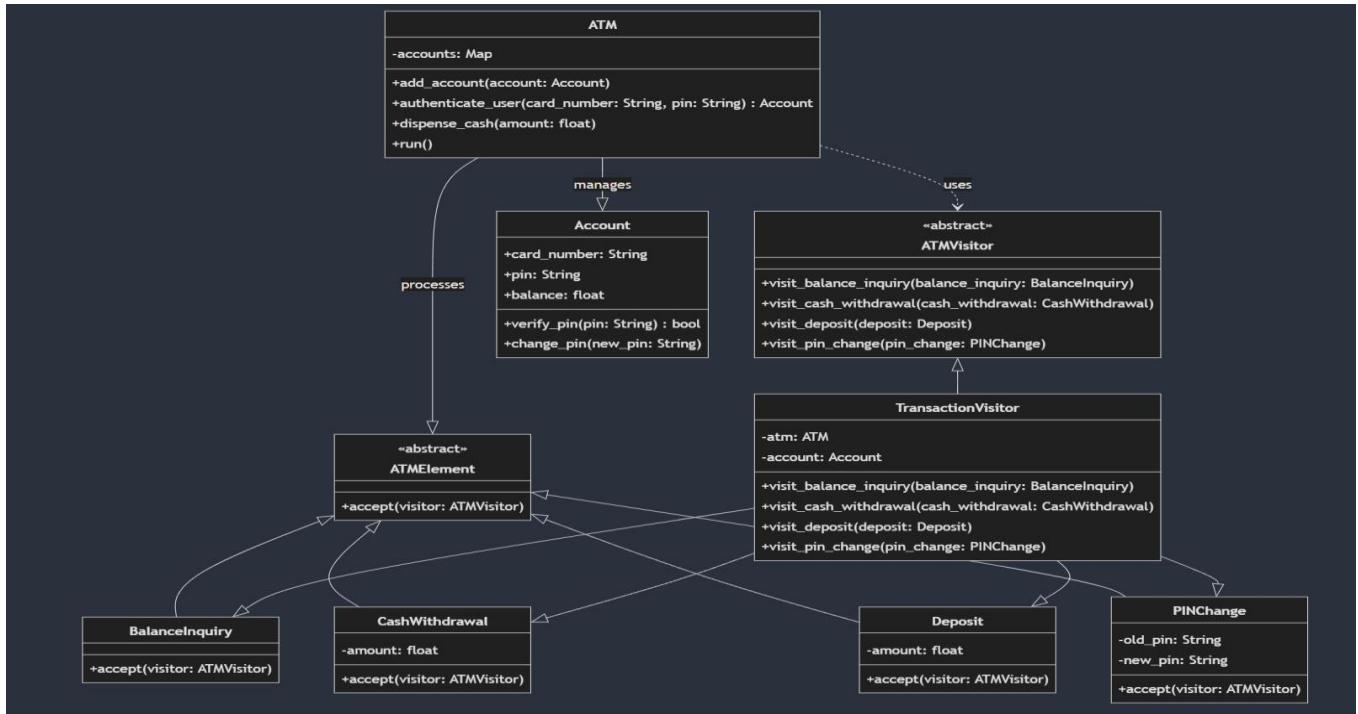


รอบที่ 3

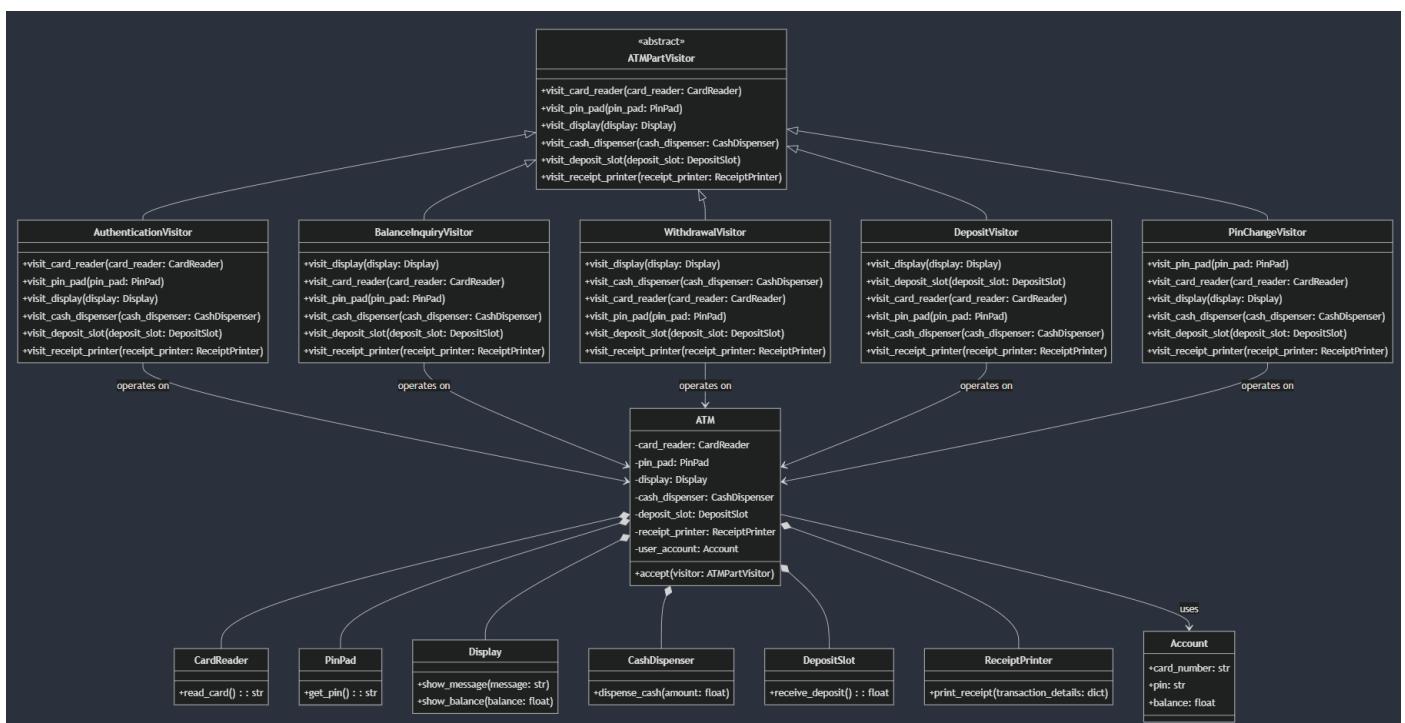


โดยใช้ Gemini 1.5 Pro

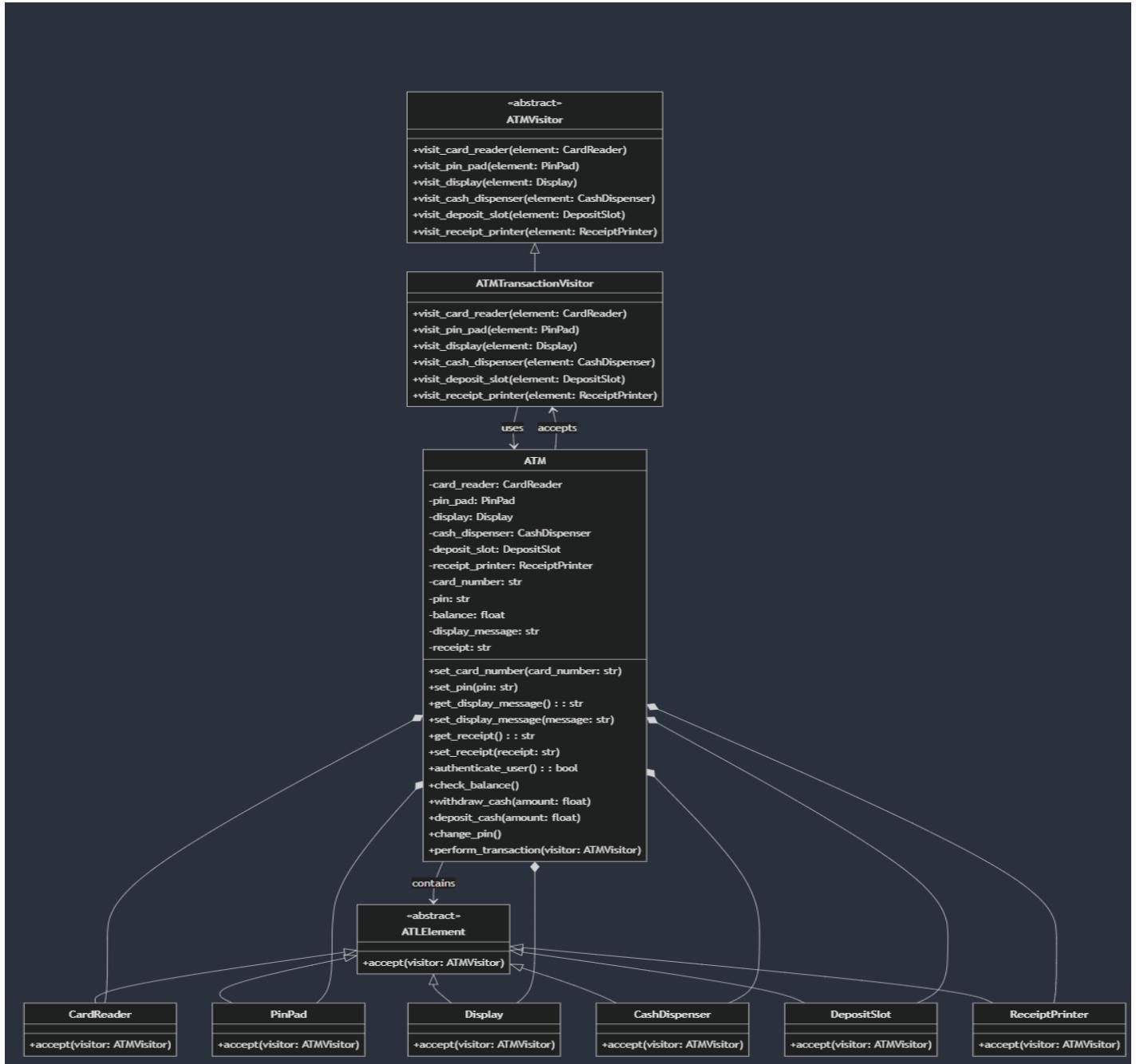
รอบที่ 1



รอบที่ 2



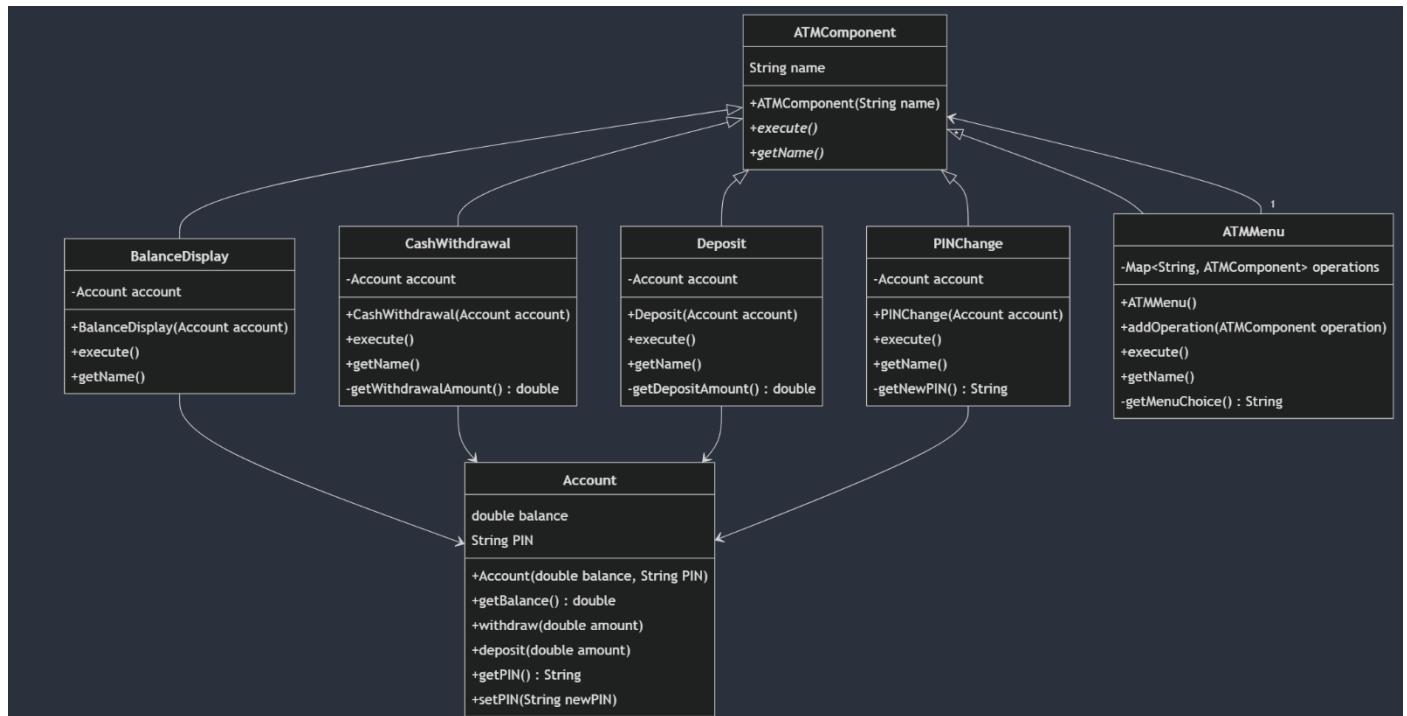
รอบที่ 3



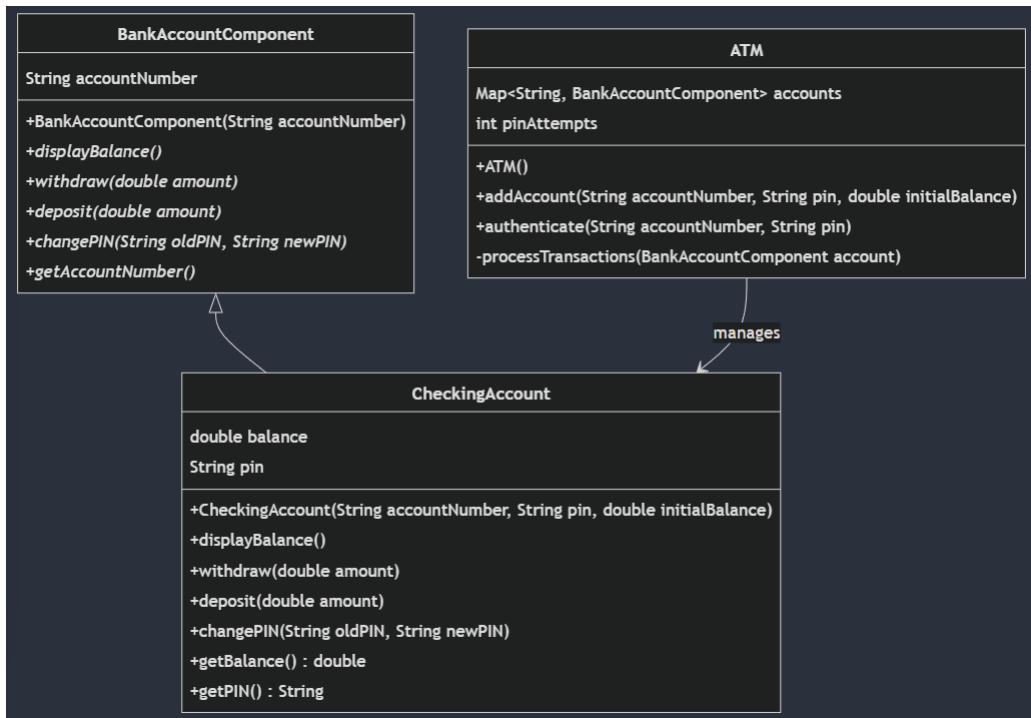
ผลลัพธ์ของ Java // Composite Design Pattern

โดยใช้ Gemini 1.5 Flash

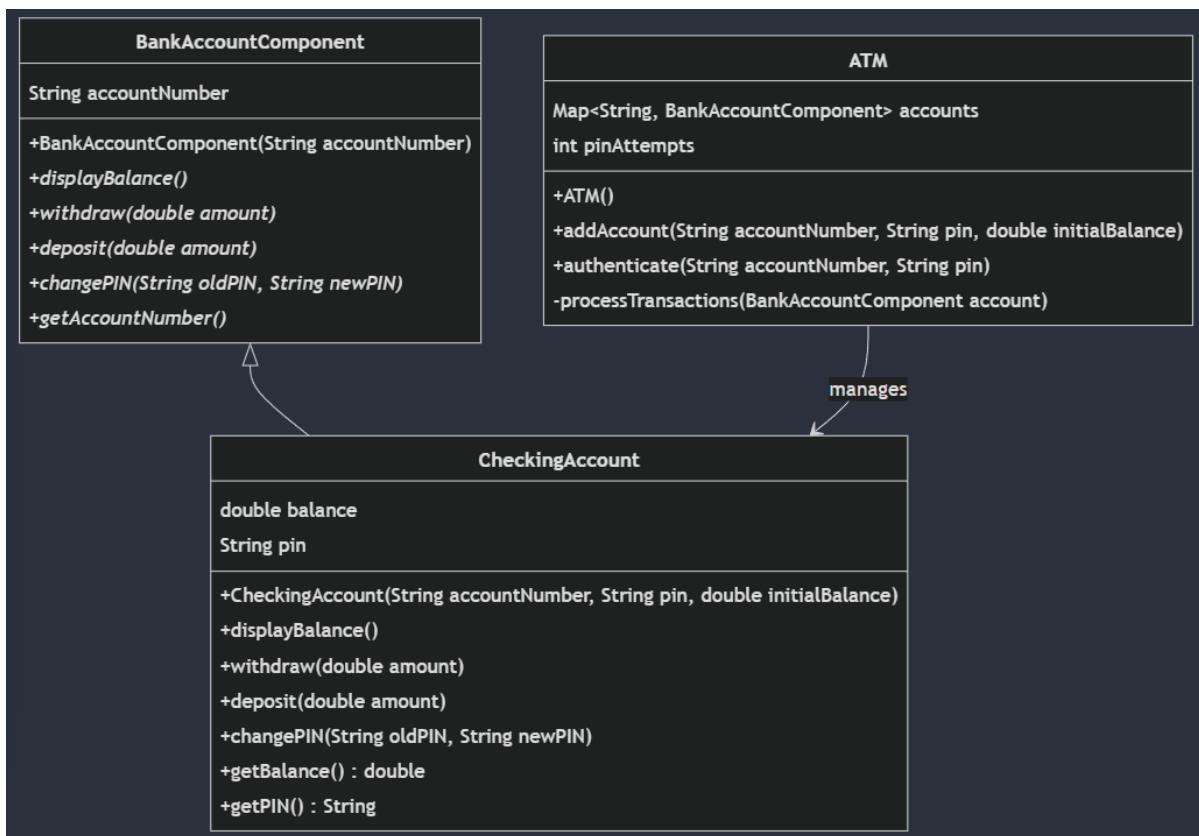
รอบที่ 1



รอบที่ 2

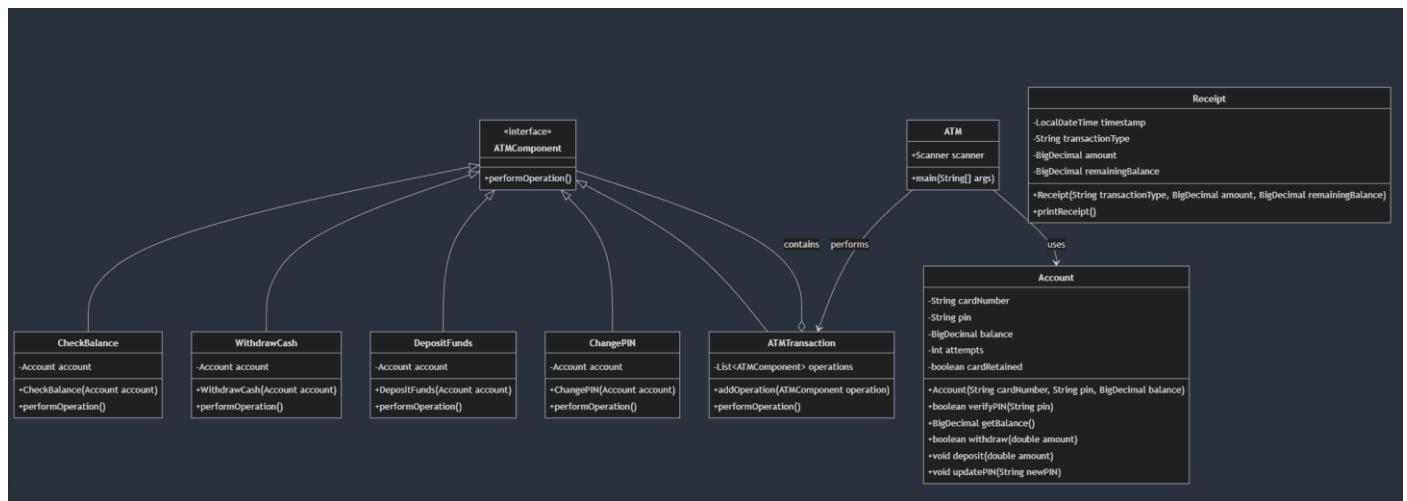


ຮອບທີ 3

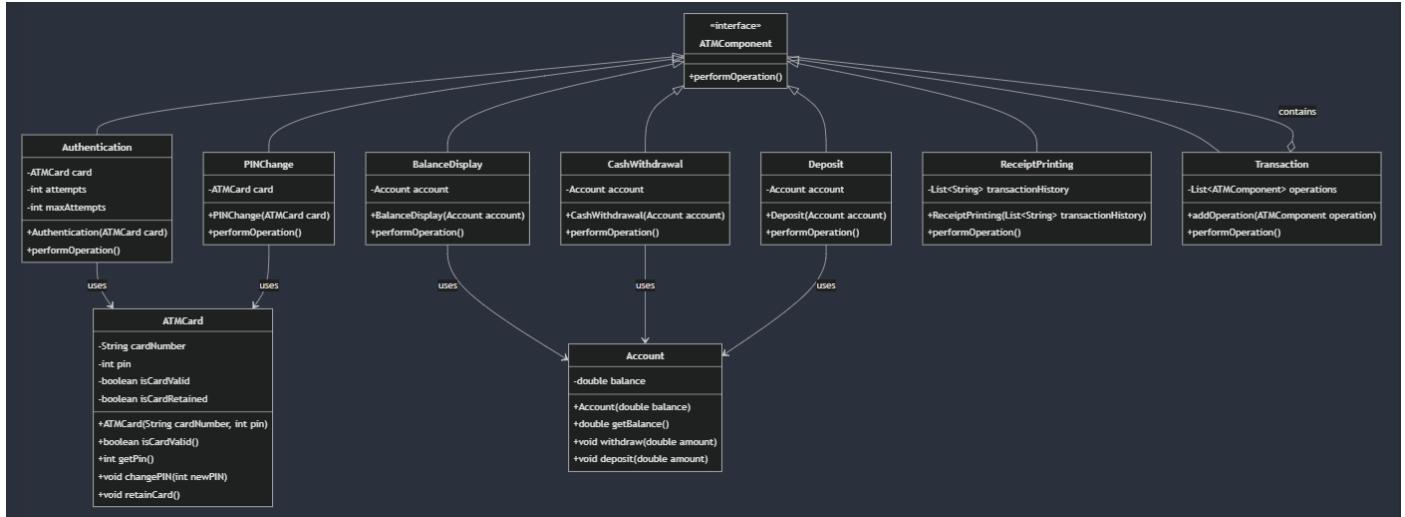


โดยใช้ Gemini 1.5 Pro

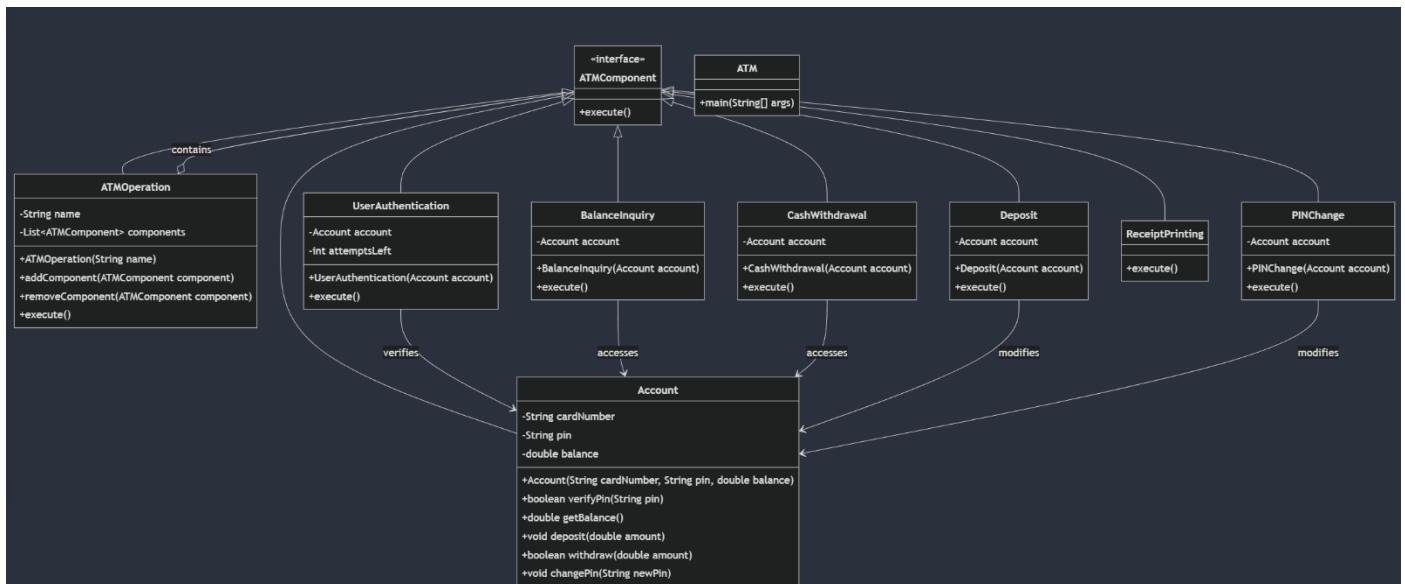
รอบที่ 1



ຮອບທີ່ 2



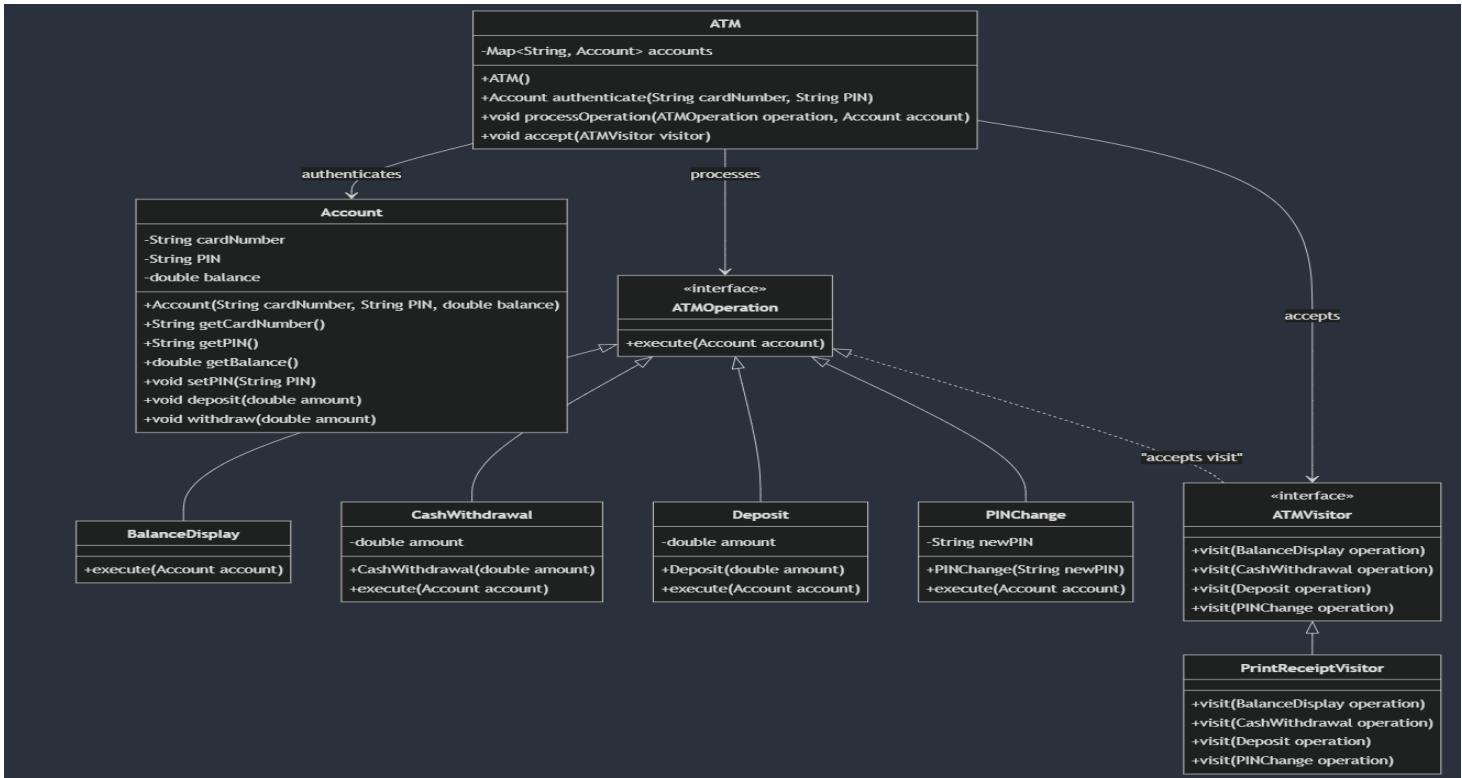
ຮອບທີ່ 3



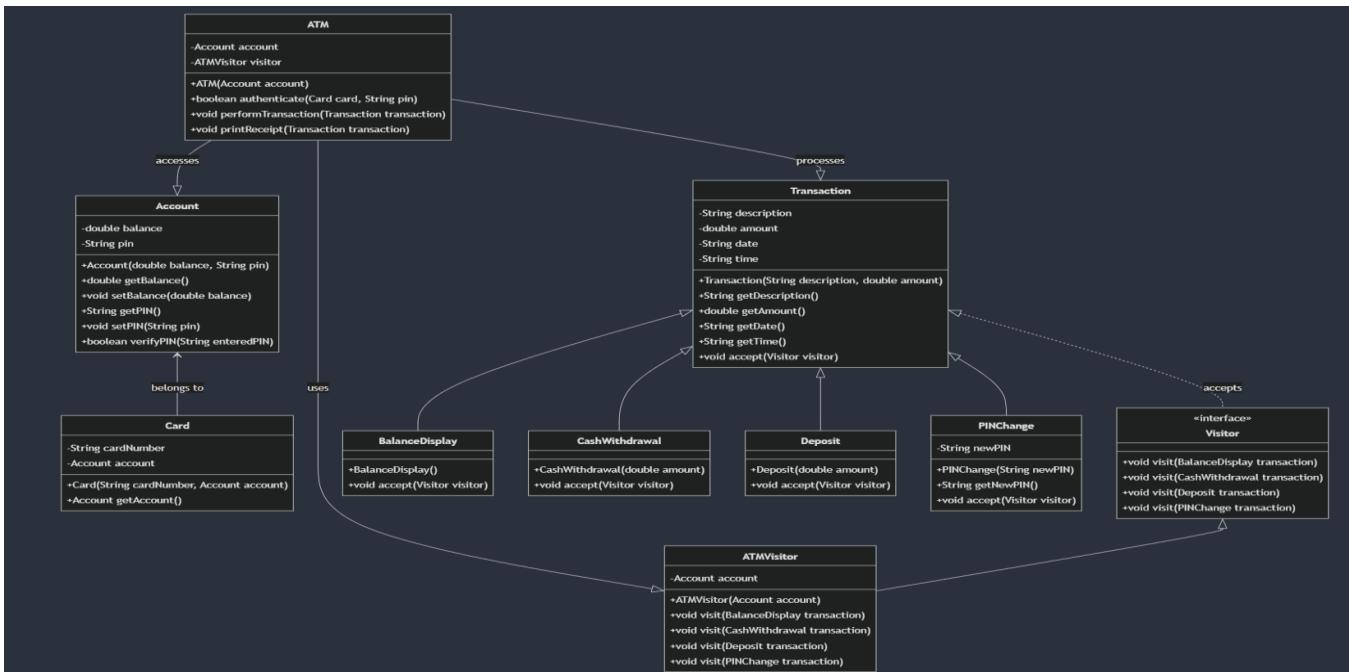
ผลลัพธ์ของ Java // Visitor Design Pattern

โดยใช้ Gemini 1.5 Flash

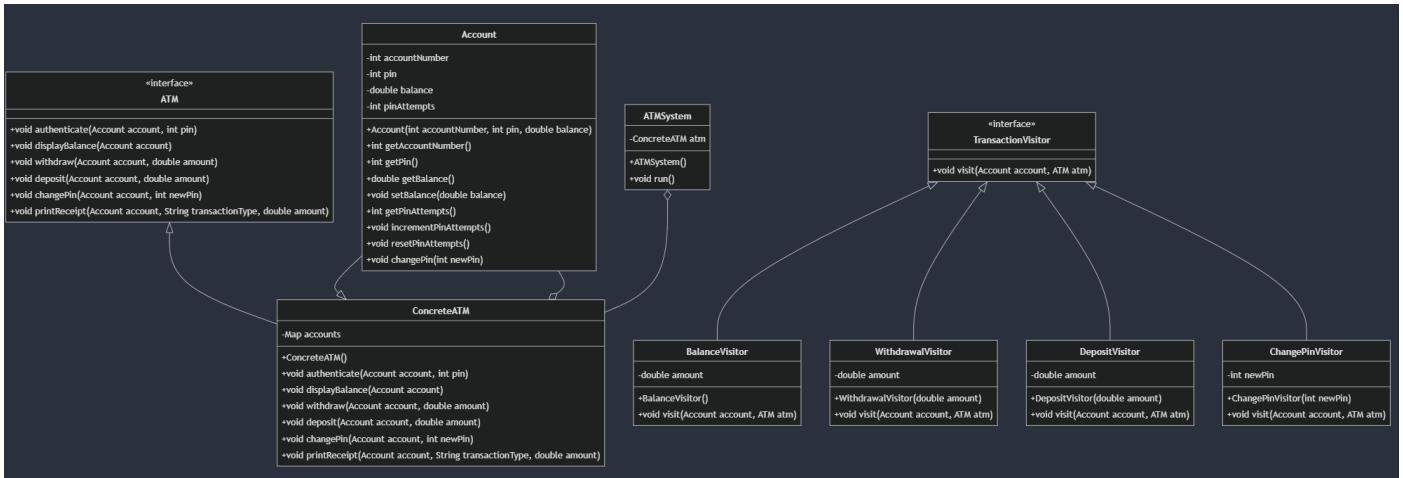
รอบที่ 1



รอบที่ 2

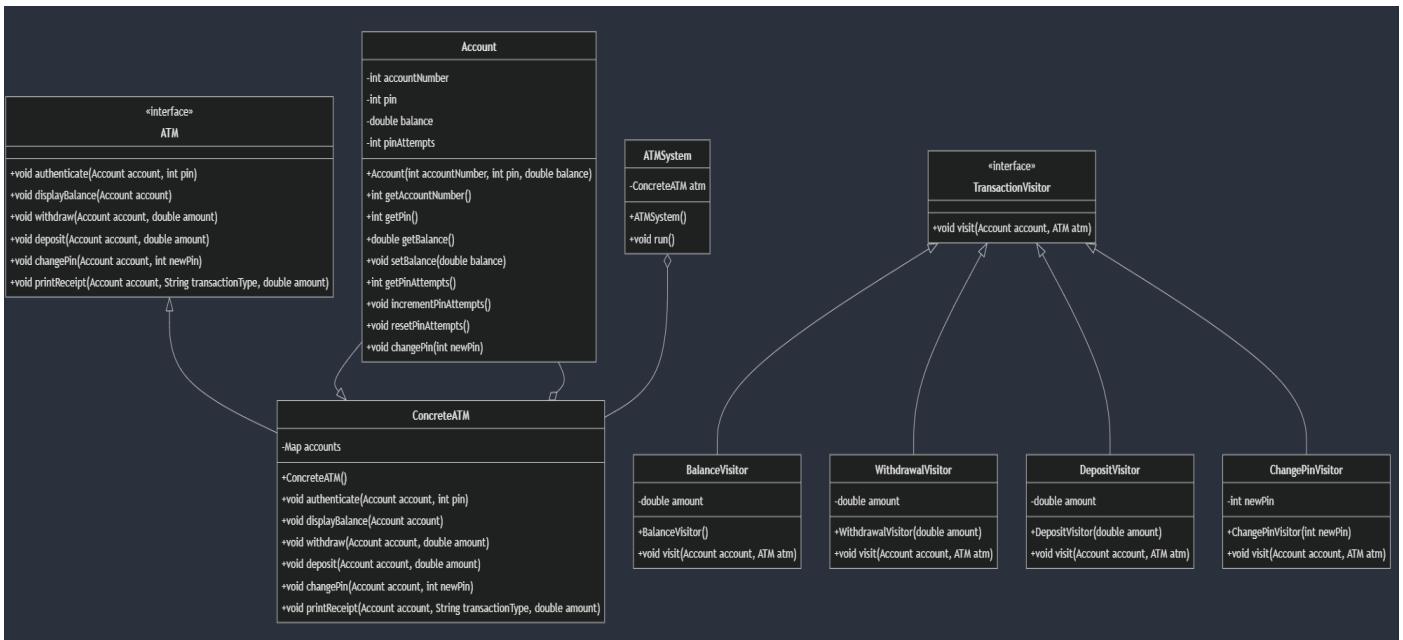


ຮອບທີ 3

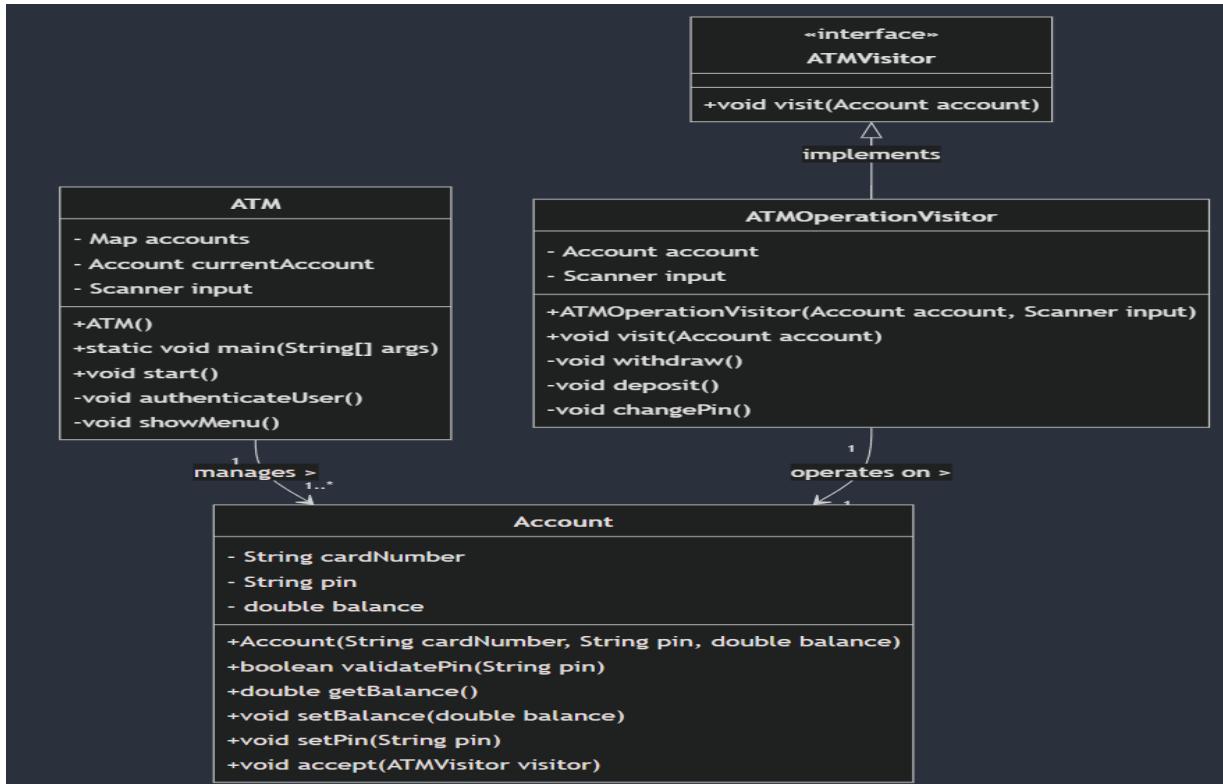


ໄດ້ໃຊ້ Gemini 1.5 Pro

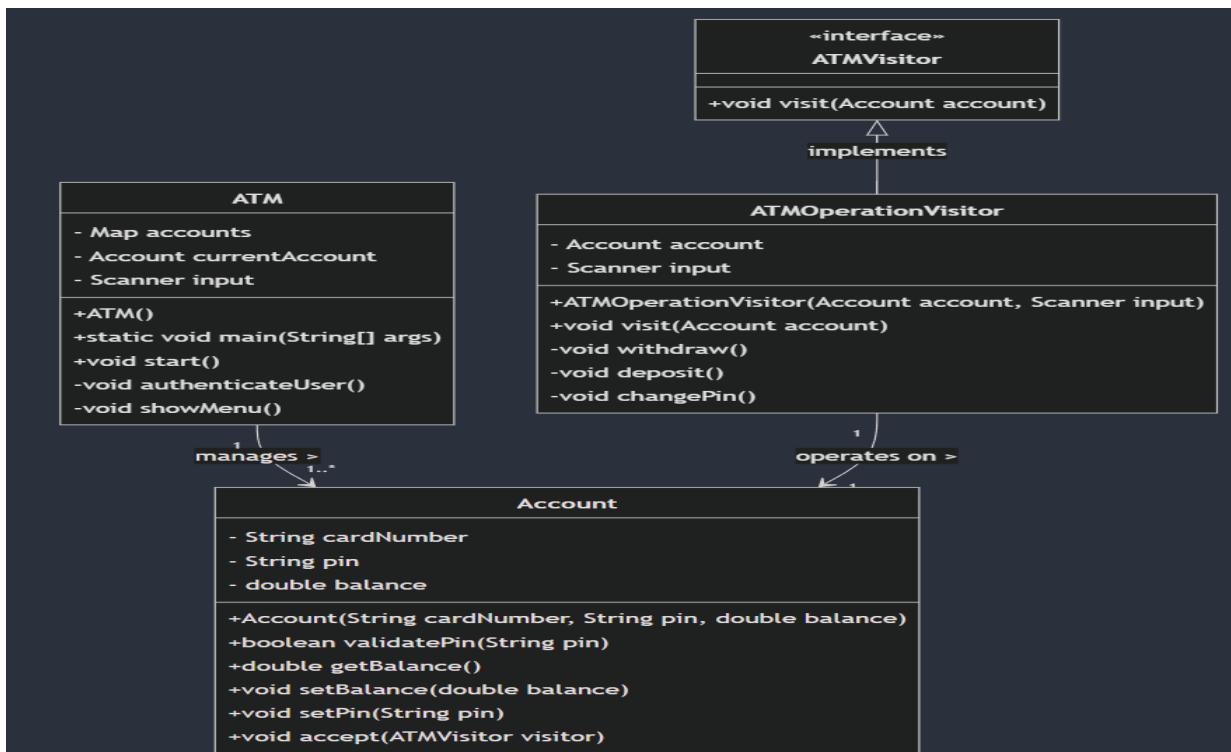
ຮອບທີ 1



รูปที่ 2



รูปที่ 3



วิเคราะห์ผลการทดลอง Gemini 1.5 Flash / Gemini 1.5 Pro

จากผลลัพธ์ พบว่า

ในส่วนของภาษาโปรแกรมมิ่งที่กำหนดให้ในพร้อมพต์ไม่เป็นปัญหา โดยเดลทั้งสองสามารถใช้งาน Java JUnit5 และ Python Pytest ได้อย่างถูกต้อง

ความแตกต่างหลักระหว่าง Gemini 1.5 Flash และ Gemini 1.5 Pro

ในรุ่น Flash เป็น Model ที่เน้นความไวในการสร้างคำ แต่ก็แลกมาด้วยความแม่นยำที่ต่ำลงมาทำให้โค้ดที่ได้อาจไม่ครอบคลุมกับ Requirement ทั้งหมดที่ได้ให้ไป

ในรุ่น Pro เป็น Model ที่เน้นความแม่นยำในการสร้าง แต่จะใช้เวลานานมากกว่าตัว Flash และกินทรัพยากรามากกว่า เช่นกัน แต่ผลลัพธ์ที่ได้ก็สามารถนำไปประยุกต์ใช้ได้มากยิ่งขึ้น

Gemini 1.5 Flash สามารถสร้างโค้ดถูกต้องตาม Requirement แต่รายละเอียดการทำงานของฟังก์ชันนั้นจะไม่ซับซ้อน และ ในบางครั้งจะมี Design Pattern ที่ไม่ตรงกับ Requirement เช่น State Design Pattern และบางกรณีที่ขาด ฟังก์ชันหลักที่ได้ให้ Requirement ไป

Gemini 1.5 Pro สามารถสร้างโครงสร้างพื้นฐานของระบบ ATM ได้ตามหลัก Composite และ Visitor Design Pattern และครอบคลุมฟังก์ชันหลักที่กำหนด แต่ยังขาดรายละเอียดบางส่วน เช่น Logic ของฟังก์ชันบางส่วนที่ให้มาเป็น placeholder ในส่วนโค้ดที่ให้มีจุดผิดพลาดน้อยถึงน้อยมากในการออกแบบตาม Design Pattern ที่กำหนดไว้ คาดเดาว่าปรับปรุงเพียงเล็กน้อยก็สามารถนำไปใช้งานได้

โดยสรุป โดยเดล AI แบบ Generative อย่าง Gemini สามารถเป็นเครื่องมือช่วยในการพัฒนาโปรแกรมได้ในระดับหนึ่ง โดยสามารถช่วยในขั้นตอนการสร้าง ออกแบบ หรือนำร่องการพัฒนาได้ อย่างไรก็ตาม นักพัฒนาควรมีทักษะเฉพาะทางและความเชี่ยวชาญของตนเอง

สิ่งสำคัญคือนักพัฒนาควรตรวจสอบผลลัพธ์ที่ได้จาก AI อย่างละเอียดถี่ถ้วนก่อนนำไปใช้งานจริงหรือพัฒนาต่อ เพื่อให้มั่นใจในความถูกต้องและความสมบูรณ์ของระบบ AI สามารถให้แนวทางและช่วยเหลือได้ แต่การตัดสินใจสุดท้ายและการรับรองคุณภาพยังคงเป็นความรับผิดชอบของนักพัฒนา

การใช้ AI เป็นเครื่องมือช่วยในการพัฒนาอย่างช่วยเหลือ โดยผสมผสานกับความเชี่ยวชาญของนักพัฒนา จะช่วยเพิ่มประสิทธิภาพในกระบวนการพัฒนาซอฟต์แวร์ได้อย่างมีประสิทธิผล

3.GitHub Copilot

Prompt ที่ใช้ในการ Generate Code Python และ Generate Test ทั้งสามรอบ โดยใช้ Composite Design Pattern

Use the composite design pattern to Write code in Python programming language from the requirements below.

"ความต้องการระบบตู้ ATM

1. การยืนยันตัวตนผู้ใช้

การป้อนข้อมูล: ผู้ใช้ต้องสามารถใส่บัตร ATM และรหัส PIN ได้

การตรวจสอบ: ระบบต้องตรวจสอบความถูกต้องของบัตรและรหัส PIN

การจำกัดความพยายาม: หากใส่ผิดเกินจำนวนครั้งที่กำหนด บัตรจะถูกบล็อก

2. การแสดงยอดเงินคงเหลือ

การเข้าถึงข้อมูล: ผู้ใช้สามารถเลือกดูยอดเงินคงเหลือในบัญชีได้

การแสดงผล: ระบบต้องแสดงยอดเงินคงเหลือปัจจุบัน

3. การถอนเงิน

การเลือกจำนวนเงิน: ผู้ใช้สามารถเลือกจำนวนเงินที่ต้องการถอนได้

การตรวจสอบยอดเงิน: ระบบต้องตรวจสอบว่ามียอดเงินเพียงพอสำหรับการถอน

การจ่ายเงิน: หากมียอดเงินเพียงพอ ระบบจะจ่ายเงินให้ผู้ใช้ และปรับปรุงยอดเงินคงเหลือ

การแจ้งเตือน: หากมียอดเงินไม่เพียงพอ ระบบจะแจ้งเตือนผู้ใช้

4. การฝากเงิน

การรับเงินสด: ผู้ใช้สามารถใส่เงินสดเข้าเครื่อง ATM ได้

การนับเงิน: ระบบต้องนับจำนวนเงินที่ฝากเข้ามา

การปรับปรุงยอดเงิน: ระบบต้องปรับปรุงยอดเงินคงเหลือในบัญชี

5. การเปลี่ยนรหัส PIN

การเข้าถึงพังก์ชัน: ผู้ใช้สามารถเปลี่ยนรหัส PIN ของตนเองได้

การยืนยัน: ระบบต้องขอรหัส PIN เก่าและรหัส PIN ใหม่ 2 ครั้งเพื่อยืนยัน

การเปลี่ยนรหัส: หากรหัส PIN ใหม่ตรงกัน ระบบจะเปลี่ยนรหัส PIN ให้

6. การพิมพ์สลิป

การเลือกพิมพ์: หลังจากทำการเสร็จสิ้น ผู้ใช้สามารถเลือกพิมพ์สลิปได้

เนื้อหาสิป: สลิปต้องแสดงข้อมูลรายละเอียดของการซื้อขาย เช่น วันที่ เวลา จำนวนเงิน และยอดเงินคงเหลือ"

Write pytest to test the given code that have 100% statement coverage and 100% Branch coverage.

Prompt ที่ใช้ในการ Generate Code Python และ Generate Test ทั้งสามรอบ โดยใช้ Visitor Design Pattern

Use the visitor design pattern to Write code in Python programming language from the requirements below.

"ความต้องการระบบ ATM

1. การยืนยันตัวตนผู้ใช้

การป้อนข้อมูล: ผู้ใช้ต้องสามารถใส่บัตร ATM และรหัส PIN ได้

การตรวจสอบ: ระบบต้องตรวจสอบความถูกต้องของบัตรและรหัส PIN

การจำกัดความพยายาม: หากใส่ผิดเกินจำนวนครั้งที่กำหนด บัตรจะถูกยกยื่ด

2. การแสดงยอดเงินคงเหลือ

การเข้าถึงข้อมูล: ผู้ใช้สามารถเลือกดูยอดเงินคงเหลือในบัญชีได้

การแสดงผล: ระบบต้องแสดงยอดเงินคงเหลือปัจจุบัน

3. การถอนเงิน

การเลือกจำนวนเงิน: ผู้ใช้สามารถเลือกจำนวนเงินที่ต้องการถอนได้

การตรวจสอบยอดเงิน: ระบบต้องตรวจสอบว่ามียอดเงินเพียงพอสำหรับการถอน

การจ่ายเงิน: หากมียอดเงินเพียงพอ ระบบจะจ่ายเงินให้ผู้ใช้ และปรับปรุงยอดเงินคงเหลือ

การแจ้งเตือน: หากมียอดเงินไม่เพียงพอ ระบบจะแจ้งเตือนผู้ใช้

4. การฝากเงิน

การรับเงินสด: ผู้ใช้สามารถใส่เงินสดเข้าเครื่อง ATM ได้

การนับเงิน: ระบบต้องนับจำนวนเงินที่ฝากเข้ามา

การปรับปรุงยอดเงิน: ระบบต้องปรับปรุงยอดเงินคงเหลือในบัญชี

5. การเปลี่ยนรหัส PIN

การเข้าถึงฟังก์ชัน: ผู้ใช้สามารถเปลี่ยนรหัส PIN ของตนเองได้

การยืนยัน: ระบบต้องขอรหัส PIN เก่าและรหัส PIN ใหม่ 2 ครั้งเพื่อยืนยัน

การเปลี่ยนรหัส: หากรหัส PIN ใหม่ตรงกัน ระบบจะเปลี่ยนรหัส PIN ให้

6. การพิมพ์สลิป

การเลือกพิมพ์: หลังจากทำการเสร็จสิ้น ผู้ใช้สามารถเลือกพิมพ์สลิปได้

เนื้อหาสิป: สลิปต้องแสดงข้อมูลรายละเอียดของรายการ เช่น วันที่ เวลา จำนวนเงิน และยอดเงินคงเหลือ"

Write pytest to test the given code that have 100% statement coverage and 100% Branch coverage.

Prompt ที่ใช้ในการ Generate Code Java และ Generate Test ทั้งสามรอบ โดยใช้ Composite Design Pattern

Use the composite design pattern to Write code in Java programming language from the requirements below.

"ความต้องการระบบตู้ ATM

1. การยืนยันตัวตนผู้ใช้

การป้อนข้อมูล: ผู้ใช้ต้องสามารถใส่บัตร ATM และรหัส PIN ได้

การตรวจสอบ: ระบบต้องตรวจสอบความถูกต้องของบัตรและรหัส PIN

การจำกัดความพยายาม: หากใส่ผิดเกินจำนวนครั้งที่กำหนด บัตรจะถูกยกยื่ด

2. การแสดงยอดเงินคงเหลือ

การเข้าถึงข้อมูล: ผู้ใช้สามารถเลือกดูยอดเงินคงเหลือในบัญชีได้

การแสดงผล: ระบบต้องแสดงยอดเงินคงเหลือปัจจุบัน

3. การถอนเงิน

การเลือกจำนวนเงิน: ผู้ใช้สามารถเลือกจำนวนเงินที่ต้องการถอนได้

การตรวจสอบยอดเงิน: ระบบต้องตรวจสอบว่ามียอดเงินเพียงพอสำหรับการถอน

การจ่ายเงิน: หากมียอดเงินเพียงพอ ระบบจะจ่ายเงินให้ผู้ใช้ และปรับปรุงยอดเงินคงเหลือ

การแจ้งเตือน: หากมียอดเงินไม่เพียงพอ ระบบจะแจ้งเตือนผู้ใช้

4. การฝากเงิน

การรับเงินสด: ผู้ใช้สามารถใส่เงินสดเข้าเครื่อง ATM ได้

การนับเงิน: ระบบต้องนับจำนวนเงินที่ฝากเข้ามา

การปรับปรุงยอดเงิน: ระบบต้องปรับปรุงยอดเงินคงเหลือในบัญชี

5. การเปลี่ยนรหัส PIN

การเข้าถึงฟังก์ชัน: ผู้ใช้สามารถเปลี่ยนรหัส PIN ของตนเองได้

การยืนยัน: ระบบต้องขอรหัส PIN เก่าและรหัส PIN ใหม่ 2 ครั้งเพื่อยืนยัน

การเปลี่ยนรหัส: หากรหัส PIN ใหม่ตรงกัน ระบบจะเปลี่ยนรหัส PIN ให้

6. การพิมพ์สลิป

การเลือกพิมพ์: หลังจากทำการเสร็จสิ้น ผู้ใช้สามารถเลือกพิมพ์สลิปได้

เนื้อหาสิป: สลิปต้องแสดงข้อมูลรายละเอียดของรายการ เช่น วันที่ เวลา จำนวนเงิน และยอดเงินคงเหลือ"

Write to test the given code that have 100% statement coverage and 100% Branch coverage.

Prompt ที่ใช้ในการ Generate Code Java และ Generate Test ทั้งสามรอบ โดยใช้ Visitor Design Pattern

Use the visitor design pattern to Write code in Java programming language from the requirements below.

"ความต้องการระบบ ATM

1. การยืนยันตัวตนผู้ใช้

การป้อนข้อมูล: ผู้ใช้ต้องสามารถใส่บัตร ATM และรหัส PIN ได้

การตรวจสอบ: ระบบต้องตรวจสอบความถูกต้องของบัตรและรหัส PIN

การจำกัดความพยายาม: หากใส่ผิดเกินจำนวนครั้งที่กำหนด บัตรจะถูกยกยื่ด

2. การแสดงยอดเงินคงเหลือ

การเข้าถึงข้อมูล: ผู้ใช้สามารถเลือกดูยอดเงินคงเหลือในบัญชีได้

การแสดงผล: ระบบต้องแสดงยอดเงินคงเหลือปัจจุบัน

3. การถอนเงิน

การเลือกจำนวนเงิน: ผู้ใช้สามารถเลือกจำนวนเงินที่ต้องการถอนได้

การตรวจสอบยอดเงิน: ระบบต้องตรวจสอบว่ามียอดเงินเพียงพอสำหรับการถอน

การจ่ายเงิน: หากมียอดเงินเพียงพอ ระบบจะจ่ายเงินให้ผู้ใช้ และปรับปรุงยอดเงินคงเหลือ

การแจ้งเตือน: หากมียอดเงินไม่เพียงพอ ระบบจะแจ้งเตือนผู้ใช้

4. การฝากเงิน

การรับเงินสด: ผู้ใช้สามารถใส่เงินสดเข้าเครื่อง ATM ได้

การนับเงิน: ระบบต้องนับจำนวนเงินที่ฝากเข้ามา

การปรับปรุงยอดเงิน: ระบบต้องปรับปรุงยอดเงินคงเหลือในบัญชี

5. การเปลี่ยนรหัส PIN

การเข้าถึงฟังก์ชัน: ผู้ใช้สามารถเปลี่ยนรหัส PIN ของตนเองได้

การยืนยัน: ระบบต้องขอรหัส PIN เก่าและรหัส PIN ใหม่ 2 ครั้งเพื่อยืนยัน

การเปลี่ยนรหัส: หากรหัส PIN ใหม่ตรงกัน ระบบจะเปลี่ยนรหัส PIN ให้

6. การพิมพ์สลิป

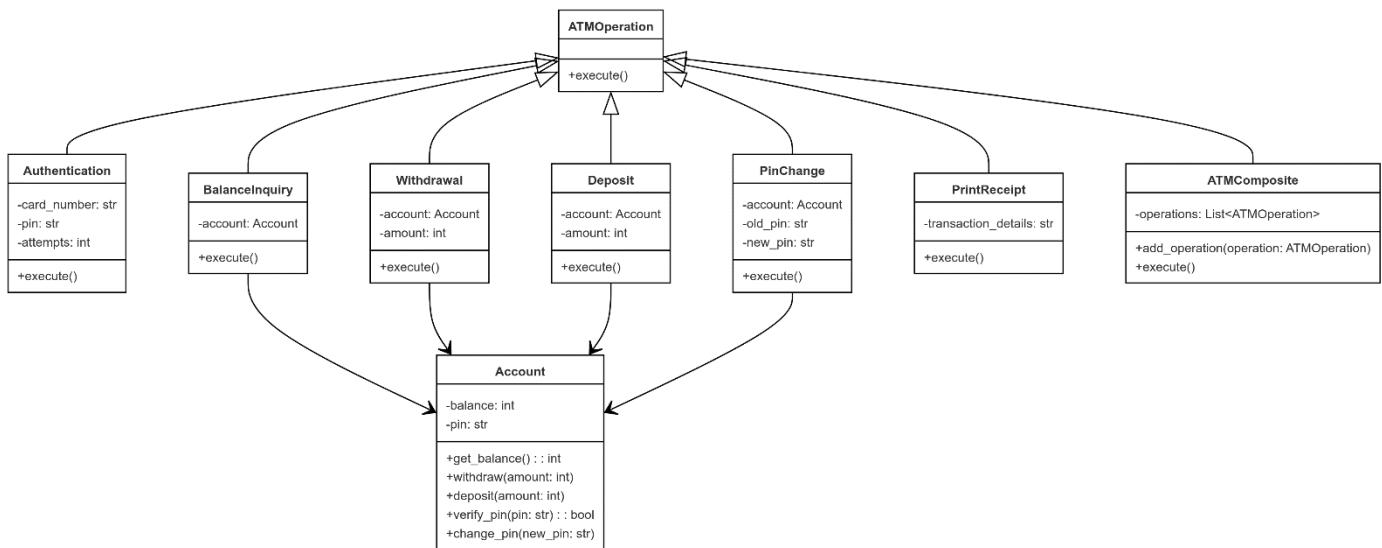
การเลือกพิมพ์: หลังจากทำการเสร็จสิ้น ผู้ใช้สามารถเลือกพิมพ์สลิปได้

เนื้อหาสำคัญ: สลิปต้องแสดงข้อมูลรายละเอียดของรายการ เช่น วันที่ เวลา จำนวนเงิน และยอดเงินคงเหลือ"

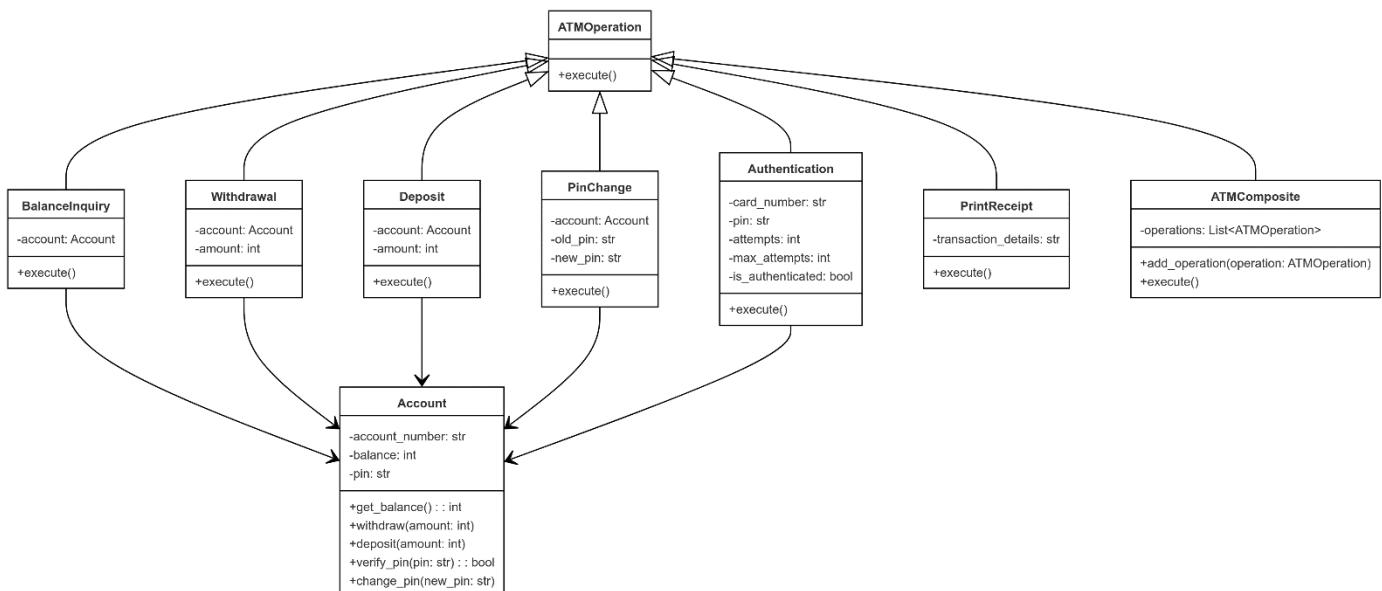
Write to test the given code that have 100% statement coverage and 100% Branch coverage.

ผลลัพธ์ของ Python // Composite Design Pattern

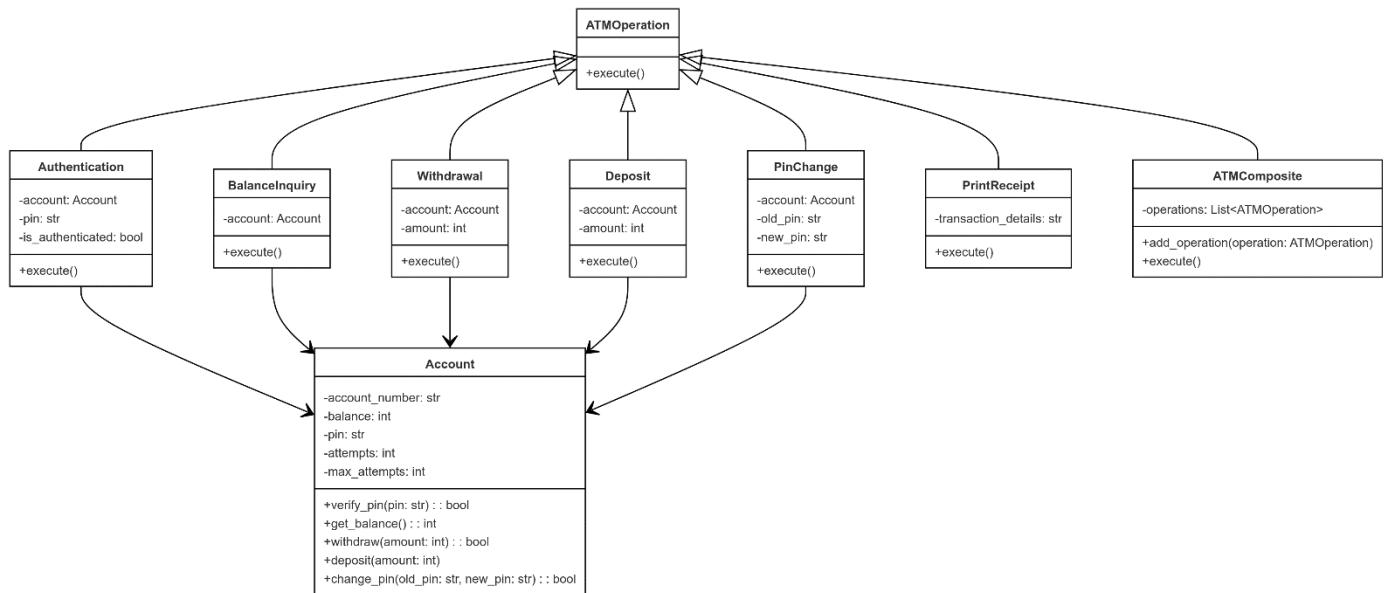
รอบที่ 1



รอบที่ 2

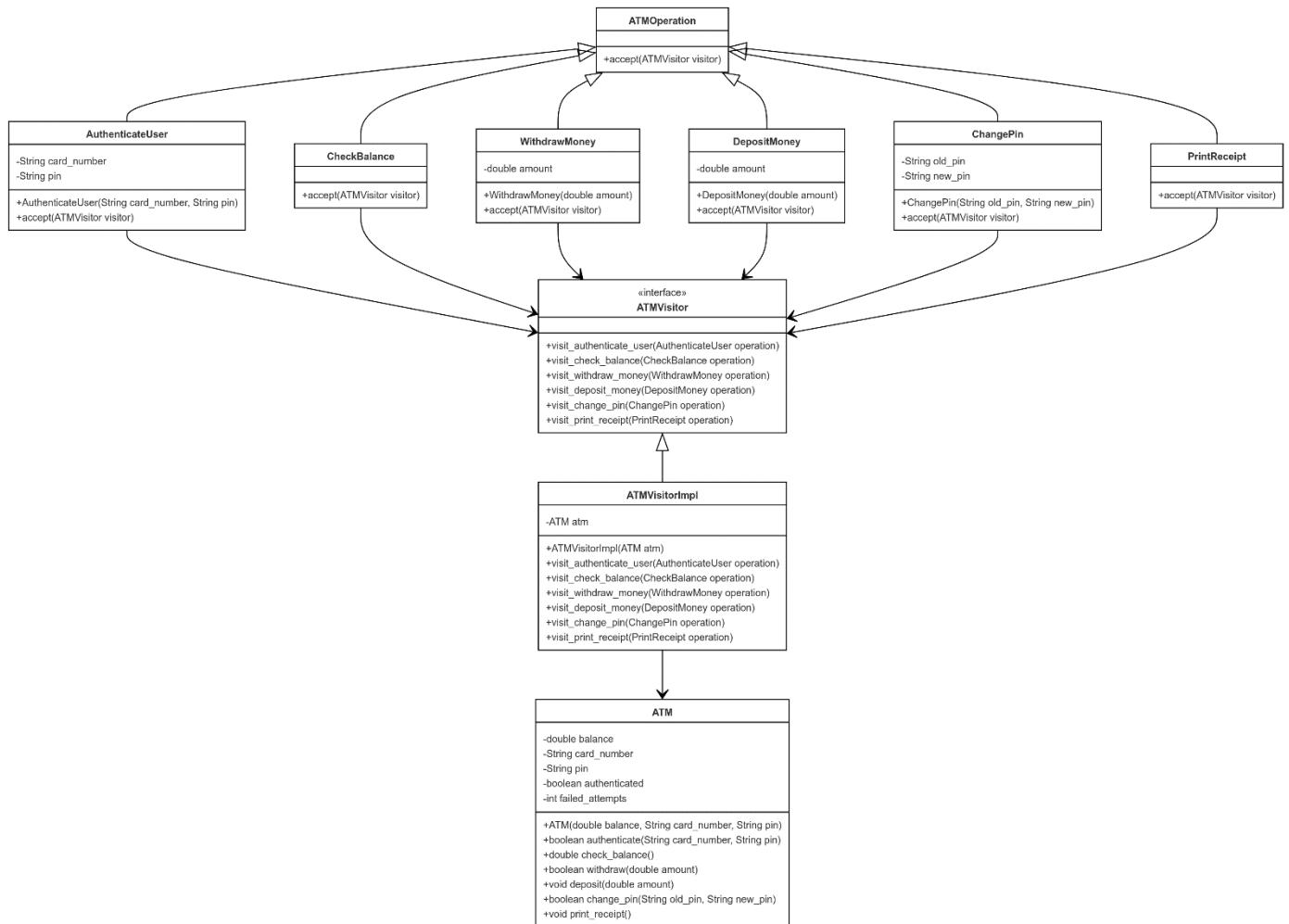


รอบที่ 3

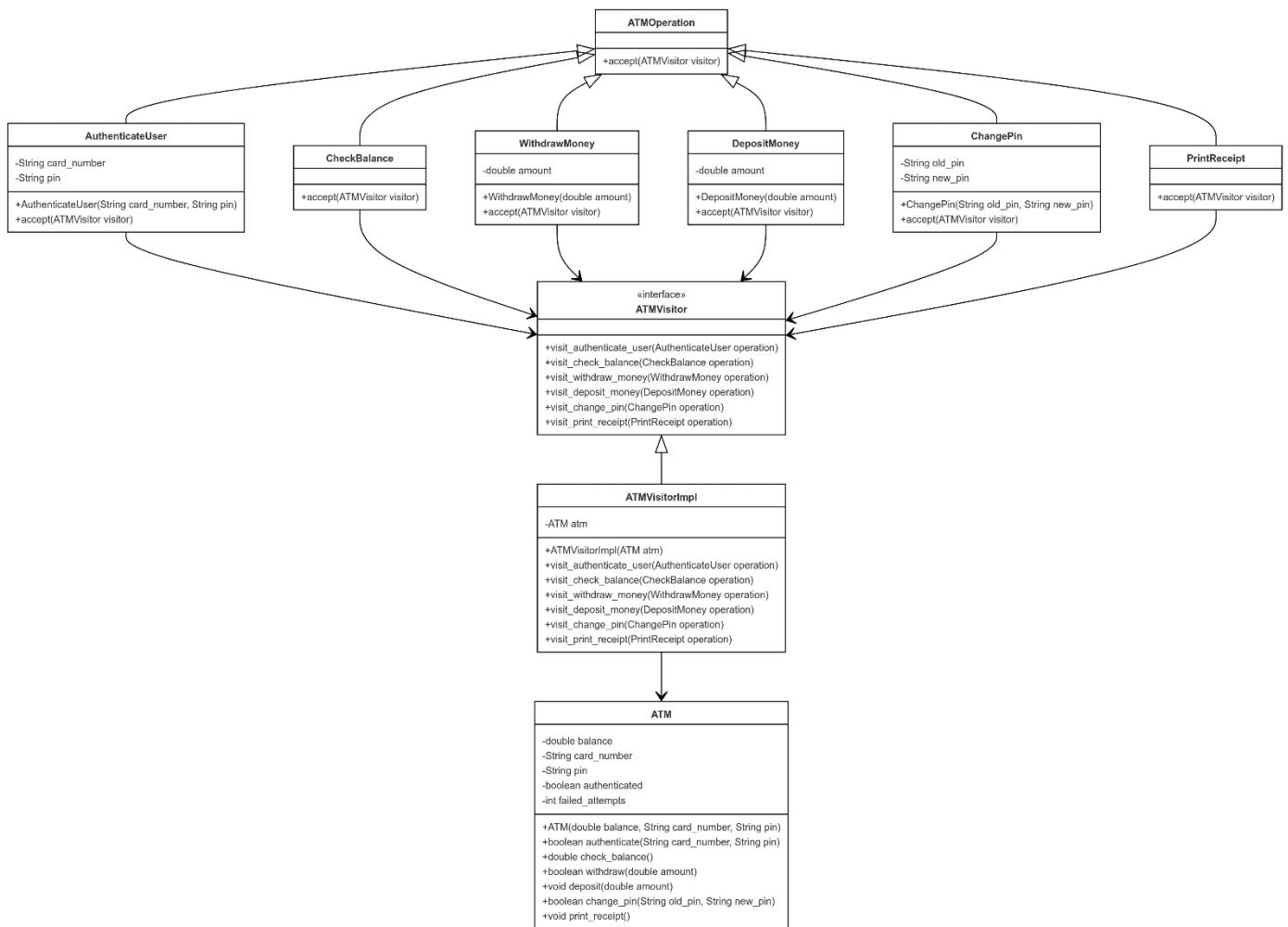


ผลลัพธ์ของ Python // Visitor Design Pattern

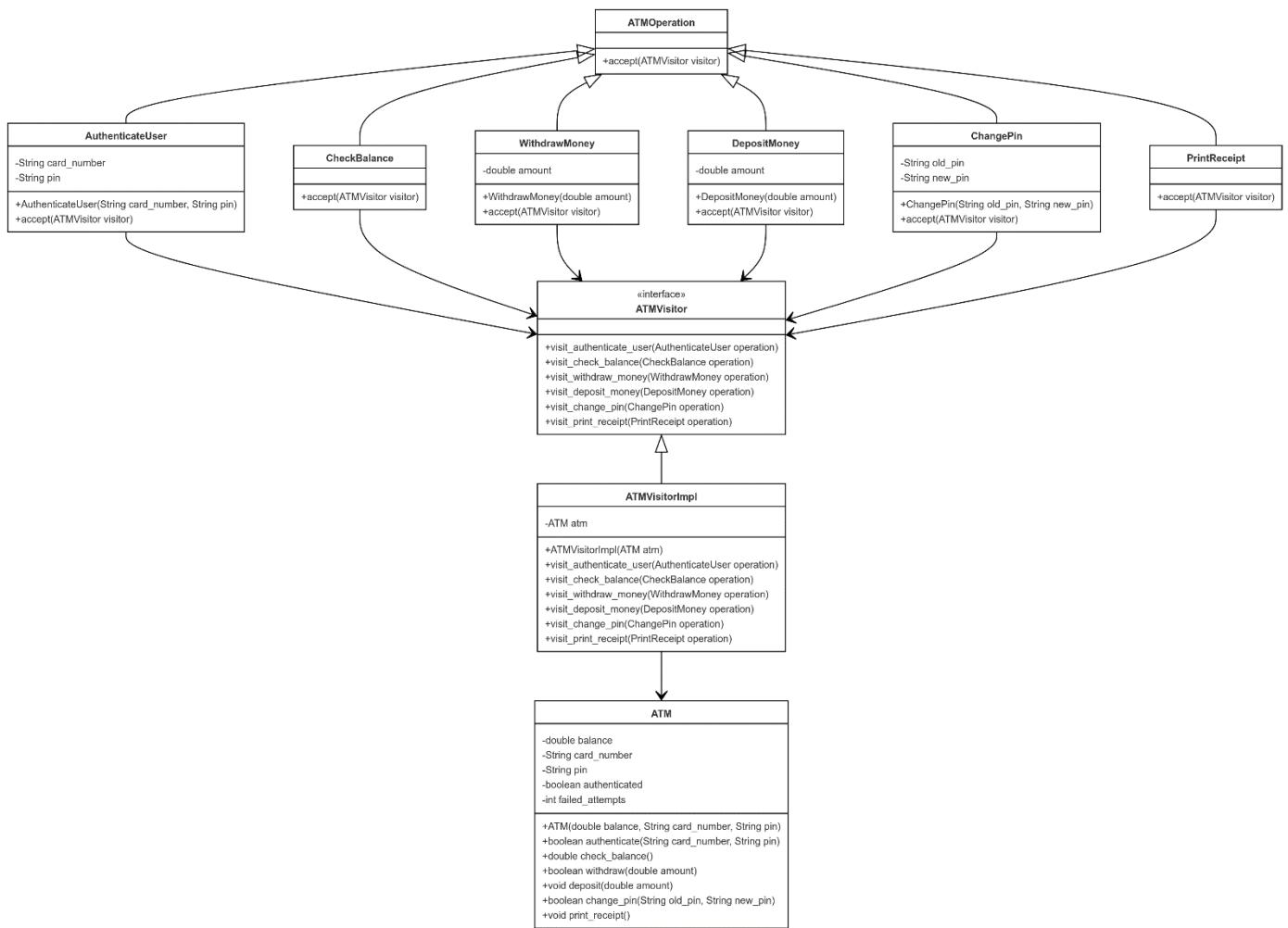
รอบที่ 1



รอบที่ 2

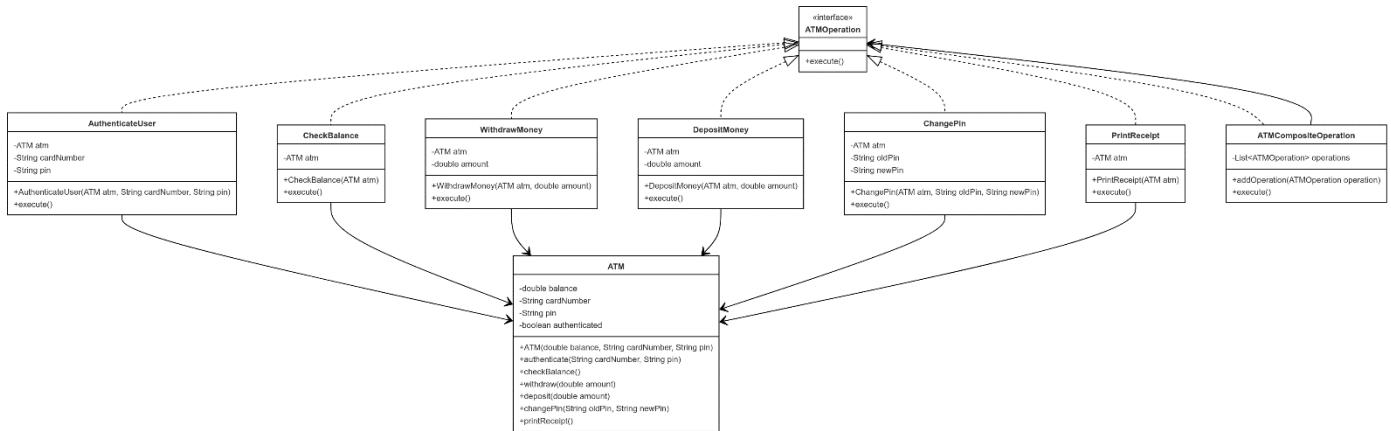


รอบที่ 3

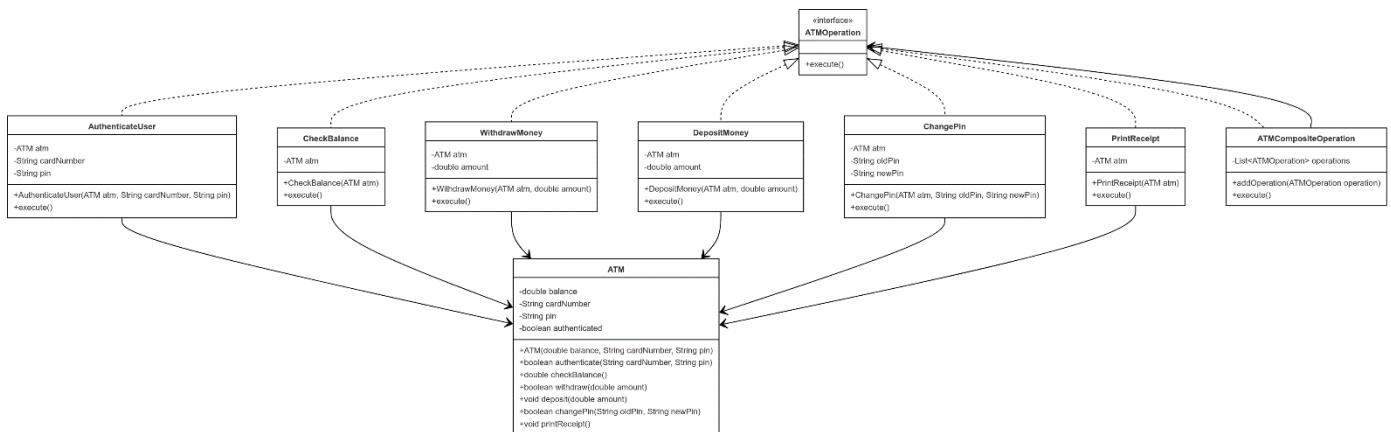


ผลลัพธ์ของ Java // Composite Design Pattern

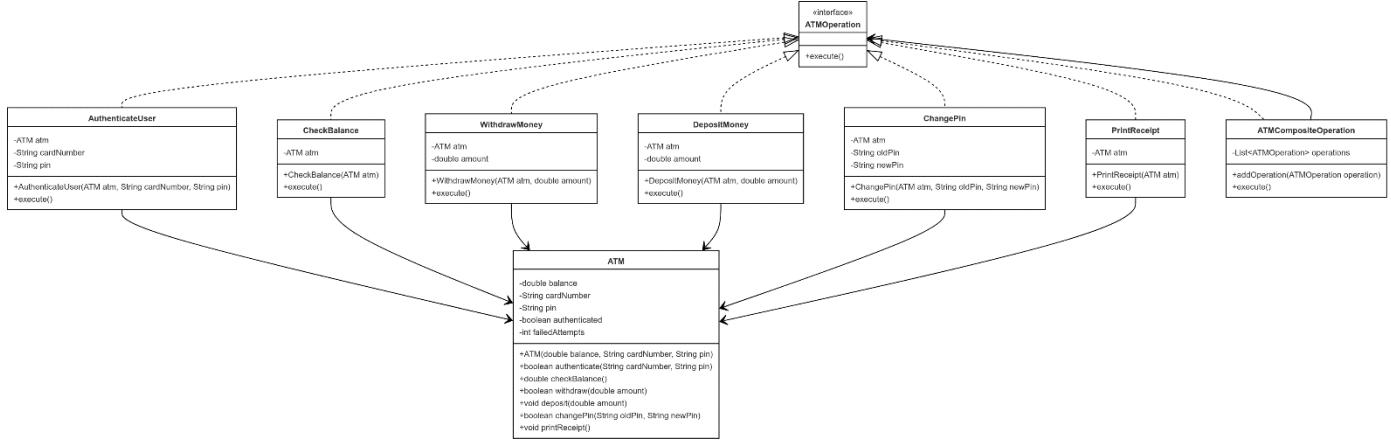
รอบที่1



รอบที่2



รอบที่ 3

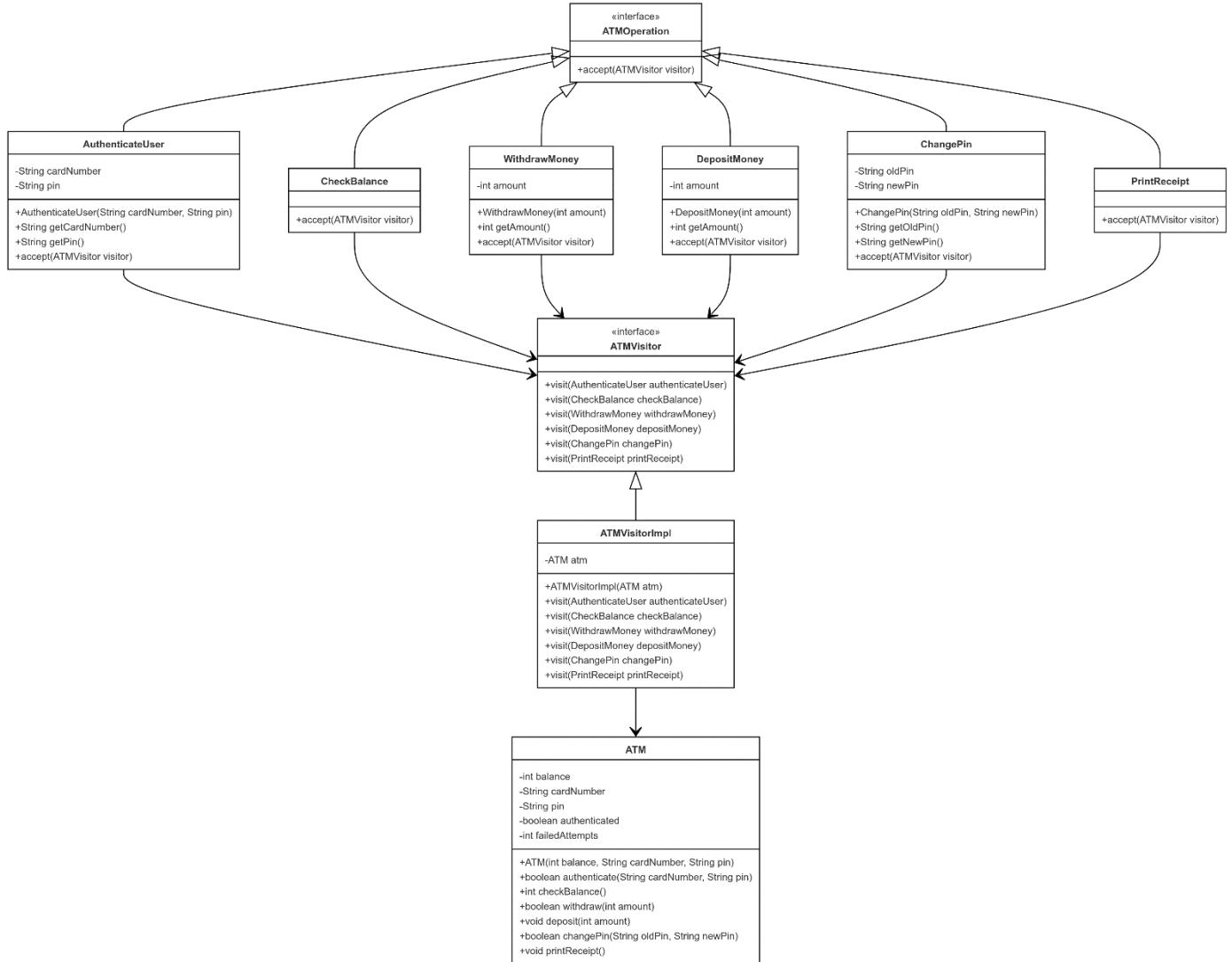


ผลลัพธ์ของ Java // Visitor Design Pattern

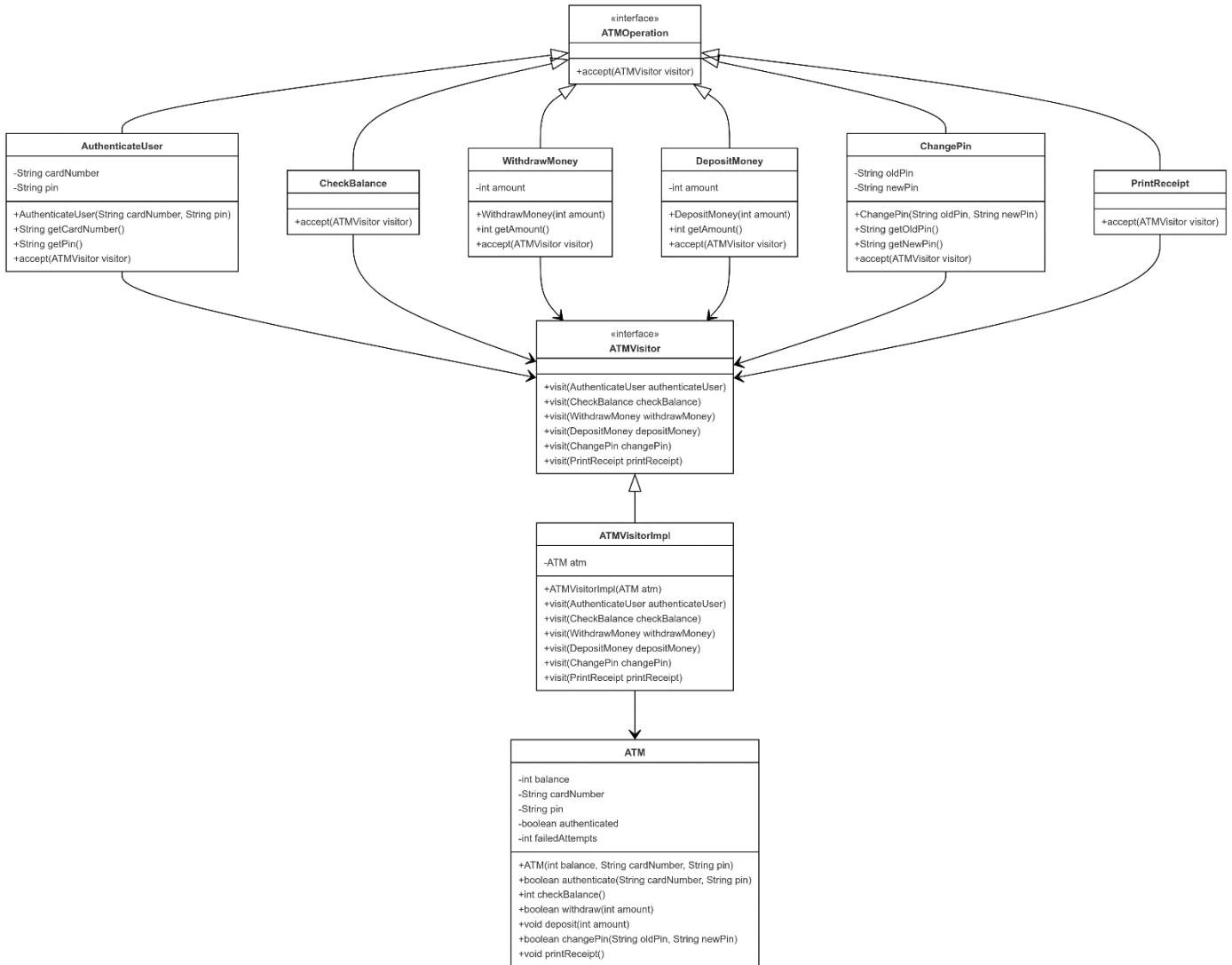
รอบที่ 1



รอบที่ 2



รอบที่ 3



วิเคราะห์ผลการทดลอง copilot

จากการลองใช้ copilot ในด้านภาษา programming นั้น copilot เขียนออกมาระบุอย่างถูกต้องและแม่นยำทั้ง python และ java แต่ทุกครั้งที่ใส่ prompt code ที่ออกแบบแต่ละครั้งจะมีการเปลี่ยนแปลงนิดหน่อย เป็นการปรับและเปลี่ยนแปลงให้เข้าใจ prompt ก่อนๆ ส่วนในด้าน Testing นั้น code ที่ออกแบบจะไม่ค่อยมีความแตกต่างจาก code ก่อนหน้าแต่ทำการ test ได้ปกติถึงจะมี test fail ในบางครั้งก็ตาม

โดยสรุป โมเดล AI copilot ถือว่าเป็นเครื่องมือที่มีประสิทธิภาพเป็นอย่างมากในการช่วยเขียนโคด และยังสามารถให้คำแนะนำนำมือใช้และแก้ข้อผิดพลาดเพื่อเพิ่มประสิทธิภาพในการเขียนโคดได้ออกด้วย แต่ผู้ใช้ควรใช้ copilot เป็นผู้ช่วยในการเขียนแทนที่จะเป็นตัวหลัก เพราะ copilot คิดได้แม่นยำ 100% และการตัดสินใจหลักควรอยู่ในมือของผู้พัฒนา

2. นำซอสโค้ดที่ถูกสร้างขึ้นไปทดลองรันและบันทึกผล ได้ผลลัพธ์ดังนี้

Model AI	Programming language	Design pattern	Round	Executable	Description
Chat-GPT 4o	Python	Composite	1	yes	ผลลัพธ์ที่ได้ไม่ถูกต้อง เนื่องจากมีการหารค่าเงินในบัญชี ให้แต่ละฟังก์ชัน เช่นการฝากเงิน ถอนเงิน ดังนั้นหลังจากที่ทำงานเสร็จ ค่าเงินจะกลับไปที่ค่าเริ่มต้นเสมอ
			2	yes	ผลลัพธ์ที่ได้สามารถทำงานด้านการคำนวณได้ถูกต้อง แต่ไม่ถูกหลักการเนื่องจากก่อนทำการ ฝาก ถอน และปริ้นใบเสร็จ ผู้ใช้ต้องเป็นคนใส่ค่าเงินฝากตัวเอง ไปด้วย
			3	yes	ผลลัพธ์ที่ได้สามารถทำงานด้านการคำนวณได้ถูกต้อง แต่ไม่ถูกหลักการเนื่องจากก่อนทำการ ฝาก ถอน และปริ้นใบเสร็จ ผู้ใช้ต้องเป็นคนใส่ค่าเงินฝากตัวเอง ไปด้วย
Chat-GPT 4o	Python	Visitor	1	yes	ผลลัพธ์ที่ได้ถูกต้อง เนื่องจากมีการหารค่าเงินในบัญชี ให้แต่ละฟังก์ชัน เช่นการฝากเงิน ถอนเงิน ดังนั้นหลังจากที่ฟังก์ชันทำงานเสร็จ ค่าเงินในบัญชี จะเปลี่ยนแปลงตามไปด้วย

			2	yes	ผลลัพธ์ที่ได้สามารถทำงานด้านการคำนวณได้ถูกต้อง แต่ไม่ถูกหลักการเนื่องจากก่อนทำการคำนวณ ผู้ใช้ต้องเป็นคนใส่ค่าเงินฝากตัวเองไปด้วย
			3	yes	ผลลัพธ์ที่ได้ไม่ถูกต้อง เนื่องจากไม่มีการแบ่งค่าเงินในบัญชี ให้แต่ละฟังก์ชัน เข่นการฝากเงิน ถอนเงิน ดังนั้นหลังจากที่ทำงานเสร็จ ค่าเงินจะกลับไปที่ค่าเริ่มต้นเสมอ
Chat-GPT 4o	Java	Composite	1	yes	ผลลัพธ์ที่ได้ไม่ถูกต้อง เนื่องจากระบบไม่สามารถคำนวณ การรับฝาก ถอนเงินได้มีเพียงแค่การปรินต์ชื่อการทำงานของฟังก์ชัน
			2	yes	ผลลัพธ์ที่ได้ไม่ถูกต้อง เนื่องจากระบบมีการนำการทำงานของฟังก์ชันก่อนหน้าที่เราเคยเรียกใช้ มาเรียกใช้ซ้ำ ทำให้จำนวนเงินในบัญชีไม่ถูกต้อง
			3	yes	ผลลัพธ์ที่ได้สามารถทำงานด้านการคำนวณ ถอนเงิน ฝากเงิน ได้ถูกต้อง
Chat-GPT 4o	Java	Visitor	1	yes	ผลลัพธ์ที่ได้ มีฟังก์ชันที่ไม่มีการคำนวณ มีเพียงแค่ปรับค่าเงินที่รับเข้าไป
			2	yes	ผลลัพธ์ที่ได้สามารถทำงานด้านการคำนวณได้ถูกต้อง

			3	yes	ผลลัพธ์ที่ได้สามารถทำการคำนวณและการเปลี่ยนรหัสผ่านได้อย่างถูกต้อง
Gemini 1.5 Flash	Python	Composite	1	yes	ฟังก์ชันหลักๆทำงานได้ ยกเว้นการยืนยันตัวตน และการพิมพ์ใบเสร็จเนื่องจากตัว AI ไม่ได้สร้างส่วนการทำงานของฟังก์ชัน
			2	yes	ฟังก์ชันหลักๆทำงานได้ ยกเว้นการยืนยันตัวตน และการพิมพ์ใบเสร็จเนื่องจากตัว AI ไม่ได้สร้างส่วนการทำงานของฟังก์ชัน
			3	yes	ฟังก์ชันหลักๆทำงานได้ ยกเว้นการยืนยันตัวตน เนื่องจากตัว AI มีการสร้างส่วนการทำงานของฟังก์ชัน แต่ยังมีการใช้ตรรกะการทำงานที่ผิด
Gemini 1.5 Flash	Python	Visitor	1	yes	ทำงานได้แค่การยืนยันตัวตน การแสดงยอดเงินและการถอนเงิน ส่วนฟังก์ชันอื่นๆ ไม่มีการใส่การทำงานไว้ในตัวโค้ด
			2	yes	สามารถรับได้แต่ ไม่สามารถทำงานได้เนื่องจากไม่มีการใส่การทำงานในส่วนการเลือกธุกรรมที่ต้องการ
			3	yes	ทำงานได้ทุกฟังก์ชันตามความต้องการ
Gemini 1.5 Flash	Java	Composite	1	yes	ฟังก์ชันหลักๆทำงานได้ ยกเว้นการยืนยันตัวตน และการพิมพ์ใบเสร็จเนื่องจากตัว AI ไม่ได้สร้างส่วนการทำงานของฟังก์ชัน
			2	yes	ทำงานได้ทุกฟังก์ชันตามความต้องการ แต่ขาดฟังก์ชันพิมพ์ใบเสร็จ
			3	yes	สามารถทำงานทุกฟังก์ชันได้แต่ไม่ได้มีการสร้างระบบเลือกธุกรรมตาม choice

Gemini 1.5 Flash	Java	Visitor	1	yes	สามารถทำงานได้บางฟังก์ชันแต่ไม่ได้มีการสร้างระบบเลือกธุกรรมตาม choice
			2	yes	สามารถทำงานได้บางฟังก์ชันได้แต่ไม่ได้มีการสร้างระบบเลือกธุกรรมตาม choice
			3	yes	สามารถทำงานได้บางฟังก์ชันได้แต่ไม่ได้มีการสร้างระบบเลือกธุกรรมตาม choice
Gemini 1.5 Pro	Python	Composite	1	yes	ทำงานได้บางส่วนแต่ยังไม่ถูกต้องทั้งหมด เช่นการทำงานของส่วนฝากถอน
			2	yes	ฟังก์ชันหลักๆ ทำงานได้ ยกเว้นการยืนยันตัวตน และการพิมพ์ใบเสร็จเนื่องจากตัว AI มีการสร้างส่วนการทำงานของฟังก์ชัน แต่ยังไม่สามารถใช้ตรรกะการทำงานที่ผิด
			3	yes	สามารถรันได้แต่ ไม่สามารถทำงานได้เนื่องจากไม่มีการใส่การทำงานในแต่ละส่วน
Gemini 1.5 Pro	Python	Visitor	1	yes	ทำงานได้ทุกฟังก์ชันตามความต้องการ แต่ขาดฟังก์ชันพิมพ์ใบเสร็จ
			2	yes	ทำงานได้ทุกฟังก์ชันตามความต้องการ แต่ฟังก์ชันพิมพ์ใบเสร็จจะไม่มีให้เลือก แต่จะพิมพ์อัตโนมัติทุกรายการ
			3	yes	ทำงานได้ทุกฟังก์ชันตามความต้องการ

Gemini 1.5 Pro	Java	Composite	1	yes	สามารถทำงานทุกฟังก์ชันได้แต่ไม่ได้มีการสร้างระบบเลือกธุกรรมตาม choice
			2	yes	สามารถทำงานทุกฟังก์ชันได้แต่ไม่ได้มีการสร้างระบบเลือกธุกรรมตาม choice
			3	yes	สามารถทำงานทุกฟังก์ชันได้แต่ไม่ได้มีการสร้างระบบเลือกธุกรรมตาม choice
Gemini 1.5 Pro	Java	Visitor	1	yes	ทำงานได้ทุกฟังก์ชันตามความต้องการ
			2	yes	ทำงานได้ทุกฟังก์ชันตามความต้องการ แต่ขาดฟังก์ชันพิมพ์ใบเสร็จ
			3	yes	ทำงานได้ทุกฟังก์ชันตามความต้องการ แต่ขาดฟังก์ชันพิมพ์ใบเสร็จ
GitHub Copilot	Python	Composite	1	yes	สามารถคำนวณจำนวนเงินในบัญชีได้อย่างถูกต้อง ตามฟังก์ชันกิจกรรมการทำงาน แต่การยืนยันตัวตนการเข้าถึงบัญชียังไม่ถูกต้อง เนื่องจาก ไม่มีการเช็ครหัสผ่าน
			2	yes	สามารถคำนวณจำนวนเงินในบัญชีได้อย่างถูกต้อง ตามฟังก์ชันกิจกรรมการทำงาน แต่การยืนยันตัวตนการเข้าถึงบัญชียังไม่ถูกต้อง เนื่องจาก ไม่มีการเช็ครหัสผ่าน
			3	yes	สามารถคำนวณจำนวนเงินในบัญชีได้อย่างถูกต้อง ตามฟังก์ชันกิจกรรมการทำงาน แต่การยืนยันตัวตนการเข้าถึงบัญชียังไม่ถูกต้องเนื่องจาก ไม่มีการเช็ครหัสผ่าน
GitHub Copilot	Python	Visitor	1	yes	สามารถยืนยันตัวตนก่อนการเข้าถึงบัญชีได้ถูกต้อง แต่ไม่มีการแสดงผลการคำนวณการฝากถอนเงิน

			2	yes	สามารถยืนยันตัวตนก่อนการเข้าถึงบัญชีได้ถูกต้อง แต่ไม่มีการแสดงผลการคำนวณการฝากถอนเงิน
			3	yes	สามารถยืนยันตัวตนก่อนการเข้าถึงบัญชีได้ถูกต้อง แต่ไม่มีการแสดงผลการคำนวณการฝากถอนเงิน
GitHub Copilot	Java	Composite	1	yes	สามารถยืนยันตัวตนก่อนการเข้าถึงบัญชีได้ถูกต้อง แต่ไม่มีการแสดงผลการคำนวณการฝากถอนเงิน
			2	yes	สามารถยืนยันตัวตนก่อนการเข้าถึงบัญชีได้ถูกต้อง แต่ไม่มีการแสดงผลการคำนวณการฝากถอนเงิน
			3	yes	สามารถยืนยันตัวตนก่อนการเข้าถึงบัญชีได้ถูกต้อง แต่ไม่มีการแสดงผลการคำนวณการฝากถอนเงิน
GitHub Copilot	Java	Visitor	1	yes	สามารถยืนยันตัวตนก่อนการเข้าถึงบัญชีได้ถูกต้อง แต่ไม่มีการแสดงผลการคำนวณการฝากถอนเงิน
			2	yes	สามารถยืนยันตัวตนก่อนการเข้าถึงบัญชีได้ถูกต้อง แต่ไม่มีการแสดงผลการคำนวณการฝากถอนเงิน
			3	yes	สามารถยืนยันตัวตนก่อนการเข้าถึงบัญชีได้ถูกต้อง แต่ไม่มีการแสดงผลการคำนวณการฝากถอนเงิน

3. นำโค้ดทดสอบที่ถูกสร้างขึ้นไปทดลองรันและบันทึกผล ได้ผลลัพธ์ดังนี้

Model AI	Programing language	Design pattern	Round	Executable	Quality Test Code	Fix Test Code
Chat-GPT 4o	Python	Composite	1	yes	ผลการทดสอบ Test Case Total: 10 Pass: 9 Fail: 1 ส่วนผลการทดสอบ Branch and statements coverage ได้ 92% วัดโดยใช้ pytest - cov	ไม่มีการแก้ไข
			2	yes	ผลการทดสอบ Test Case Total: 10 Pass: 10 Fail: 0 ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest - cov	ไม่มีการแก้ไข
			3	yes	ผลการทดสอบ Test Case Total: 13 Pass: 13 Fail: 0 ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest - cov	ไม่มีการแก้ไข

Chat-GPT 4o	Python	Visitor	1	yes	ผลการทดสอบ Test Case Total: 12 Pass: 10 Fail: 2 ส่วนผลการทดสอบ Branch and statements coverage ได้ 92% วัดโดยใช้ pytest - cov	ไม่มีการแก้ไข
			2	yes	ผลการทดสอบ Test Case Total: 12 Pass: 8 Fail: 4 ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest - cov	ไม่มีการแก้ไข
			3	yes	ผลการทดสอบ Test Case Total: 11 Pass: 11 Fail: 0 ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest - cov	ไม่มีการแก้ไข
Chat-GPT 4o	Java	Composite	1	yes	ผลการทดสอบ Test Case Total: 8 Pass: 8 Fail: 0 ผล การทดสอบ Branch coverage ได้ 100% statements coverage ได้ 100% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข

			2	yes	<p>ผลการทดสอบ Test Case Total: 10 Pass: 10 Fail: 0</p> <p>ผลการทดสอบ Branch coverage ได้ 90% statements coverage ได้ 100%</p> <p>วัดโดยใช้ JaCoCo</p>	แก้ไขโค้ดให้เปลี่ยนไปใช้ JUnit 5
			3	yes	<p>ผลการทดสอบ Test Case Total: 11 Pass: 11 Fail: 0</p> <p>ผลการทดสอบ Branch coverage ได้ 100% statements coverage ได้ 97%</p> <p>วัดโดยใช้ JaCoCo</p>	แก้ไขโค้ดให้เปลี่ยนไปใช้ JUnit 5
Chat-GPT 4o	Java	Visitor	1	yes	<p>ผลการทดสอบ Test Case Total: 11 Pass: 11 Fail: 0</p> <p>ผลการทดสอบ Branch coverage ได้ 100% statements coverage ได้ 98%</p> <p>วัดโดยใช้ JaCoCo</p>	แก้ไขโค้ดให้เปลี่ยนไปใช้ JUnit 5

			2	yes	<p>ผลการทดสอบ Test Case Total: 11 Pass: 11 Fail: 0</p> <p>ผลการทดสอบ Branch coverage ได้ 75% statements coverage ได้ 72%</p> <p>วัดโดยใช้ JaCoCo</p>	แก้ไขโค้ดให้เปลี่ยนไปใช้ JUnit 5
			3	yes	<p>ผลการทดสอบ Test Case Total: 11 Pass: 11 Fail: 0</p> <p>ผลการทดสอบ Branch coverage ได้ 90% statements coverage ได้ 99%</p> <p>วัดโดยใช้ JaCoCo</p>	แก้ไขโค้ดให้เปลี่ยนไปใช้ JUnit 5
Gemini 1.5 Flash	Python	Composite	1	yes	<p>ผลการทดสอบ Test Case Total: 11 Pass: 10 Fail: 1</p> <p>ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest - cov</p>	ไม่มีการแก้ไข
			2	yes	<p>ผลการทดสอบ Test Case Total: 11 Pass: 11 Fail: 0</p> <p>ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest - cov</p>	ไม่มีการแก้ไข

			3	no	ผลการทดสอบ เกิดความผิดพลาดเมื่อ test ที่ generate มา กลับเป็นต้องรับค่า User Input	สร้าง Mock และแก้ไขโค้ด
Gemini 1.5 Flash	Python	Visitor	1	yes	ผลการทดสอบ Test Case Total: 8 Pass: 3 Fail: 5 ส่วนผลการทดสอบ Branch and statements coverage ได้ 83% วัดโดยใช้ pytest -cov	ไม่มีการแก้ไข
			2	yes	ผลการทดสอบ Test Case Total: 10 Pass: 10 Fail: 0 ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest -cov	ไม่มีการแก้ไข
			3	yes	ผลการทดสอบ Test Case Total: 9 Pass: 4 Fail: 5 ส่วนผลการทดสอบ Branch and statements coverage ได้ 95% วัดโดยใช้ pytest -cov	ไม่มีการแก้ไข

Gemini 1.5 Flash	Java	Composite	1	yes	ผลการทดสอบ Test Case Total: 5 Pass: 5 Fail: 0 และ คุณภาพของเทสโค้ดด้านความ ความครอบคลุมได้ Coverage 100% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข
					ผลการทดสอบ Test Case Total: 9 Pass: 9 Fail: 0 และ คุณภาพของเทสโค้ดด้านความ ความครอบคลุมได้ Coverage 100% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข
					ผลการทดสอบ Test Case Total: 11 Pass: 10 Fail: 1 และ คุณภาพของเทสโค้ดด้าน ความความครอบคลุมได้ Coverage 90% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข
Gemini 1.5 Flash	Java	Visitor	1	yes	ผลการทดสอบ Test Case Total: 10 Pass: 10 Fail: 0 และ คุณภาพของเทสโค้ดด้าน ความความครอบคลุมได้ Coverage 100% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข

			2	yes	ผลการทดสอบ Test Case Total: 7 Pass: 7 Fail: 0 และ คุณภาพของเทสโค้ดด้านความ ความครอบคลุมได้ Coverage 100% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข
			3	yes	ผลการทดสอบ Test Case Total: 9 Pass: 8 Fail: 1 และ คุณภาพของเทสโค้ดด้านความ ความครอบคลุมได้ Coverage 89% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข
Gemini 1.5 Pro	Python	Composite	1	yes	ผลการทดสอบ Test Case Total: 12 Pass: 2 Fail: 10 ส่วนผลการทดสอบ Branch and statements coverage ได้ 90% วัดโดยใช้ pytest - cov	ไม่มีการแก้ไข
			2	no	ผลการทดสอบ เกิดความ ผิดพลาดเมื่อ test ที่ generate มา กล้ายเป็นต้อง รับค่า User Input	สร้าง Mock และแก้ไขโค้ด

			3	yes	ผลการทดสอบ Test Case Total: 14 Pass: 13 Fail: 1 ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest -cov	ไม่มีการแก้ไข
Gemini 1.5 Pro	Python	Visitor	1	yes	ผลการทดสอบ Test Case Total: 8 Pass: 3 Fail: 5 ส่วน ผลการทดสอบ Branch and statements coverage ได้ 83% วัดโดยใช้ pytest -cov	ไม่มีการแก้ไข
			2	yes	ผลการทดสอบ Test Case Total: 10 Pass: 10 Fail: 0 ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest -cov	ไม่มีการแก้ไข
			3	yes	ผลการทดสอบ Test Case Total: 9 Pass: 4 Fail: 5 ส่วน ผลการทดสอบ Branch and statements coverage ได้ 95% วัดโดยใช้ pytest -cov	ไม่มีการแก้ไข

Gemini 1.5 Pro	Java	Composite	1	yes	ผลการทดสอบ Test Case Total: 10 Pass: 10 Fail: 0 และ คุณภาพของเทสโค้ดด้าน ^{ความความครอบคลุมได้} Coverage 100% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข
			2	no	โค้ดที่สร้างไม่มีการใส่ assertion	แก้ไขโค้ด
			3	no	ผลการทดสอบ Test Case Total: 11 Pass: 6 Fail: 5 และ คุณภาพของเทสโค้ดด้าน ^{ความความครอบคลุมได้} Coverage 54% วัดโดยใช้ JaCoCo	แก้ไขโค้ด บางส่วนที่ไม่มี assertion
Gemini 1.5 Pro	Java	Visitor	1	yes	ผลการทดสอบ Test Case Total: 10 Pass: 10 Fail: 0 และ คุณภาพของเทสโค้ดด้าน ^{ความความครอบคลุมได้} Coverage 100% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข
			2	yes	ผลการทดสอบ Test Case Total: 8 Pass: 8 Fail: 0 และ ^{คุณภาพของเทสโค้ดด้านความความครอบคลุมได้} Coverage 100% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข

			3	yes	ผลการทดสอบ Test Case Total: 10 Pass: 10 Fail: 0 และ คุณภาพของเทสโค้ดด้าน ความความครอบคลุมได้ Coverage 100% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข
GitHub Copilot	Python	Composite	1	yes	ผลการทดสอบ Test Case Total: 9 Pass: 9 Fail: 0 ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest - cov	ไม่มีการแก้ไข
			2	yes	ผลการทดสอบ Test Case Total: 11 Pass: 9 Fail: 2 ส่วนผลการทดสอบ Branch and statements coverage ได้ 97% วัดโดยใช้ pytest - cov	ไม่มีการแก้ไข
			3	yes	ผลการทดสอบ Test Case Total: 11 Pass: 11 Fail: 0 ส่วนผลการทดสอบ Branch and statements coverage ได้ 100% วัดโดยใช้ pytest - cov	ไม่มีการแก้ไข

GitHub Copilot	Python	Visitor	1	yes	<p>ผลการทดสอบ Test Case Total: 6 Pass: 5 Fail: 1 ส่วนผลการทดสอบ Branch and statements coverage ได้ 88% วัดโดยใช้ pytest - cov</p>	ไม่มีการแก้ไข
			2	yes	<p>ผลการทดสอบ Test Case Total: 6 Pass: 5 Fail: 1 ส่วนผลการทดสอบ Branch and statements coverage ได้ 88% วัดโดยใช้ pytest - cov</p>	ไม่มีการแก้ไข
			3	yes	<p>ผลการทดสอบ Test Case Total: 6 Pass: 5 Fail: 1 ส่วนผลการทดสอบ Branch and statements coverage ได้ 88% วัดโดยใช้ pytest - cov</p>	ไม่มีการแก้ไข

GitHub Copilot	Java	Composite	1	yes	<p>ผลการทดสอบ Test Case Total: 6 Pass: 6 Fail: 0</p> <p>ผลการทดสอบ Branch coverage ได้ 59% statements coverage ได้ 47% วัดโดยใช้ JaCoCo</p>	<p>แก้ไขโค้ดให้เปลี่ยนไปใช้ JUnit 5 และ แก้ไขการทดสอบที่ผิดพลาดเนื่องจากนำฟังก์ชันประเภท Void มากำหนดค่าให้กับตัวแปร String</p>
			2	yes	<p>ผลการทดสอบ Test Case Total: 6 Pass: 6 Fail: 0</p> <p>ผลการทดสอบ Branch coverage ได้ 68% statements coverage ได้ 48% วัดโดยใช้ JaCoCo</p>	<p>แก้ไขโค้ดให้เปลี่ยนไปใช้ JUnit 5 และ แก้ไขการทดสอบที่ผิดพลาดเนื่องจากนำฟังก์ชันประเภท Void มากำหนดค่าให้กับตัวแปร String</p>

					ผลการทดสอบ Test Case Total: 6 Pass: 4 Fail: 2 ผลการทดสอบ Branch coverage ได้ 60% statements coverage ได้ 45% วัดโดยใช้ JaCoCo	แก้ไขโค้ดให้ เปลี่ยนไปใช้ JUnit 5
GitHub Copilot	Java	Visitor	1	yes	ผลการทดสอบ Test Case Total: 6 Pass: 5 Fail: 1 ผลการทดสอบ Branch coverage ได้ 65% statements coverage ได้ 91% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข
			2	yes	ผลการทดสอบ Test Case Total: 6 Pass: 5 Fail: 1 ผลการทดสอบ Branch coverage ได้ 65% statements coverage ได้ 91% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข
			3	yes	ผลการทดสอบ Test Case Total: 6 Pass: 5 Fail: 1 ผลการทดสอบ Branch coverage ได้ 65% statements coverage ได้ 91% วัดโดยใช้ JaCoCo	ไม่มีการแก้ไข

4. วัดคุณภาพของโค้ด และ โค้ดทดสอบ โดยใช้ Static code analysis tools ได้ผลลัพธ์ดังนี้

4.1 Python programming language

4.1.1 Chat-GPT 4o

4.1.1.1 Composite Design Pattern

ตารางแสดงจำนวนเปอร์เซ็นต์ข้อผิดพลาดหรือการแจ้งเตือน

module	error	warning	refactor	convention
round3.unit_testAI_3	33.33	0	21.43	20.22
round1.unit_testAI	33.33	0	0	21.35
round2.unit_testAI_2	33.33	0	0	16.85
round3.codeAI_3	0	50	25	13.48
round2.codeAI_2	0	50	25	13.48
round1.codeAI_1	0	0	28.57	14.61

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ที่ 1 วัดโดยใช้ radon python library

ประเภท	ชื่อ	บรรทัด	ความซับซ้อน	เกรด
Class	Authentication	12	3	A
Method	Authentication.execute	20	3	A
Class	Withdrawal	40	3	A
Class	PinChange	61	3	A
Method	PinChange.execute	65	3	A
Class	ATMOperation	6	2	A
Class	Balancelnquiry	33	2	A
Method	Withdrawal.execute	44	2	A
Class	Deposit	52	2	A
Class	PrintReceipt	78	2	A
Class	ATMComposite	86	2	A
Method	ATMComposite.execute	96	2	A
Method	ATMOperation.execute	8	1	A
Method	Authentication.__init__	13	1	A
Method	Balancelnquiry.__init__	34	1	A

Method	BalanceInquiry.execute	37	1	A
Method	Withdrawal.__init__	41	1	A
Method	Deposit.__init__	53	1	A
Method	Deposit.execute	56	1	A
Method	PinChange.__init__	62	1	A
Method	PrintReceipt.execute	79	1	A
Method	ATMComposite.__init__	87	1	A
Method	ATMComposite.add_operation	90	1	A
Method	ATMComposite.remove_operation	93	1	A

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ radon python library

ประเภท	ชื่อ	บรรทัด	ความซับซ้อน	เกรด
Function	test_atm_composite	75	7	B
Function	test_authentication_success	10	2	A
Function	test_authentication_failure	15	2	A
Function	test_balance_inquiry	20	2	A
Function	test_withdrawal_success	27	2	A
Function	test_withdrawal_insufficient_funds	35	2	A
Function	test_deposit	43	2	A
Function	test_pin_change_success	51	2	A
Function	test_pin_change_failure	60	2	A
Function	test_print_receipt	68	2	A

ค่า Maintainability ของซอสโค้ดในรอบที่ 1 วัดโดยใช้ radon python library คือ A (69.46)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ radon python library คือ A (48.53)

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 2 วัดโดยใช้ radon python library

ประเภท	ชื่อ	บรรทัด	ความซับซ้อน	เกรด
Class	Authentication	11	3	A
Method	Authentication.execute	16	3	A
Class	Withdrawal	33	3	A
Class	PINChange	56	3	A

Method	PINChange.execute	62	3	A
Class	ATMComponent	5	2	A
Class	BalanceInquiry	25	2	A
Method	Withdrawal.execute	38	2	A
Class	Deposit	46	2	A
Class	PrintSlip	70	2	A
Class	ATMComposite	82	2	A
Method	ATMComposite.execute	89	2	A
Method	ATMComponent.execute	7	1	A
Method	Authentication.init	12	1	A
Method	BalanceInquiry.init	26	1	A
Method	BalanceInquiry.execute	29	1	A
Method	Withdrawal.init	34	1	A
Method	Deposit.init	47	1	A
Method	Deposit.execute	51	1	A
Method	PINChange.init	57	1	A
Method	PrintSlip.init	71	1	A
Method	PrintSlip.execute	74	1	A
Method	ATMComposite.init	83	1	A
Method	ATMComposite.add	86	1	A

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ radon python library

ประเภท	ชื่อ	บรรทัด	ความซับซ้อน	เกรด
Function	test_atm_composite	81	11	C
Function	test_print_slip	65	6	B
Function	test_authentication_success	11	4	A
Function	test_pin_change_success	52	4	A
Function	test_balance_inquiry	24	2	A
Function	test_withdrawal_success	31	2	A
Function	test_withdrawal_failure	38	2	A
Function	test_deposit	45	2	A
Function	test_authentication_failure	19	1	A
Function	test_pin_change_failure	60	1	A

ค่า Maintainability ของซอสโค้ดในรอบที่ 2 วัดโดยใช้ radon python library คือ A (71.76)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ radon python library คือ A (47.02)

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 3 วัดโดยใช้ radon python library

ประเภท	ชื่อ	บรรทัด	ความซับซ้อน	เกรด
Class	Authentication	11	4	A

Method	Authentication.execute	20	4	A
Class	Withdrawal	38	3	A
Class	PINChange	61	3	A
Method	PINChange.execute	68	3	A
Class	ATMComponent	5	2	A
Class	BalanceInquiry	30	2	A
Method	Withdrawal.execute	43	2	A
Class	Deposit	51	2	A
Class	PrintSlip	79	2	A
Class	ATMComposite	91	2	A
Method	ATMComposite.execute	98	2	A
Method	ATMComponent.execute	7	1	A
Method	Authentication.init	12	1	A
Method	BalanceInquiry.init	31	1	A
Method	BalanceInquiry.execute	34	1	A
Method	Withdrawal.init	39	1	A
Method	Deposit.init	52	1	A
Method	Deposit.execute	56	1	A
Method	PINChange.init	62	1	A
Method	PrintSlip.init	80	1	A
Method	PrintSlip.execute	83	1	A
Method	ATMComposite.init	92	1	A
Method	ATMComposite.add	95	1	A

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ radon python library

ประเภท	ชื่อ	บรรทัด	ความซับซ้อน	เกรด
Function	test_atm_composite	108:0	11	C
Function	test_print_slip	92:0	6	B
Function	test_authentication_success	12:0	4	A
Function	test_authentication_failure_max_attempts	30:0	4	A
Function	test_balance_inquiry	43:0	2	A
Function	test_withdrawal_success	50:0	2	A
Function	test_withdrawal_failure_insufficient_funds	57:0	2	A
Function	test_deposit	64:0	2	A
Function	test_pin_change_success	71:0	2	A
Function	test_pin_change_failure_incorrect_old_pin	78:0	2	A
Function	test_pin_change_failure_mismatched_new_pins	85:0	2	A
Function	test_authentication_failure_invalid_card	20:0	1	A

Function	test_authentication_failure_invalid_pin	25:0	1	A
----------	---	------	---	---

ค่า Maintainability ของซอสโค้ดในรอบที่ 3 วัดโดยใช้ radon python library คือ A (69.25)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ radon python library คือ A (45.39)

4.1.1.2 Visitor Design Pattern

ตารางแสดงจำนวนเปอร์เซ็นต์ข้อผิดพลาดหรือการแจ้งเตือน

โมดูล	ข้อผิดพลาด (%)	คำเตือน (%)	Refactor (%)	Convention (%)
round1[unitTest_Visitor1]	50	100	0	15.65
round3[unitTest_Visitor3]	25	0	20	13.04
round2[unitTest_Visitor2]	25	0	0	15.65
round2[codeAI_Visitor2]	0	0	53.33	16.52
round3[codeAI_Visitor3]	0	0	13.33	22.61
round1[codeAI_Visitor1]	0	0	13.33	16.52

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 1 วัดโดยใช้ radon python library

ชนิด	หมายเลขอรรถัด	ชื่อ	คะแนน
C	47	AuthenticationVisitor	3
M	55	visit_account (AuthenticationVisitor)	3
C	114	PinChangeVisitor	3
M	124	visit_account (PinChangeVisitor)	3
C	4	Visitor	2
C	18	ATM	2
C	29	Account	2
C	38	Transaction	2
M	51	visit_atm (AuthenticationVisitor)	2
C	69	BalanceInquiryVisitor	2
M	70	visit_atm (BalanceInquiryVisitor)	2
C	80	WithdrawalVisitor	2
M	84	visit_atm (WithdrawalVisitor)	2
M	88	visit_account (WithdrawalVisitor)	2
C	99	DepositVisitor	2
M	103	visit_atm (DepositVisitor)	2

M	120	visit_atm (PinChangeVisitor)	2
C	137	PrintReceiptVisitor	2
M	138	visit_atm (PrintReceiptVisitor)	2
M	6	visit_atm (Visitor)	1
M	10	visit_account (Visitor)	1
M	14	visit_transaction (Visitor)	1
M	19	init (ATM)	1
M	22	insert_card (ATM)	1
M	26	accept (ATM)	1
M	30	init (Account)	1
M	35	accept (Account)	1
M	39	init (Transaction)	1
M	43	accept (Transaction)	1
M	48	init (AuthenticationVisitor)	1
M	66	visit_transaction (AuthenticationVisitor)	1
M	74	visit_account (BalanceInquiryVisitor)	1
M	77	visit_transaction (BalanceInquiryVisitor)	1
M	81	init (WithdrawalVisitor)	1
M	96	visit_transaction (WithdrawalVisitor)	1
M	100	init (DepositVisitor)	1
M	107	visit_account (DepositVisitor)	1
M	111	visit_transaction (DepositVisitor)	1
M	115	init (PinChangeVisitor)	1
M	134	visit_transaction (PinChangeVisitor)	1
M	142	visit_account (PrintReceiptVisitor)	1
M	146	visit_transaction (PrintReceiptVisitor)	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ radon python library

ชนิด	หมายเลขบรรทัด	ชื่อ	คะแนน
Funtion	35	test_authentication_card_seized	4
Funtion	17	test_authentication_success	3
Funtion	26	test_authentication_incorrect_pin	3
Funtion	54	test_withdrawal_success	3
Funtion	63	test_withdrawal_insufficient_funds	3
Funtion	72	test_deposit_success	3
Funtion	81	test_pin_change_success	3

Funtion	90	test_pin_change_incorrect_old_pin	3
Funtion	99	test_pin_change_mismatch_new_pins	3
Funtion	46	test_balance_inquiry	2
Funtion	108	test_print_receipt	2
Funtion	116	test_print_transaction	2
Funtion	11	setup_atm	1

ค่า Maintainability ของซอฟต์แวร์ในรอบที่ 1 วัดโดยใช้ radon python library คือ A (63.83)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ radon python library คือ A (40.85)

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 2 วัดโดยใช้ radon python library

ประเภท	หมายเลขบรรทัด	ชื่อฟังก์ชัน / คลาส	ค่า A (คะแนน)
Method	87:4	ATMVisitorImplementation.visit_authentication	4
Class	86:0	ATMVisitorImplementation	3
Method	122:4	ATMVisitorImplementation.visit_change_pin	3
Class	4:0	ATMOperation	2
Class	10:0	Authentication	2
Class	21:0	CheckBalance	2
Class	28:0	WithdrawMoney	2
Class	36:0	DepositMoney	2
Class	44:0	ChangePin	2
Class	52:0	PrintReceipt	2
Class	60:0	ATMVisitor	2
Method	109:4	ATMVisitorImplementation.visit_withdraw_money	2
Method	6:4	ATMOperation.accept	1
Method	11:4	Authentication.init	1
Method	18:4	Authentication.accept	1
Method	22:4	CheckBalance.init	1
Method	25:4	CheckBalance.accept	1
Method	29:4	WithdrawMoney.init	1

Method	33:4	WithdrawMoney.accept	1
Method	37:4	DepositMoney.init	1
Method	41:4	DepositMoney.accept	1
Method	45:4	ChangePin.init	1
Method	49:4	ChangePin.accept	1
Method	53:4	PrintReceipt.init	1
Method	56:4	PrintReceipt.accept	1
Method	62:4	ATMVisitor.visit_authentication	1
Method	66:4	ATMVisitor.visit_check_balance	1
Method	70:4	ATMVisitor.visit_withdraw_money	1
Method	74:4	ATMVisitor.visit_deposit_money	1
Method	78:4	ATMVisitor.visit_change_pin	1
Method	82:4	ATMVisitor.visit_print_receipt	1
Method	106:4	ATMVisitorImplementation.visit_check_balance	1
Method	117:4	ATMVisitorImplementation.visit_deposit_money	1
Method	135:4	ATMVisitorImplementation.visit_print_receipt	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ radon python library

ประเภท	หมายเลขบรรทัด	ชื่อฟังก์ชัน	คะแนน
Function	17:0	test_authentication	B (6)
Function	41:0	test_withdraw_money_success	A (4)
Function	62:0	test_deposit_money	A (4)
Function	52:0	test_withdraw_money_insufficient_funds	A (3)
Function	73:0	test_change_pin_success	A (3)
Function	83:0	test_change_pin_incorrect_old	A (3)
Function	93:0	test_change_pin_mismatch	A (3)
Function	103:0	test_print_receipt	A (3)
Function	32:0	test_check_balance	A (2)

ค่า Maintainability ของซอฟต์แวร์ในรอบที่ 2 วัดโดยใช้ radon python library คือ A (64.06)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ radon python library คือ A (58.65)

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 3 วัดโดยใช้ radon python library

ประเภท	หมายเลขอาร์ทัค	ชื่อคลาส/ฟังก์ชัน	คะแนน
Method	61:4	Authentication.authenticate	A (4)
Class	56:0	Authentication	A (3)
Class	118:0	PinChange	A (3)
Method	119:4	PinChange.change_pin	A (3)
Class	4:0	ATMVisitor	A (2)
Class	30:0	ATMOperation	A (2)
Class	50:0	ATMFeature	A (2)
Class	81:0	BalanceInquiry	A (2)
Class	91:0	Withdrawal	A (2)
Method	95:4	Withdrawal.process_withdrawal	A (2)
Class	106:0	Deposit	A (2)
Class	134:0	PrintReceipt	A (2)
Method	6:4	ATMVisitor.visit_authentication	A (1)
Method	10:4	ATMVisitor.visit_balance_inquiry	A (1)
Method	14:4	ATMVisitor.visit_withdrawal	A (1)
Method	18:4	ATMVisitor.visit_deposit	A (1)
Method	22:4	ATMVisitor.visit_pin_change	A (1)
Method	26:4	ATMVisitor.visit_print_receipt	A (1)
Method	31:4	ATMOperation.visit_authentication	A (1)
Method	34:4	ATMOperation.visit_balance_inquiry	A (1)
Method	37:4	ATMOperation.visit_withdrawal	A (1)
Method	40:4	ATMOperation.visit_deposit	A (1)
Method	43:4	ATMOperation.visit_pin_change	A (1)
Method	46:4	ATMOperation.visit_print_receipt	A (1)
Method	52:4	ATMFeature.accept	A (1)
Method	57:4	Authentication.init	A (1)
Method	74:4	Authentication.verify_pin	A (1)

Method	78:4	Authentication.accept	A (1)
Method	82:4	BalanceInquiry.init	A (1)
Method	85:4	BalanceInquiry.display_balance	A (1)
Method	88:4	BalanceInquiry.accept	A (1)
Method	92:4	Withdrawal.init	A (1)
Method	103:4	Withdrawal.accept	A (1)
Method	107:4	Deposit.init	A (1)
Method	110:4	Deposit.process_deposit	A (1)
Method	115:4	Deposit.accept	A (1)
Method	131:4	PinChange.accept	A (1)
Method	135:4	PrintReceipt.print_receipt	A (1)
Method	139:4	PrintReceipt.accept	A (1)

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ radon python library

Type	Line	Function Name	Grade
F	114	test_print_receipt	A (5)
F	60	test_withdrawal_insufficient_funds	A (3)
F	14	test_authentication_success	A (2)
F	23	test_authentication_failure_max_attempts	A (2)
F	32	test_authentication_partial_failure_then_success	A (2)
F	41	test_balance_inquiry	A (2)
F	51	test_withdrawal_success	A (2)
F	72	test_deposit	A (2)
F	81	test_pin_change_success	A (2)
F	92	test_pin_change_incorrect_old_pin	A (2)
F	103	test_pin_change_mismatch_new_pin	A (2)

ค่า Maintainability ของซอสโค้ดในรอบที่ 3 วัดโดยใช้ radon python library คือ A (63.75)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ radon python library คือ A (47.58)

4.1.2 Gemini 1.5 Flash

4.1.2.1 Composite Design Pattern

ตารางแสดงจำนวนเปอร์เซ็นต์ข้อผิดพลาดหรือการแจ้งเตือน

Module	Error	Warning	Refactor	Convention
round3.unit_test	38.46	0	10	23.94
round2.unit_test	34.62	0	0	28.87
round1.unit_test	26.92	100	0	19.01
round1.code	0	0	40	9.15
round2.code	0	0	30	10.56
round3.code	0	0	20	8.45

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 1 วัดโดยใช้ radon python library

Type	Name	Complexity
Class	PINVerification	3
Method	PINVerification.execute	3
Class	CashWithdrawal	3
Class	PINChange	3
Method	PINChange.execute	3
Class	ReceiptPrinting	3
Class	ATMComposite	3
Method	ATMComposite.execute	3
Class	ATMComponent	2
Class	ATMCardReader	2
Class	BalanceDisplay	2
Method	CashWithdrawal.execute	2
Class	Deposit	2

Method	ReceiptPrinting.execute	2
Method	ATMComponent.execute	1
Method	ATMCardReader.execute	1
Method	PINVerification.init	1
Method	BalanceDisplay.init	1
Method	BalanceDisplay.execute	1
Method	CashWithdrawal.init	1
Method	Deposit.init	1
Method	Deposit.execute	1
Method	PINChange.init	1
Method	ATMComposite.init	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ทดสอบในรอบที่ 1 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	test_atm_card_reader	2
Function	test_pin_verification_success	2
Function	test_pin_verification_failure	2
Function	test_balance_display	2
Function	test_cash_withdrawal_sufficient_funds	2
Function	test_cash_withdrawal_insufficient_funds	2
Function	test_deposit	2
Function	test_pin_change_success	2
Function	test_pin_change_incorrect_old_pin	2
Function	test_pin_change_mismatched_new_pin	2
Function	test_receipt_printing_yes	2
Function	test_receipt_printing_no	2

Function	balance_display	1
Function	cash_withdrawal	1
Function	deposit	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 2 วัดโดยใช้ radon python library

Type	Name	Complexity
Method	Authentication.execute	5
Class	Authentication	4
Class	CashWithdrawal	3
Class	PinChange	3
Method	PinChange.execute	3
Class	ATMComponent	2
Class	Leaf	2
Class	Composite	2
Method	Composite.execute	2
Class	BalanceInquiry	2
Method	CashWithdrawal.execute	2
Class	Deposit	2
Class	ReceiptPrinting	2
Method	ATMComponent.execute	1
Method	Leaf.init	1
Method	Leaf.execute	1
Method	Composite.init	1
Method	Composite.add	1

Method	Composite.remove	1
Method	Authentication.init	1
Method	BalanceInquiry.init	1
Method	BalanceInquiry.execute	1
Method	CashWithdrawal.init	1
Method	Deposit.init	1
Method	Deposit.execute	1
Method	PinChange.init	1
Method	ReceiptPrinting.init	1
Method	ReceiptPrinting.execute	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ที่ 2 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	test_Composite_initialization	4
Function	test_Composite_add_and_remove	4
Function	test_Leaf_initialization	3
Function	test_Leaf_execute	2
Function	test_Composite_execute	2
Function	test_Authentication_successful_login	2
Function	test_Authentication_failed_login	2
Function	test_BalanceInquiry_execute	2
Function	test_CashWithdrawal_successful_withdrawal	2
Function	test_CashWithdrawal_insufficient_funds	2
Function	test_Deposit_execute	2
Function	test_PinChange_successful_change	2

Function	test_PinChange_incorrect_old_pin	2
Function	test_PinChange_new_pins_mismatch	2
Function	test_ReceiptPrinting_execute	2
Function	mock_input	1
Function	test_ATMComponent_abstract_methods	1

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 3 วัดโดยใช้ radon python library

Type	Name	Complexity
Class	ATMComponent	2
Class	Leaf	2
Class	Composite	2
Method	Composite.execute	2
Class	Authentication	2
Class	BalanceDisplay	2
Class	CashWithdrawal	2
Class	Deposit	2
Class	PinChange	2
Class	ReceiptPrinting	2
Class	ATMSystem	2
Method	ATMComponent.execute	1
Method	Leaf.init	1
Method	Leaf.execute	1
Method	Composite.init	1
Method	Composite.add	1
Method	Composite.remove	1

Method	Authentication.init	1
Method	BalanceDisplay.init	1
Method	CashWithdrawal.init	1
Method	Deposit.init	1
Method	PinChange.init	1
Method	ReceiptPrinting.init	1
Method	ATMSystem.init	1
Method	ATMSystem.perform_transaction	1

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	test_Composite_creation	4
Function	test_Composite_add_remove	4
Function	test_CashWithdrawal	4
Function	test_Deposit	4
Function	test_PinChange	4
Function	test_ATMSystem_creation	4
Function	test_Leaf_creation	3
Function	test_ATMComponent_abstract	1
Function	test_Leaf_execute	1
Function	test_Composite_execute	1
Function	test_ATMSystem_perform_transaction	1

ตารางแสดงค่า Maintainability Index ที่ได้จากไลบรารี Radon Python

Round	1	2	3
-------	---	---	---

Code	50.74	50.47	76.71
Unit test	75.04	64.47	60.5

4.1.2.2 Visitor Design Pattern

ตารางแสดงจำนวนเปอร์เซ็นต์ข้อผิดพลาดหรือการแจ้งเตือน

Module	Error	Warning	Refactor	Convention
round3.unit_test	38.89	2.56	11.11	13.13
round1.unit_test	33.33	46.15	0	14.14
round2.unit_test	27.78	30.77	0	15.15
round1.code	0	10.26	11.11	19.19
round3.code	0	7.69	66.67	20.2
round2.code	0	2.56	11.11	18.18

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 1 วัดโดยใช้ radon python library

Type	Name	Complexity
Method	ATM.run	4
Class	ATM	4
Method	ATM.authenticate	4
Class	ATMCard	2
Class	ATMVisitor	2
Class	BalanceDisplayVisitor	2
Class	CashWithdrawalVisitor	2
Method	CashWithdrawalVisitor.visit_cash_withdrawal	2
Class	DepositVisitor	2
Class	PINChangeVisitor	2

Method	PINChangeVisitor.visit_pin_change	2
Method	ATMCard.init	1
Method	ATMCard.get_balance	1
Method	ATMCard.update_balance	1
Method	ATMCard.change_pin	1
Method	ATMVisitor.visit_balance_display	1
Method	ATMVisitor.visit_cash_withdrawal	1
Method	ATMVisitor.visit_deposit	1
Method	ATMVisitor.visit_pin_change	1
Method	BalanceDisplayVisitor.visit_balance_display	1
Method	BalanceDisplayVisitor.visit_cash_withdrawal	1
Method	BalanceDisplayVisitor.visit_deposit	1
Method	BalanceDisplayVisitor.visit_pin_change	1
Method	CashWithdrawalVisitor.visit_balance_display	1
Method	CashWithdrawalVisitor.visit_deposit	1
Method	CashWithdrawalVisitor.visit_pin_change	1
Method	DepositVisitor.visit_balance_display	1
Method	DepositVisitor.visit_cash_withdrawal	1
Method	DepositVisitor.visit_deposit	1
Method	DepositVisitor.visit_pin_change	1
Method	PINChangeVisitor.visit_balance_display	1
Method	PINChangeVisitor.visit_cash_withdrawal	1
Method	PINChangeVisitor.visit_deposit	1
Method	ATM.init	1
Method	ATM.process_transaction	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ที่ทดสอบในรอบที่ 1 วัดโดยใช้ radon python library

Type	Name	Complexity

Function	test_authenticate_success	4
Function	test_cash_withdrawal_sufficient_funds	3
Function	test_cash_withdrawal_insufficient_fund s	3
Function	test_deposit	3
Function	test_authenticate_invalid_card	2
Function	test_authenticate_incorrect_pin	2
Function	test_balance_display	2
Function	test_pin_change	2
Function	atm	1
Function	atm_card	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 2 วัดโดยใช้ radon python library

Type	Name	Complexity
Method	ATM.authenticate	4
Class	ATM	3
Method	ATM.process_transaction	3
Method	PINChangeVisitor.execute	3
Class	ATMVisitor	2
Class	BalanceInquiryVisitor	2
Class	CashWithdrawalVisitor	2
Method	CashWithdrawalVisitor.execute	2
Class	DepositVisitor	2
Class	PINChangeVisitor	2
Method	ATM.init	1

Method	ATM.add_user	1
Method	ATM.print_receipt	1
Method	ATMVisitor.execute	1
Method	ATMVisitor.get_pin	1
Method	ATMVisitor.should_print_receipt	1
Method	ATMVisitor.get_transaction_details	1
Method	BalanceInquiryVisitor.init	1
Method	BalanceInquiryVisitor.execute	1
Method	BalanceInquiryVisitor.get_pin	1
Method	BalanceInquiryVisitor.should_print_receipt	1
Method	BalanceInquiryVisitor.get_transaction_details	1
Method	CashWithdrawalVisitor.init	1
Method	CashWithdrawalVisitor.get_pin	1
Method	CashWithdrawalVisitor.should_print_receipt	1
Method	CashWithdrawalVisitor.get_transaction_details	1
Method	DepositVisitor.init	1
Method	DepositVisitor.execute	1
Method	DepositVisitor.get_pin	1
Method	DepositVisitor.should_print_receipt	1
Method	DepositVisitor.get_transaction_details	1
Method	PINChangeVisitor.init	1
Method	PINChangeVisitor.get_pin	1

Method	PINChangeVisitor.should_print_receipt	1
Method	PINChangeVisitor.get_transaction_details	1
Method	PINChangeVisitor.confirm_new_pin	1

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	test_cash_withdrawal_success	3
Function	test_cash_withdrawal_failure	3
Function	test_deposit	3
Function	test_pin_change_success	3
Function	test_pin_change_failure	3
Function	test_authenticate_success	2
Function	test_authenticate_failure	2
Function	test_authenticate_max_attempts	2
Function	test_balance_inquiry	2
Function	atm	1
Function	test_print_receipt	1

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 3 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	main	10
Method	ATM.authenticate	3
Class	CashWithdrawalVisitor	3
Class	PINChangeVisitor	3
Class	ReceiptPrintingVisitor	3

Class	Account	2
Method	Account.withdraw	2
Class	ATM	2
Class	ATMVisitor	2
Class	BalanceDisplayVisitor	2
Method	CashWithdrawalVisitor.visit	2
Class	DepositVisitor	2
Method	PINChangeVisitor.visit	2
Method	ReceiptPrintingVisitor.visit	2
Method	Account.init	1
Method	Account.change_pin	1
Method	ATM.init	1
Method	ATM.add_account	1
Method	ATM.process_transaction	1
Method	ATM.get_transaction_history	1
Method	ATMVisitor.visit	1
Method	BalanceDisplayVisitor.visit	1
Method	CashWithdrawalVisitor.init	1
Method	DepositVisitor.init	1
Method	DepositVisitor.visit	1
Method	PINChangeVisitor.init	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ที่ได้ทดสอบในรอบที่ 3 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	test_cash_withdrawal_success	3

Function	test_cash_withdrawal_failure	3
Function	test_deposit	3
Function	test_pin_change	3
Function	test_add_account	2
Function	test_authenticate_success	2
Function	test_authenticate_failure	2
Function	test_balance_display	2
Function	test_receipt_printing	2
Function	test_authenticate_max_attempts	1
Function	atm	1

ตารางแสดงค่า Maintainability Index ที่ได้จากไลบรารี Radon Python

Round	1	2	3
code.py	43.79	39.82	43.33
unit_test.py	62.55	56.6	42.89

4.1.3 Gemini 1.5 Pro

4.1.3.1 Composite Design Pattern

ตารางแสดงจำนวนเปอร์เซ็นต์ข้อผิดพลาดหรือการแจ้งเตือน

Module	Error	Warning	Refactor	Convention
round3.unit_test	38.46	0	10	23.94
round2.unit_test	34.62	0	0	28.87
round1.unit_test	26.92	100	0	19.01
round1.code	0	0	40	9.15
round2.code	0	0	30	10.56

round3.code	0	0	20	8.45
-------------	---	---	----	------

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 1 วัดโดยใช้ radon python library

Type	Name	Complexity
Class	PINVerification	3
Method	PINVerification.execute	3
Class	CashWithdrawal	3
Class	PINChange	3
Method	PINChange.execute	3
Class	ReceiptPrinting	3
Class	ATMComposite	3
Method	ATMComposite.execute	3
Class	ATMComponent	2
Class	ATMCardReader	2
Class	BalanceDisplay	2
Method	CashWithdrawal.execute	2
Class	Deposit	2
Method	ReceiptPrinting.execute	2
Method	ATMComponent.execute	1
Method	ATMCardReader.execute	1
Method	PINVerification.init	1
Method	BalanceDisplay.init	1

Method	BalanceDisplay.execute	1
Method	CashWithdrawal.init	1
Method	Deposit.init	1
Method	Deposit.execute	1
Method	PINChange.init	1
Method	ATMComposite.init	1
Method	ATMComposite.add	1

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	test_atm_card_reader	2
Function	test_pin_verification_success	2
Function	test_pin_verification_failure	2
Function	test_balance_display	2
Function	test_cash_withdrawal_sufficient_funds	2
Function	test_cash_withdrawal_insufficient_funds	2
Function	test_deposit	2
Function	test_pin_change_success	2
Function	test_pin_change_incorrect_old_pin	2
Function	test_pin_change_mismatched_new_pin	2
Function	test_receipt_printing_yes	2
Function	test_receipt_printing_no	2
Function	balance_display	1
Function	cash_withdrawal	1

Function	deposit	1
----------	---------	---

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 2 วัดโดยใช้ radon python library

Type	Name	Complexity
Method	Authentication.execute	5
Class	Authentication	4
Class	CashWithdrawal	3
Class	PinChange	3
Method	PinChange.execute	3
Class	ATMComponent	2
Class	Leaf	2
Class	Composite	2
Method	Composite.execute	2
Class	BalanceInquiry	2
Method	CashWithdrawal.execute	2
Class	Deposit	2
Class	ReceiptPrinting	2
Method	ATMComponent.execute	1
Method	Leaf.init	1
Method	Leaf.execute	1
Method	Composite.init	1
Method	Composite.add	1
Method	Composite.remove	1
Method	Authentication.init	1
Method	BalanceInquiry.init	1

Method	BalanceInquiry.execute	1
Method	CashWithdrawal.init	1
Method	Deposit.init	1
Method	Deposit.execute	1
Method	PinChange.init	1
Method	ReceiptPrinting.init	1
Method	ReceiptPrinting.execute	1

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	test_Composite_initialization	4
Function	test_Composite_add_and_remove	4
Function	test_Leaf_initialization	3
Function	test_Leaf_execute	2
Function	test_Composite_execute	2
Function	test_Authentication_successful_login	2
Function	test_Authentication_failed_login	2
Function	test_BalanceInquiry_execute	2
Function	test_CashWithdrawal_successful_withdrawal	2
Function	test_CashWithdrawal_insufficient_funds	2
Function	test_Deposit_execute	2
Function	test_PinChange_successful_change	2
Function	test_PinChange_incorrect_old_pin	2
Function	test_PinChange_new_pins_mismatch	2

Function	test_ReceiptPrinting_execute	2
Function	mock_input	1
Function	test_ATMComponent_abstract_methods	1

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 3 วัดโดยใช้ radon python library

Type	Name	Complexity
Class	ATMComponent	2
Class	Leaf	2
Class	Composite	2
Method	Composite.execute	2
Class	Authentication	2
Class	BalanceDisplay	2
Class	CashWithdrawal	2
Class	Deposit	2
Class	PinChange	2
Class	ReceiptPrinting	2
Class	ATMSystem	2
Method	ATMComponent.execute	1
Method	Leaf.init	1
Method	Leaf.execute	1
Method	Composite.init	1
Method	Composite.add	1
Method	Composite.remove	1
Method	Authentication.init	1
Method	BalanceDisplay.init	1

Method	CashWithdrawal.init	1
Method	Deposit.init	1
Method	PinChange.init	1
Method	ReceiptPrinting.init	1
Method	ATMSystem.init	1
Method	ATMSystem.perform_transaction	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ทดสอบในรอบที่ 3 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	test_Composite_creation	4
Function	test_Composite_add_remove	4
Function	test_CashWithdrawal	4
Function	test_Deposit	4
Function	test_PinChange	4
Function	test_ATMSystem_creation	4
Function	test_Leaf_creation	3
Function	test_ATMComponent_abstract	1
Function	test_Leaf_execute	1
Function	test_Composite_execute	1
Function	test_ATMSystem_perform_transaction	1

ตารางแสดงค่า Maintainability Index ที่ได้จากไลบรารี Radon Python

Round	1	2	3
code.py	50.74	50.47	76.71
unit_test.py	75.04	64.47	60.5

4.1.3.2 Visitor Design Pattern

ตารางแสดงจำนวนเบอร์เซ็นต์ข้อผิดพลาดหรือการแจ้งเตือน

Module	Error	Warning	Refactor	Convention
round2.unit_test	43.75	23.44	14.71	19.55
round1.unit_test	43.75	23.44	0	11.28
round3.unit_test	12.5	26.56	14.71	16.54
round3.code	0	14.06	26.47	24.81
round2.code	0	10.94	26.47	11.28
round1.code	0	1.56	17.65	16.54

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 1 วัดโดยใช้ radon python library

Type	Name	Complexity
Method	ATM.run	9
Method	Account.verify_pin	4
Class	ATM	4
Class	Account	3
Method	ATM.authenticate_user	3
Class	ATMVisitor	2
Class	TransactionVisitor	2
Method	TransactionVisitor.visit_cash_withdrawal	2
Method	TransactionVisitor.visit_pin_change	2
Class	ATMElement	2
Class	BalanceInquiry	2
Class	CashWithdrawal	2

Class	Deposit	2
Class	PINChange	2
Method	ATMVisitor.visit_balance_inquiry	1
Method	ATMVisitor.visit_cash_withdrawal	1
Method	ATMVisitor.visit_deposit	1
Method	ATMVisitor.visit_pin_change	1
Method	TransactionVisitor.init	1
Method	TransactionVisitor.visit_balance_inquiry	1
Method	TransactionVisitor.visit_deposit	1
Method	ATMElement.accept	1
Method	BalanceInquiry.accept	1
Method	CashWithdrawal.init	1
Method	CashWithdrawal.accept	1
Method	Deposit.init	1
Method	Deposit.accept	1
Method	PINChange.init	1
Method	PINChange.accept	1
Method	Account.init	1
Method	Account.change_pin	1
Method	ATM.init	1
Method	ATM.add_account	1
Method	ATM.dispense_cash	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ที่ได้ทดสอบในรอบที่ 1 วัดโดยใช้ radon python library

Type	Name	Complexity

Function	test_pin_change_success	3
Function	test_pin_change_incorrect_old_pin	3
Function	test_verify_pin_card_retained	3
Function	test_balance_inquiry	2
Function	test_cash_withdrawal_sufficient_funds	2
Function	test_cash_withdrawal_insufficient_fund s	2
Function	test_deposit	2
Function	test_verify_pin_success	2
Function	test_verify_pin_failure	2
Function	test_dispense_cash	2
Function	atm_with_account	1

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 2 วัดโดยใช้ radon python library

Type	Name	Complexity
Method	WithdrawalVisitor.visit_display	6
Method	PinChangeVisitor.visit_pin_pad	5
Class	DepositSlot	4
Class	ReceiptPrinter	4
Method	AuthenticationVisitor.visit_pin_pad	4
Method	DepositSlot.receive_deposit	3
Method	ReceiptPrinter.print_receipt	3
Method	AuthenticationVisitor.authenticate	3
Class	WithdrawalVisitor	3
Method	WithdrawalVisitor.visit_cash_dispenser	3

Method	WithdrawalVisitor.visit_receipt_printer	3
Method	DepositVisitor.visit_deposit_slot	3
Method	DepositVisitor.visit_receipt_printer	3
Class	ATMPartVisitor	2
Class	ATM	2
Class	CardReader	2
Class	PinPad	2
Class	Display	2
Class	CashDispenser	2
Class	AuthenticationVisitor	2
Class	BalanceInquiryVisitor	2
Method	BalanceInquiryVisitor.visit_display	2
Class	DepositVisitor	2
Class	PinChangeVisitor	2
Class	Account	2
Method	ATMPartVisitor.visit_card_reader	1
Method	ATMPartVisitor.visit_pin_pad	1
Method	ATMPartVisitor.visit_display	1
Method	ATMPartVisitor.visit_cash_dispenser	1
Method	ATMPartVisitor.visit_deposit_slot	1
Method	ATMPartVisitor.visit_receipt_printer	1
Method	ATM.init	1
Method	ATM.accept	1
Method	CardReader.read_card	1
Method	PinPad.get_pin	1

Method	Display.show_message	1
Method	Display.show_balance	1
Method	CashDispenser.dispense_cash	1
Method	AuthenticationVisitor.init	1
Method	AuthenticationVisitor.visit_card_reader	1
Method	AuthenticationVisitor.visit_display	1
Method	AuthenticationVisitor.visit_cash_dispenser	1
Method	AuthenticationVisitor.visit_deposit_slot	1
Method	AuthenticationVisitor.visit_receipt_printer	1
Method	BalanceInquiryVisitor.init	1
Method	BalanceInquiryVisitor.visit_card_reader	1
Method	BalanceInquiryVisitor.visit_pin_pad	1
Method	BalanceInquiryVisitor.visit_cash_dispenser	1
Method	BalanceInquiryVisitor.visit_deposit_slot	1
Method	BalanceInquiryVisitor.visit_receipt_printer	1
Method	WithdrawalVisitor.init	1
Method	WithdrawalVisitor.visit_card_reader	1
Method	WithdrawalVisitor.visit_pin_pad	1
Method	WithdrawalVisitor.visit_deposit_slot	1
Method	DepositVisitor.init	1
Method	DepositVisitor.visit_display	1
Method	DepositVisitor.visit_card_reader	1
Method	DepositVisitor.visit_pin_pad	1
Method	DepositVisitor.visit_cash_dispenser	1
Method	PinChangeVisitor.init	1

Method	PinChangeVisitor.visit_card_reader	1
Method	PinChangeVisitor.visit_display	1
Method	PinChangeVisitor.visit_cash_dispenser	1
Method	PinChangeVisitor.visit_deposit_slot	1
Method	PinChangeVisitor.visit_receipt_printer	1
Method	Account.init	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ทดสอบในรอบที่ 2 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	test_withdrawal_success	5
Function	test_withdrawal_insufficient_funds	4
Function	test_deposit_success	4
Function	test_authentication_success	3
Function	test_authentication_incorrect_pin	2
Function	test_balance_inquiry	2
Function	test_pin_change_success	2
Function	test_pin_change_mismatch	2
Class	MockCardReader	2
Class	MockPinPad	2
Class	MockDisplay	2
Class	MockCashDispenser	2
Class	MockDepositSlot	2
Class	MockReceiptPrinter	2
Function	atm_with_account	1
Method	MockCardReader.init	1

Method	MockCardReader.read_card	1
Method	MockPinPad.init	1
Method	MockPinPad.get_pin	1
Method	MockDisplay.init	1
Method	MockDisplay.show_message	1
Method	MockDisplay.show_balance	1
Method	MockCashDispenser.init	1
Method	MockCashDispenser.dispense_cash	1
Method	MockDepositSlot.init	1
Method	MockDepositSlot.receive_deposit	1
Method	MockReceiptPrinter.init	1
Method	MockReceiptPrinter.print_receipt	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 3 วัดโดยใช้ radon python library

Type	Name	Complexity
Method	WithdrawalVisitor.visit_withdrawal	5
Method	DepositVisitor.visit_deposit	4
Class	ATM	4
Class	ATMPartVisitor	4
Class	ReceiptPrinter	3
Method	WithdrawalVisitor.visit_display	3
Method	ATMPartVisitor.visit_receipt_printer	3
Class	DepositSlot	3
Method	DepositSlot.receive_deposit	3
Method	ReceiptPrinter.print_receipt	3

Method	BalanceInquiryVisitor.visit_display	2
Class	BalanceInquiryVisitor	2
Class	AuthenticationVisitor	2
Method	ATMPartVisitor.visit_balance_inquiry	2
Class	CardReader	2
Class	PinPad	2
Class	CashDispenser	2
Method	AuthenticationVisitor.visit_card_reader	1
Method	AuthenticationVisitor.authenticate	1
Method	ATMPartVisitor.visit_card_reader	1
Method	ATM.init	1
Method	ATM.accept	1
Method	CardReader.read_card	1
Method	PinPad.get_pin	1
Method	CashDispenser.dispense_cash	1
Method	ATMPartVisitor.init	1

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ที่ได้ทดสอบในรอบที่ 3 วัดโดยใช้ radon python library

Type	Name	Complexity
Function	test_withdrawal_success	4
Function	test_withdrawal_insufficient_funds	3
Function	test_deposit_success	3
Function	test_authentication_success	2
Function	test_balance_inquiry	2
Function	test_pin_change_success	2

Class	MockATM	2
Class	MockCardReader	1
Class	MockPinPad	1
Class	MockCashDispenser	1
Class	MockDepositSlot	1
Function	atm_with_account	1
Method	MockATM.init	1
Method	MockATM.accept	1
Method	MockCardReader.init	1
Method	MockCardReader.read_card	1
Method	MockPinPad.init	1
Method	MockPinPad.get_pin	1
Method	MockCashDispenser.init	1
Method	MockCashDispenser.dispense_cash	1
Method	MockDepositSlot.init	1
Method	MockDepositSlot.receive_deposit	1

ตารางแสดงค่า Maintainability Index ที่ได้จากไลบรารี Radon Python

Round	1	2	3
code.py	43.86	35.36	47.19
unit_test.py	50.57	50.79	55.88

4.1.4 GitHub Copilot

4.1.4.1 Composite Design Pattern

ตารางแสดงจำนวนเปอร์เซ็นต์ข้อผิดพลาดหรือการแจ้งเตือน

Module	Error	Warning	Refactor	Convention
round3.unit_test	33.33	35.71	34.38	15.31
round2.unit_test	33.33	25	0	15.31
round1.unit_test	33.33	25	0	13.27
round2.code_2	0	14.29	21.88	19.39
round3.code_3	0	0	21.88	18.37
round1.code_1	0	0	21.88	18.37

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 1 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
C	29	Withdrawal	A	3
C	52	PinChange	A	3
C	4	ATMOperation	A	2
C	10	Authentication	A	2
C	21	BalanceInquiry	A	2
M	34	Withdrawal.execute	A	2
C	42	Deposit	A	2
M	58	PinChange.execute	A	2
C	66	PrintReceipt	A	2
C	76	ATMComposite	A	2
M	83	ATMComposite.execute	A	2
C	88	Account	A	2
M	6	ATMOperation.execute	A	1
M	11	Authentication.init	A	1
M	16	Authentication.execute	A	1
M	22	BalanceInquiry.init	A	1
M	25	BalanceInquiry.execute	A	1
M	30	Withdrawal.init	A	1
M	43	Deposit.init	A	1

M	47	Deposit.execute	A	1
M	53	PinChange.init	A	1
M	67	PrintReceipt.init	A	1
M	70	PrintReceipt.execute	A	1
M	77	ATMComposite.init	A	1
M	80	ATMComposite.add_operation	A	1
M	89	Account.init	A	1
M	93	Account.get_balance	A	1
M	96	Account.withdraw	A	1
M	99	Account.deposit	A	1
M	102	Account.verify_pin	A	1
M	105	Account.change_pin	A	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
F	19	test_withdrawal_success	A	2
F	24	test_withdrawal_insufficient_funds	A	2
F	29	test_deposit	A	2
F	34	test_pin_change_success	A	2
F	39	test_pin_change_failure	A	2
F	5	account	A	1
F	8	test_authentication_success	A	1
F	14	test_balance_inquiry	A	1
F	44	test_print_receipt	A	1
F	49	test_atm_composite	A	1

ค่า Maintainability ของซอสโค้ดในรอบที่ 1 วัดโดยใช้ radon python library คือ A (73.19)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ radon python library คือ A (69.09)

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 2 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
C	10	Authentication	A	3
C	37	Withdrawal	A	3
C	60	PinChange	A	3
C	4	ATMOperation	A	2
M	18	Authentication.execute	A	2
C	29	BalanceInquiry	A	2
M	42	Withdrawal.execute	A	2
C	50	Deposit	A	2
M	66	PinChange.execute	A	2
C	74	PrintReceipt	A	2
C	84	ATMComposite	A	2
M	91	ATMComposite.execute	A	2
C	96	Account	A	2
M	6	ATMOperation.execute	A	1
M	11	Authentication.init	A	1
M	30	BalanceInquiry.init	A	1
M	33	BalanceInquiry.execute	A	1
M	38	Withdrawal.init	A	1
M	51	Deposit.init	A	1
M	55	Deposit.execute	A	1
M	61	PinChange.init	A	1
M	75	PrintReceipt.init	A	1
M	78	PrintReceipt.execute	A	1
M	85	ATMComposite.init	A	1
M	88	ATMComposite.add_operation	A	1
M	97	Account.init	A	1
M	102	Account.get_balance	A	1
M	105	Account.withdraw	A	1
M	108	Account.deposit	A	1
M	111	Account.verify_pin	A	1

M	114	Account.change_pin	A	1
---	-----	--------------------	---	---

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
F	8	test_authentication_success	A	2
F	13	test_authentication_failure	A	2
F	18	test_authentication_max_attempts	A	2
F	24	test_balance_inquiry	A	2
F	29	test_withdrawal_success	A	2
F	34	test_withdrawal_insufficient_funds	A	2
F	39	test_deposit	A	2
F	44	test_pin_change_success	A	2
F	49	test_pin_change_failure	A	2

ค่า Maintainability ของซอสโค้ดในรอบที่ 2 วัดโดยใช้ radon python library คือ A (**62.72**)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ radon python library คือ A (**60.71**)

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 3 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
C	37	Authentication	A	3
M	43	Authentication.execute	A	3
C	60	Withdrawal	A	3
C	80	PinChange	A	3
C	3	ATMOperation	A	2
C	8	Account	A	2
M	22	Account.withdraw	A	2
M	31	Account.change_pin	A	2
C	53	BalanceInquiry	A	2

M	65	Withdrawal.execute	A	2
C	71	Deposit	A	2
M	86	PinChange.execute	A	2
C	92	PrintReceipt	A	2
C	99	ATMComposite	A	2
M	106	ATMComposite.execute	A	2
M	5	ATMOperation.execute	A	1
M	9	Account.init	A	1
M	16	Account.verify_pin	A	1
M	19	Account.get_balance	A	1
M	28	Account.deposit	A	1
M	38	Authentication.init	A	1
M	54	BalanceInquiry.init	A	1
M	57	BalanceInquiry.execute	A	1
M	61	Withdrawal.init	A	1
M	72	Deposit.init	A	1
M	76	Deposit.execute	A	1
M	81	PinChange.init	A	1
M	93	PrintReceipt.init	A	1
M	96	PrintReceipt.execute	A	1
M	100	ATMComposite.init	A	1
		ATMComposite.add_ope		
M	103	ration	A	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
F	18	test_authentication_max_attempts	A	4
F	8	test_authentication_success	A	2
F	13	test_authentication_failure	A	2
F	25	test_balance_inquiry	A	2

F	30	test_withdrawal_success	A	2
F	35	test_withdrawal_insufficient_funds	A	2
F	40	test_deposit	A	2
F	45	test_pin_change_success	A	2
F	50	test_pin_change_failure	A	2
F	55	account	A	1
F	61	test_print_receipt	A	1
F	61	test_atm_composite	A	1

ค่า Maintainability ของซอสโค้ดในรอบที่ 3 วัดโดยใช้ radon python library คือ A (**66.07**)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ radon python library คือ A (**59.64**)

4.1.4.2 Visitor Design Pattern

ตารางแสดงจำนวนเปอร์เซ็นต์ข้อผิดพลาดหรือการแจ้งเตือน

Module	Error	Warning	Refactor	Convention
round3.unit_test	33.33	33.33	15.38	9.57
round2.unit_test	33.33	33.33	0	9.57
round1.unit_test	33.33	33.33	0	9.57
round2.code_2	0	0	28.21	25.22
round1.code_1	0	0	28.21	24.35
round3.code_3	0	0	28.21	21.74

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 1 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score

M	70	ATM.authenticate	A	4
C	62	ATM	A	3
M	93	ATM.withdraw	A	3
M	113	ATM.change_pin	A	3
C	1	ATMOperation	A	2
C	5	AuthenticateUser	A	2
C	13	CheckBalance	A	2
C	17	WithdrawMoney	A	2
C	24	DepositMoney	A	2
C	31	ChangePin	A	2
C	39	PrintReceipt	A	2
C	43	ATMVisitor	A	2
M	85	ATM.check_balance	A	2
M	106	ATM.deposit	A	2
M	126	ATM.print_receipt	A	2
C	133	ATMVisitorImpl	A	2
M	2	ATMOperation.accept	A	1
M	6	AuthenticateUser.init	A	1
M	10	AuthenticateUser.accept	A	1
M	14	CheckBalance.accept	A	1
M	18	WithdrawMoney.init	A	1
M	21	WithdrawMoney.accept	A	1
M	25	DepositMoney.init	A	1
M	28	DepositMoney.accept	A	1
M	32	ChangePin.init	A	1
M	36	ChangePin.accept	A	1
M	40	PrintReceipt.accept	A	1
M	44	ATMVisitor.visit_authenticate_user	A	1
M	47	ATMVisitor.visit_check_balance	A	1
M	50	ATMVisitor.visit_withdraw_money	A	1
M	53	ATMVisitor.visit_deposit_money	A	1

M	56	ATMVisitor.visit_change_pin	A	1
M	59	ATMVisitor.visit_print_receipt	A	1
M	63	ATM.init	A	1
M	134	ATMVisitorImpl.init	A	1
M	137	ATMVisitorImpl.visit_authenticate_user	A	1
M	140	ATMVisitorImpl.visit_check_balance	A	1
M	143	ATMVisitorImpl.visit_withdraw_money	A	1
M	146	ATMVisitorImpl.visit_deposit_money	A	1
M	149	ATMVisitorImpl.visit_change_pin	A	1
M	152	ATMVisitorImpl.visit_print_receipt	A	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
F	12	test_authenticate_user	A	3
F	27	test_withdraw_money	A	3
F	43	test_change_pin	A	3
F	21	test_check_balance	A	2
F	37	test_deposit_money	A	2
F	5	atm	A	1
F	9	visitor	A	1
F	53	test_print_receipt	A	1

ค่า Maintainability ของซอฟต์แวร์ในรอบที่ 1 วัดโดยใช้ radon python library คือ A (57.54)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ radon python library คือ A (58.73)

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 2 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
M	70	ATM.authenticate	A	4
C	62	ATM	A	3
M	93	ATM.withdraw	A	3
M	113	ATM.change_pin	A	3
C	1	ATMOperation	A	2
C	5	AuthenticateUser	A	2
C	13	CheckBalance	A	2
C	17	WithdrawMoney	A	2
C	24	DepositMoney	A	2
C	31	ChangePin	A	2
C	39	PrintReceipt	A	2
C	43	ATMVisitor	A	2
M	85	ATM.check_balance	A	2
M	106	ATM.deposit	A	2
M	126	ATM.print_receipt	A	2
C	133	ATMVisitorImpl	A	2
M	2	ATMOperation.accept	A	1
M	6	AuthenticateUser.init	A	1
M	10	AuthenticateUser.accept	A	1
M	14	CheckBalance.accept	A	1
M	18	WithdrawMoney.init	A	1
M	21	WithdrawMoney.accept	A	1
M	25	DepositMoney.init	A	1
M	28	DepositMoney.accept	A	1
M	32	ChangePin.init	A	1
M	36	ChangePin.accept	A	1
M	40	PrintReceipt.accept	A	1
M	44	ATMVisitor.visit_authenticate_user	A	1

M	47	ATMVisitor.visit_check_balance	A	1
M	50	ATMVisitor.visit_withdraw_money	A	1
M	53	ATMVisitor.visit_deposit_money	A	1
M	56	ATMVisitor.visit_change_pin	A	1
M	59	ATMVisitor.visit_print_receipt	A	1
M	63	ATM.init	A	1
M	134	ATMVisitorImpl.init	A	1
M	137	ATMVisitorImpl.visit_authenticate_user	A	1
M	140	ATMVisitorImpl.visit_check_balance	A	1
M	143	ATMVisitorImpl.visit_withdraw_money	A	1
M	146	ATMVisitorImpl.visit_deposit_money	A	1
M	149	ATMVisitorImpl.visit_change_pin	A	1
M	152	ATMVisitorImpl.visit_print_receipt	A	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
F	12	test_authenticate_user	A	3
F	27	test_withdraw_money	A	3
F	43	test_change_pin	A	3
F	21	test_check_balance	A	2
F	37	test_deposit_money	A	2
F	5	atm	A	1
F	9	visitor	A	1
F	53	test_print_receipt	A	1

ค่า Maintainability ของซอฟต์แวร์ในรอบที่ 2 วัดโดยใช้ radon python library คือ A (57.54)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ radon python library คือ A (58.73)

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 3 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
M	70	ATM.authenticate	A	4
C	62	ATM	A	3
M	93	ATM.withdraw	A	3
M	113	ATM.change_pin	A	3
C	1	ATMOperation	A	2
C	5	AuthenticateUser	A	2
C	13	CheckBalance	A	2
C	17	WithdrawMoney	A	2
C	24	DepositMoney	A	2
C	31	ChangePin	A	2
C	39	PrintReceipt	A	2
C	43	ATMVisitor	A	2
M	85	ATM.check_balance	A	2
M	106	ATM.deposit	A	2
M	126	ATM.print_receipt	A	2
C	133	ATMVisitorImpl	A	2
M	2	ATMOperation.accept	A	1
M	6	AuthenticateUser.init	A	1
M	10	AuthenticateUser.accept	A	1
M	14	CheckBalance.accept	A	1
M	18	WithdrawMoney.init	A	1
M	21	WithdrawMoney.accept	A	1
M	25	DepositMoney.init	A	1
M	28	DepositMoney.accept	A	1
M	32	ChangePin.init	A	1
M	36	ChangePin.accept	A	1
M	40	PrintReceipt.accept	A	1
M	44	ATMVisitor.visit_authenticate_user	A	1

M	47	ATMVisitor.visit_check_balance	A	1
M	50	ATMVisitor.visit_withdraw_money	A	1
M	53	ATMVisitor.visit_deposit_money	A	1
M	56	ATMVisitor.visit_change_pin	A	1
M	59	ATMVisitor.visit_print_receipt	A	1
M	63	ATM.init	A	1
M	134	ATMVisitorImpl.init	A	1
M	137	ATMVisitorImpl.visit_authenticate_user	A	1
M	140	ATMVisitorImpl.visit_check_balance	A	1
M	143	ATMVisitorImpl.visit_withdraw_money	A	1
M	146	ATMVisitorImpl.visit_deposit_money	A	1
M	149	ATMVisitorImpl.visit_change_pin	A	1
M	152	ATMVisitorImpl.visit_print_receipt	A	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ radon python library

Type	Line	Class/Method	Grade	Score
F	12	test_authenticate_user	A	3
F	27	test_withdraw_money	A	3
F	43	test_change_pin	A	3
F	21	test_check_balance	A	2
F	37	test_deposit_money	A	2
F	5	atm	A	1
F	9	visitor	A	1
F	53	test_print_receipt	A	1

ค่า Maintainability ของซอฟต์แวร์ในรอบที่ 3 วัดโดยใช้ radon python library คือ A (57.54)

ค่า Maintainability ของโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ radon python library คือ A (58.73)

4.2 Java Programming Language

4.2.1 Chat-GPT 4o

4.2.1.1 Composite Design Pattern

ตารางแสดงค่า Cyclomatic Complexity ของซอสโค้ดในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
Authentication::performOperation	12	15	1	3	0
BalanceInquiry::performOperation	20	23	1	3	0
CashWithdrawal::performOperation	28	31	1	3	0
CashDeposit::performOperation	36	39	1	3	0
ChangePin::performOperation	44	47	1	3	0
PrintReceipt::performOperation	52	55	1	3	0
ATMComposite::addComponent	61	63	1	3	1
ATMComposite::removeComponent	65	67	1	3	1
ATMComposite::performOperation	70	74	2	5	0
CodeComposite1::main	78	92	1	11	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count

ATMSystemTest::setUp	18	31	1	10	0
ATMSystemTest::testAuthenticationOperation	34	38	1	5	0
ATMSystemTest::testBalanceInquiryOperation	41	45	1	5	0
ATMSystemTest::testCashWithdrawalOperation	48	52	1	5	0
ATMSystemTest::testCashDepositOperation	55	59	1	5	0
ATMSystemTest::testChangePinOperation	62	66	1	5	0
ATMSystemTest::testPrintReceiptOperation	69	73	1	5	0
ATMSystemTest::testATMCompositeOperation	76	88	1	10	0
ATMSystemTest::testAddingAndRemovingComponentsInATMComposite	91	114	1	17	0

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
AuthenticateUser::AuthenticateUser	17	20	1	4	2
AuthenticateUser::execute	23	31	3	8	0
BalanceInquiry::BalanceInquiry	38	40	1	3	1
BalanceInquiry::execute	43	46	1	3	0
CashWithdrawal::CashWithdrawal	54	57	1	4	2
CashWithdrawal::execute	60	68	2	8	0
CashWithdrawal::getBalance	70	72	1	3	0
CashDeposit::CashDeposit	80	83	1	4	2
CashDeposit::execute	86	90	1	4	0
CashDeposit::getBalance	92	94	1	3	0
ChangePIN::ChangePIN	103	107	1	5	3
ChangePIN::execute	110	117	2	7	0
PrintReceipt::PrintReceipt	124	126	1	3	1
PrintReceipt::execute	129	133	1	4	0
CompositeATMOperation::addOperation	140	142	1	3	1
CompositeATMOperation::execute	145	149	2	5	0
CodeComposite2::main	154	217	8	53	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
ATMSystemTest::setUp	18	31	1	10	0
ATMSystemTest::testAuthenticationOperation	34	38	1	5	0
ATMSystemTest::testBalanceInquiryOperation	41	45	1	5	0
ATMSystemTest::testCashWithdrawalOperation	48	52	1	5	0
ATMSystemTest::testCashDepositOperation	55	59	1	5	0
ATMSystemTest::testChangePinOperation	62	66	1	5	0
ATMSystemTest::testPrintReceiptOperation	69	73	1	5	0
ATMSystemTest::testATMCompositeOperation	76	88	1	10	0
ATMSystemTest::testAddingAndRemovingComponentsInATMComposite	91	114	1	17	0

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
AuthenticateUser::AuthenticateUser	17	20	1	4	2
AuthenticateUser::execute	23	31	3	8	0
BalanceInquiry::BalanceInquiry	38	40	1	3	1
BalanceInquiry::execute	43	46	1	3	0
CashWithdrawal::CashWithdrawal	54	57	1	4	2
CashWithdrawal::execute	60	68	2	8	0
CashWithdrawal::getBalance	70	72	1	3	0
CashDeposit::CashDeposit	80	83	1	4	2
CashDeposit::execute	86	90	1	4	0
CashDeposit::getBalance	92	94	1	3	0

ChangePIN::ChangePIN	103	107	1	5	3
ChangePIN::execute	110	117	2	7	0
PrintReceipt::PrintReceipt	124	126	1	3	1
PrintReceipt::execute	129	133	1	4	0
CompositeATMOperation::addOperation	140	142	1	3	1
CompositeATMOperation::execute	145	149	2	5	0
CodeComposite2::main	154	217	8	53	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
ATMSystemTest::testAuthenticationSuccessful	8	12	1	4	0
ATMSystemTest::testAuthenticationWithWrongPin	15	21	1	7	0
ATMSystemTest::testBalanceInquiry	24	28	1	4	0
ATMSystemTest::testWithdrawalWithSufficientFunds	31	35	1	4	0
ATMSystemTest::testWithdrawalWithInsufficientFunds	38	42	1	4	0
ATMSystemTest::testDeposit	45	49	1	4	0
ATMSystemTest::testPinChangeWithCorrectOldPinAndMatchingNewPins	52	56	1	4	0
ATMSystemTest::testPinChangeWithIncorrectOldPin	59	63	1	4	0
ATMSystemTest::testPinChangeWithNonMatchingNewPins	66	70	1	4	0
ATMSystemTest::testPrintReceipt	73	77	1	4	0
ATMSystemTest::testATMComposite	80	95	1	10	0

4.2.1.2 Visitor Design Pattern

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
Authentication::Authentication	25	30	1	6	2
Authentication::getCardNumber	32	32	1	1	0
Authentication::getPin	33	33	1	1	0
Authentication::getFailedAttempts	34	34	1	1	0
Authentication::isAuthenticated	35	35	1	1	0
Authentication::incrementFailedAttempts	36	36	1	1	0
Authentication::authenticate	37	37	1	1	0
Authentication::accept	40	42	1	3	1
BalanceInquiry::BalanceInquiry	48	50	1	3	1
BalanceInquiry::getBalance	52	52	1	1	0
BalanceInquiry::accept	55	57	1	3	1
Withdrawal::Withdrawal	64	67	1	4	2
Withdrawal::getAmount	69	69	1	1	0
Withdrawal::getBalance	70	70	1	1	0
Withdrawal::updateBalance	71	71	1	1	1
Withdrawal::accept	74	76	1	3	1
Deposit::Deposit	82	84	1	3	1
Deposit::getDepositAmount	86	86	1	1	0
Deposit::accept	89	91	1	3	1
PinChange::PinChange	99	103	1	5	3
PinChange::getOldPin	105	105	1	1	0
PinChange::getNewPin	106	106	1	1	0
PinChange::getConfirmNewPin	107	107	1	1	0
PinChange::accept	110	112	1	3	1
PrintReceipt::PrintReceipt	118	120	1	3	1
PrintReceipt::getTransactionDetails	122	122	1	1	0
PrintReceipt::accept	125	127	1	3	1
ATMOperationVisitor::visit	134	147	3	13	1
ATMOperationVisitor::visit	150	152	1	3	1
ATMOperationVisitor::visit	155	163	2	9	1

ATMOperationVisitor::visit	166	169	1	3	1
ATMOperationVisitor::visit	172	182	3	11	1
ATMOperationVisitor::visit	185	187	1	3	1
CodeVisitor1::main	192	220	2	17	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
ATMSystemTest::setUp	12	14	1	3	0
ATMSystemTest::testAuthenticationSuccess	17	22	1	6	0
ATMSystemTest::testAuthenticationFailure	25	40	1	12	0
ATMSystemTest::testBalanceInquiry	43	47	1	5	0
ATMSystemTest::testWithdrawalSuccess	50	54	1	5	0
ATMSystemTest::testWithdrawalFailureInsufficientFunds	57	61	1	5	0
ATMSystemTest::testDeposit	64	68	1	5	0
ATMSystemTest::testPinChangeSuccess	71	75	1	4	0
ATMSystemTest::testPinChangeFailureOldPinIncorrect	78	82	1	4	0
ATMSystemTest::testPinChangeFailureNewPinsDoNotMatch	85	89	1	4	0
ATMSystemTest::testPrintReceipt	92	96	1	5	0

ตารางแสดงค่า Cyclomatic Complexity ของซอฟต์แวร์ในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
Authentication::Authentication	24	27	1	4	0
Authentication::authenticate	29	40	3	11	2
Authentication::isAuthenticated	42	44	1	3	0
Authentication::accept	47	49	1	3	1
BalanceInquiry::BalanceInquiry	55	57	1	3	1

BalanceInquiry::getBalance	59	61	1	3	0
BalanceInquiry::accept	64	66	1	3	1
Withdrawal::Withdrawal	72	74	1	3	1
Withdrawal::withdraw	76	83	2	8	1
Withdrawal::accept	86	88	1	3	1
Deposit::Deposit	94	96	1	3	1
Deposit::deposit	98	101	1	4	1
Deposit::accept	104	106	1	3	1
ChangePin::ChangePin	113	115	1	3	1
ChangePin::changePin	117	126	3	10	3
ChangePin::accept	129	131	1	3	1
PrintReceipt::PrintReceipt	137	139	1	3	1
PrintReceipt::print	141	144	1	4	0
PrintReceipt::accept	147	149	1	3	1
ATMACTIONVisitor::visit	155	157	1	3	1
ATMACTIONVisitor::visit	160	162	1	3	1
ATMACTIONVisitor::visit	165	167	1	3	1
ATMACTIONVisitor::visit	170	172	1	3	1
ATMACTIONVisitor::visit	175	177	1	3	1
ATMACTIONVisitor::visit	180	182	1	3	1
CodeVisitor2::main	187	205	1	15	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
ATMSYSTEMTest::setUp	19	27	1	9	0
ATMSYSTEMTest::testAuthenticationSuccess	30	34	1	4	0
ATMSYSTEMTest::testAuthenticationFailure	37	41	1	3	0
ATMSYSTEMTest::testMaxAttemptsExceeded	44	53	2	6	0

ATMSystemTest::testBalanceInquiry	56	60	1	4	0
ATMSystemTest::testWithdrawalSuccess	63	67	1	4	0
ATMSystemTest::testWithdrawalInsufficientFunds	70	74	1	4	0
ATMSystemTest::testDeposit	77	81	1	4	0
ATMSystemTest::testChangePinSuccess	84	89	1	3	0
ATMSystemTest::testChangePinFailure	92	97	1	3	0
ATMSystemTest::testPrintReceipt	100	104	1	3	0

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
User::User	22	25	1	4	2
User::getCard	27	29	1	3	0
User::getAccount	31	33	1	3	0
User::accept	36	40	1	5	1
Card::Card	49	54	1	6	2
Card::getCardNumber	56	58	1	3	0
Card::getPin	60	62	1	3	0
Card::isLocked	64	66	1	3	0
Card::lockCard	68	70	1	3	0
Card::getFailedAttempts	72	74	1	3	0
Card::incrementFailedAttempts	76	78	1	3	0
Card::resetFailedAttempts	80	82	1	3	0
Card::setPin	84	86	1	3	1
Card::accept	89	91	1	3	1
Account::Account	97	99	1	3	1
Account::getBalance	101	103	1	3	0
Account::deposit	105	107	1	3	1
Account::withdraw	109	115	2	7	1
Account::accept	118	120	1	3	1

AuthenticationVisitor::AuthenticationVisitor	131	134	1	4	2
AuthenticationVisitor::isAuthenticated	136	138	1	3	0
AuthenticationVisitor::isCardLocked	140	142	1	3	0
AuthenticationVisitor::visit	145	147	1	2	1
AuthenticationVisitor::visit	150	167	5	18	1
AuthenticationVisitor::visit	170	172	1	2	1
BalanceInquiryVisitor::getBalance	178	180	1	3	0
BalanceInquiryVisitor::visit	183	185	1	2	1
BalanceInquiryVisitor::visit	188	190	1	2	1
BalanceInquiryVisitor::visit	193	195	1	3	1
WithdrawalVisitor::WithdrawalVisitor	203	207	1	5	1
WithdrawalVisitor::isSuccess	209	211	1	3	0
WithdrawalVisitor::isInsufficientFunds	213	215	1	3	0
WithdrawalVisitor::visit	218	220	1	2	1
WithdrawalVisitor::visit	223	225	1	2	1
WithdrawalVisitor::visit	228	235	2	8	1
DepositVisitor::DepositVisitor	241	243	1	3	1
DepositVisitor::visit	246	248	1	2	1
DepositVisitor::visit	251	253	1	2	1
DepositVisitor::visit	256	258	1	3	1
ChangePINVisitor::ChangePINVisitor	268	274	1	7	3
ChangePINVisitor::isSuccess	276	278	1	3	0
ChangePINVisitor::isPinMismatch	280	282	1	3	0
ChangePINVisitor::visit	285	287	1	2	1
ChangePINVisitor::visit	290	299	3	10	1
ChangePINVisitor::visit	302	304	1	2	1
PrintSlipVisitor::setTransactionDetails	315	321	1	7	5
PrintSlipVisitor::getSlipContent	323	325	1	3	0
PrintSlipVisitor::visit	328	333	1	5	1
PrintSlipVisitor::visit	336	339	1	3	1
PrintSlipVisitor::visit	342	352	5	10	1

ATM::ATM	359	361	1	3	1
ATM::authenticate	364	378	3	15	2
ATM::displayBalance	381	391	1	9	0
ATM::withdraw	394	410	4	15	1
ATM::deposit	413	424	1	10	1
ATM::changePIN	427	452	5	23	3
CodeVisitor3::main	457	490	2	18	1

ตารางแสดงค่า Cyclomatic Complexity ของโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity	Lines of Code	Parameter Count
ATMSystemTest::setUp	16	22	1	6	0
ATMSystemTest::testAuthenticationSuccessful	25	27	1	3	0
ATMSystemTest::testAuthenticationFailed	30	35	1	6	0
ATMSystemTest::testCardLockedAfterMultipleFailedAttempts	38	43	1	6	0
ATMSystemTest::testDisplayBalance	46	50	1	5	0
ATMSystemTest::testWithdrawalSuccessful	53	57	1	5	0
ATMSystemTest::testWithdrawalInsufficientFunds	60	64	1	5	0
ATMSystemTest::testDeposit	67	71	1	5	0
ATMSystemTest::testChangePINSuccessful	74	78	1	5	0
ATMSystemTest::testChangePINOldPINIncorrect	81	86	1	6	0
ATMSystemTest::testChangePINNewPINsMismatch	89	94	1	6	0
ATMSystemTest::testSlipContentAfterBalanceInquiry	97	101	1	4	0
ATMSystemTest::testSlipContentAfterWithdrawal	104	108	1	4	0
ATMSystemTest::testSlipContentAfterDeposit	111	115	1	4	0
ATMSystemTest::testSlipContentAfterChangePIN	118	122	1	4	0

4.2.2 Gemini 1.5 Flash

4.2.2.1 Composite Design Pattern

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)
ATMComponent::ATMComponent	8	10	1
BalanceDisplay::BalanceDisplay	21	24	1
BalanceDisplay::execute	27	29	1
BalanceDisplay::getName	32	34	1
CashWithdrawal::CashWithdrawal	40	43	1
CashWithdrawal::execute	46	54	2
CashWithdrawal::getName	57	59	1
CashWithdrawal::getWithdrawalAmount	62	64	1
Deposit::Deposit	70	73	1
Deposit::execute	76	80	1
Deposit::getName	83	85	1
Deposit::getDepositAmount	88	90	1
PINChange::PINChange	96	99	1
PINChange::execute	102	108	2
PINChange::getName	111	113	1
PINChange::getNewPIN	116	118	1
ATMMenu::ATMMenu	125	128	1
ATMMenu::addOperation	130	132	1
ATMMenu::execute	135	147	3
ATMMenu::getName	150	152	1
ATMMenu::getMenuChoice	155	157	1
Account::Account	165	168	1

Account::getBalance	170	172	1
Account::withdraw	174	176	1
Account::deposit	178	180	1
Account::getPIN	182	184	1
Account::setPIN	186	188	1
ATMSystem::main	193	201	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMSystemTest::testBalanceDisplay	8	13	1
ATMSystemTest::testCashWithdrawal	16	21	1
ATMSystemTest::testCashWithdrawalInsufficientFunds	24	29	1
ATMSystemTest::testDeposit	32	37	1
ATMSystemTest::testPINChange	40	45	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
BankAccountComponent::BankAccountComponent	8	10	1
CheckingAccount::CheckingAccount	24	28	1
CheckingAccount::displayBalance	31	33	1
CheckingAccount::withdraw	36	43	2
CheckingAccount::deposit	46	49	1
CheckingAccount::changePIN	52	59	2
CheckingAccount::getBalance	61	63	1

CheckingAccount::getPIN	65	67	1
ATM::ATM	75	78	1
ATM::addAccount	80	82	1
ATM::authenticate	84	98	4
ATM::processTransactions	100	159	11

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMTest::testAuthenticateSuccess	8	15	1
ATMTest::testAuthenticateFailure	18	25	1
ATMTest::testAuthenticateTooManyAttempts	28	37	1
ATMTest::testDisplayBalance	40	48	1
ATMTest::testWithdrawSuccess	51	60	1
ATMTest::testWithdrawInsufficientFunds	63	72	1
ATMTest::testDeposit	75	84	1
ATMTest::testChangePIN	87	96	1
ATMTest::testChangePINIncorrectOldPIN	99	108	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
AuthenticateComponent::process	15	33	3
DisplayBalanceComponent::process	38	40	1
WithdrawCashComponent::process	45	56	2
DepositCashComponent::process	61	68	1
ChangePINComponent::process	73	88	3

PrintReceiptComponent::process	93	114	2
ATMMenuComponent::ATMMenuComponent	121	128	1
ATMMenuComponent::process	131	150	3
ATMContext::ATMContext	162	167	1
ATMContext::authenticateUser	169	171	2
ATMContext::setCurrentUser	173	175	1
ATMContext::getCurrentUser	177	179	1
ATMContext::getBalance	181	183	1
ATMContext::withdraw	185	189	1
ATMContext::deposit	191	195	1
ATMContext::changePIN	197	199	1
ATMContext::getFailedAttempts	201	203	1
ATMContext::incrementFailedAttempts	205	207	1
ATMContext::resetFailedAttempts	209	211	1
ATMContext::getTransactionType	213	215	1
ATMContext::getTransactionAmount	217	219	1
ATMSYSTEM::main	223	235	2

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMSYSTEMTest::testAuthenticateUser_Success	7	10	1
ATMSYSTEMTest::testAuthenticateUser_Failure	13	16	1
ATMSYSTEMTest::testGetBalance	19	22	1
ATMSYSTEMTest::testWithdraw_Success	25	31	1
ATMSYSTEMTest::testWithdraw_InsufficientFunds	34	38	1

ATMSystemTest::testDeposit	41	47	1
ATMSystemTest::testChangePIN_Success	50	54	1
ATMSystemTest::testChangePIN_Failure	57	61	1
ATMSystemTest::testIncrementFailedAttempts	64	68	1
ATMSystemTest::testResetFailedAttempts	71	76	1

4.2.2.2 Visitor Design Pattern

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
BalanceDisplay::execute	12	14	1
CashWithdrawal::CashWithdrawal	20	22	1
CashWithdrawal::execute	25	32	2
Deposit::Deposit	38	40	1
Deposit::execute	43	46	1
PINChange::PINChange	52	54	1
PINChange::execute	57	65	2
Account::Account	74	78	1
Account::getCardNumber	80	82	1
Account::getPIN	84	86	1
Account::getBalance	88	90	1
Account::setPIN	92	94	1
Account::deposit	96	98	1
Account::withdraw	100	102	1
ATM::ATM	117	121	1
ATM::authenticate	124	130	3
ATM::processOperation	133	135	1
ATM::accept	138	144	1

PrintReceiptVisitor::visit	150	152	1
PrintReceiptVisitor::visit	155	157	1
PrintReceiptVisitor::visit	160	162	1
PrintReceiptVisitor::visit	165	167	1
ATMSystem::main	171	190	2

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMTest::authenticate_validCredentials_shouldReturnAccount	8	14	1
ATMTest::authenticate_invalidCardNumber_shouldReturnNull	17	21	1
ATMTest::authenticate_invalidPIN_shouldReturnNull	24	28	1
ATMTest::processOperation_balanceDisplay_shouldDisplayBalance	31	44	1
ATMTest::processOperation_cashWithdrawal_sufficientFunds_shouldWithdrawAnd UpdateBalance	47	63	1
ATMTest::processOperation_cashWithdrawal_insufficientFunds_shouldDisplayAlert	66	82	1
ATMTest::processOperation_deposit_shouldDepositAndUpdateBalance	85	101	1
ATMTest::processOperation_pinChange_validNewPIN_shouldUpdatePIN	104	120	1
ATMTest::processOperation_pinChange_invalidNewPIN_shouldDisplayError	123	139	1
ATMTest::accept_shouldVisitAllOperations	142	152	1
ATMTest::TestATMVisitor::visit	161	163	1
ATMTest::TestATMVisitor::visit	166	168	1
ATMTest::TestATMVisitor::visit	171	173	1
ATMTest::TestATMVisitor::visit	176	178	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATM::ATM	6	9	1
ATM::authenticate	11	16	3
ATM::performTransaction	18	20	1
ATM::printReceipt	22	29	1
Account::Account	37	40	1
Account::getBalance	42	44	1
Account::setBalance	46	48	1
Account::getPIN	50	52	1
Account::setPIN	54	56	1
Account::verifyPIN	58	60	1
Card::Card	68	71	1
Card::getAccount	73	75	1
Transaction::Transaction	85	90	1
Transaction::getDescription	92	94	1
Transaction::getAmount	96	98	1
Transaction::getDate	100	102	1
Transaction::getTime	104	106	1
BalanceDisplay::BalanceDisplay	113	115	1
BalanceDisplay::accept	118	120	1
CashWithdrawal::CashWithdrawal	125	127	1
CashWithdrawal::accept	130	132	1
Deposit::Deposit	137	139	1
Deposit::accept	142	144	1
PINChange::PINChange	151	154	1
PINChange::getNewPIN	156	158	1
PINChange::accept	161	163	1
ATMVisitor::ATMVisitor	178	180	1
ATMVisitor::visit	183	185	1

ATMVisitor::visit	188	195	2
ATMVisitor::visit	198	201	1
ATMVisitor::visit	204	208	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMTest::testAuthenticateSuccess	7	12	1
ATMTest::testAuthenticateFailure	15	20	1
ATMTest::testBalanceDisplay	23	29	1
ATMTest::testCashWithdrawalSuccess	32	38	1
ATMTest::testCashWithdrawalFailure	41	47	1
ATMTest::testDeposit	50	56	1
ATMTest::testPINChange	59	65	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
Account::Account	11	16	1
Account::getAccountNumber	18	20	1
Account::getPin	22	24	1
Account::getBalance	26	28	1

Account::setBalance	30	32	1
Account::getPinAttempts	34	36	1
Account::incrementPinAttempts	38	40	1
Account::resetPinAttempts	42	44	1
Account::changePin	46	48	1
ConcreteATM::ConcreteATM	65	70	1
ConcreteATM::authenticate	73	84	3
ConcreteATM::displayBalance	87	89	1
ConcreteATM::withdraw	92	100	2
ConcreteATM::deposit	103	107	1
ConcreteATM::changePin	110	113	1
ConcreteATM::printReceipt	116	123	1
BalanceVisitor::visit	134	136	1
WithdrawalVisitor::WithdrawalVisitor	142	144	1
WithdrawalVisitor::visit	147	149	1
DepositVisitor::DepositVisitor	155	157	1
DepositVisitor::visit	160	162	1
ChangePinVisitor::ChangePinVisitor	168	170	1
ChangePinVisitor::visit	173	175	1
ATMSYSTEM::ATMSYSTEM	182	184	1
ATMSYSTEM::run	186	272	16
ATMSYSTEM::main	274	277	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMSYSTEMTest::testAuthenticateSuccess	8	13	1

ATMSystemTest::testAuthenticateFailure	16	21	1
ATMSystemTest::testAuthenticateCardRetained	24	31	2
ATMSystemTest::testDisplayBalance	34	39	1
ATMSystemTest::testWithdrawSuccess	42	47	1
ATMSystemTest::testWithdrawInsufficientFunds	50	55	1
ATMSystemTest::testDeposit	58	63	1
ATMSystemTest::testChangePin	66	71	1
ATMSystemTest::testPrintReceipt	74	79	1

4.2.3 Gemini 1.5 Pro

4.2.3.1 Composite Design Pattern

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
CheckBalance::CheckBalance	15	17	1
CheckBalance::performOperation	20	22	1
WithdrawCash::WithdrawCash	29	31	1
WithdrawCash::performOperation	34	42	2
DepositFunds::DepositFunds	49	51	1
DepositFunds::performOperation	54	59	1
ChangePIN::ChangePIN	66	68	1
ChangePIN::performOperation	71	88	3
ATMTransaction::addOperation	95	97	1
ATMTransaction::performOperation	100	104	2
Account::Account	115	121	1
Account::verifyPIN	123	142	4
Account::getBalance	144	146	1

Account::withdraw	148	155	2
Account::deposit	157	160	1
Account::updatePIN	162	164	1
Receipt::Receipt	174	179	1
Receipt::printReceipt	181	192	2
ATM::main	199	227	2

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMTest::setUp	15	17	1
ATMTest::testVerifyPIN_Success	21	24	1
ATMTest::testVerifyPIN_Failure	27	30	1
ATMTest::testVerifyPIN_CardRetained	33	37	1
ATMTest::testGetBalance	40	42	1
ATMTest::testWithdraw_Success	45	48	1
ATMTest::testWithdraw_InsufficientFunds	51	54	1
ATMTest::testDeposit	57	60	1
ATMTest::testUpdatePIN	63	66	1
ATMTest::testATMActions	70	90	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
Authentication::Authentication	17	20	1
Authentication::performOperation	23	39	4
Authentication::authenticatePIN	41	44	1
BalanceDisplay::BalanceDisplay	51	53	1

BalanceDisplay::performOperation	56	58	1
CashWithdrawal::CashWithdrawal	65	67	1
CashWithdrawal::performOperation	70	79	2
Deposit::Deposit	86	88	1
Deposit::performOperation	91	96	1
PINChange::PINChange	103	105	1
PINChange::performOperation	108	113	1
ReceiptPrinting::ReceiptPrinting	120	122	1
ReceiptPrinting::performOperation	125	135	3
Transaction::addOperation	142	144	1
Transaction::performOperation	147	151	2
ATMCard::ATMCard	161	166	1
ATMCard::isCardValid	168	170	1
ATMCard::getPin	172	174	1
ATMCard::changePIN	176	178	1
ATMCard::retainCard	180	182	1
Account::Account	189	191	1
Account::getBalance	193	195	1
Account::withdraw	197	199	1
Account::deposit	201	203	1
ATMSystem::main	207	246	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMSystemTest::testAuthenticationSuccess	11	16	1
ATMSystemTest::testAuthenticationFailure	19	24	1
ATMSystemTest::testAuthenticationCardRetained	27	34	2

ATMSystemTest::testBalanceDisplay	37	42	1
ATMSystemTest::testCashWithdrawalSuccess	45	50	1
ATMSystemTest::testCashWithdrawalInsufficientFunds	53	58	1
ATMSystemTest::testDeposit	61	66	1
ATMSystemTest::testPINChange	69	74	1
ATMSystemTest::testReceiptPrinting	77	82	1
ATMSystemTest::testTransaction	85	122	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
Account::Account	16	20	1
Account::verifyPin	22	24	1
Account::getBalance	26	28	1
Account::deposit	30	32	1
Account::withdraw	34	40	2
Account::changePin	42	44	1
ATMOperation::ATMOperation	52	54	1
ATMOperation::addComponent	56	58	1
ATMOperation::removeComponent	60	62	1
ATMOperation::execute	65	70	2
UserAuthentication::UserAuthentication	78	80	1
UserAuthentication::execute	83	100	3
BalanceInquiry::BalanceInquiry	107	109	1
BalanceInquiry::execute	112	114	1
CashWithdrawal::CashWithdrawal	121	123	1
CashWithdrawal::execute	126	136	2
Deposit::Deposit	142	144	1

Deposit::execute	147	154	1
PINChange::PINChange	161	163	1
PINChange::execute	166	186	3
ReceiptPrinting::execute	192	195	1
ATM::main	200	267	8

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMSystemTest::setUp	16	23	1
ATMSystemTest::testAuthenticationSuccess	27	32	1
ATMSystemTest::testAuthenticationFailure	35	40	1
ATMSystemTest::testAuthenticationCardRetained	43	53	2
ATMSystemTest::testBalanceInquiry	57	61	1
ATMSystemTest::testSuccessfulWithdrawal	65	70	1
ATMSystemTest::testInsufficientFundsWithdrawal	73	78	1
ATMSystemTest::testDeposit	82	86	1
ATMSystemTest::testSuccessfulPINChange	90	96	1
ATMSystemTest::testPINChangeMismatch	99	104	1
ATMSystemTest::testPINChangeIncorrectOldPIN	107	111	1

4.2.3.2 Visitor Design Pattern

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)
ATM::ATM	12	17	1
ATM::main	19	22	1
ATM::start	24	36	2
ATM::authenticate	38	60	4
ATM::showMenu	62	95	7
Account::Account	104	108	1
Account::validatePin	110	112	1
Account::getCardNumber	115	117	1
Account::getBalance	119	121	1
Account::setBalance	123	125	1
Account::setPin	127	129	1
BalanceInquiry::accept	140	142	1
CashWithdrawal::accept	147	149	1
CashDeposit::accept	154	156	1
PinChange::accept	161	163	1
ATMVisitorImpl::ATMVisitorImpl	179	182	1
ATMVisitorImpl::visit	185	187	1
ATMVisitorImpl::visit	190	200	3
ATMVisitorImpl::visit	203	213	2
ATMVisitorImpl::visit	216	233	3

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 1 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMTest::setUp	15	18	1
ATMTest::testAuthenticate_Success	22	28	1
ATMTest::testAuthenticate_InvalidCardNumber	31	37	1
ATMTest::testAuthenticate_IncorrectPin_MaxAttempts	40	46	1
ATMTest::testBalanceInquiry	50	57	1
ATMTest::testCashWithdrawal_Success	61	69	1
ATMTest::testCashWithdrawal_InsufficientFunds	72	80	1
ATMTest::testCashDeposit	84	92	1
ATMTest::testPinChange_Success	96	104	1
ATMTest::testPinChange_IncorrectOldPin	107	115	1
ATMTest::testPinChange_NewPinsMismatch	118	126	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATM::ATM	11	17	1
ATM::main	19	22	1
ATM::start	24	33	3
ATM::authenticateUser	35	48	3
ATM::showMenu	50	82	6
Account::Account	91	95	1
Account::validatePin	97	99	1
Account::getBalance	101	103	1
Account::setBalance	105	107	1
Account::setPin	109	111	1
Account::accept	114	116	1
ATMOperationVisitor::ATMOperationVisitor	129	132	1

ATMOperationVisitor::visit	135	162	5
ATMOperationVisitor::withdraw	164	174	3
ATMOperationVisitor::deposit	176	186	2
ATMOperationVisitor::changePin	188	206	3

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดทดสอบในรอบที่ 2 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMTest::testAuthenticationSuccess	9	20	1
ATMTest::testAuthenticationFailure	23	34	1
ATMTest::testCheckBalance	37	48	1
ATMTest::testWithdrawSuccess	51	61	1
ATMTest::testWithdrawInsufficientFunds	65	74	1
ATMTest::testDeposit	78	88	1
ATMTest::testChangePinSuccess	92	102	1
ATMTest::testChangePinMismatch	105	114	1

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สโค้ดในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATM::ATM	12	16	1
ATM::main	18	21	1
ATM::start	23	44	3
ATM::authenticateUser	46	60	4

ATM::performTransactions	62	90	7
ATM::printMenu	92	100	1
Account::Account	109	113	1
Account::accept	115	117	1
Account::getCardNumber	120	122	1
Account::getPin	124	126	1
Account::setPin	128	130	1
Account::getBalance	132	134	1
Account::setBalance	136	138	1
TransactionVisitor::TransactionVisitor	146	149	1
TransactionVisitor::visit	151	191	8

ตารางแสดงค่า Cyclomatic Complexity ของชอร์สทดสอบโค้ดในรอบที่ 3 วัดโดยใช้ Codalyze - Code Complexity Report Generator

Function Name	Start Line	End Line	Cyclomatic Complexity
ATMTest::setUp	13	15	1
ATMTest::testAuthenticateUser_Success	18	20	1
ATMTest::testAuthenticateUser_Failure	23	25	1
ATMTest::testAuthenticateUser_CardNotFound	28	30	1
ATMTest::testBalanceInquiry	33	41	1
ATMTest::testCashWithdrawal_Success	44	53	1
ATMTest::testCashWithdrawal_InsufficientFunds	56	65	1
ATMTest::testDeposit	68	77	1
ATMTest::testPinChange_Success	80	89	1
ATMTest::testPinChange_IncorrectOldPin	92	101	1
ATMTest::testPinChange_MismatchedNewPins	104	113	1

4.2.4 GitHub Copilot

4.2.4.1 Composite Design Pattern

Round1

ATM.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATM::ATM	10	16	1	7	3
ATM::authenticate	18	34	4	16	2
ATM::checkBalance	36	44	2	9	0
ATM::withdraw	46	60	3	15	1
ATM::deposit	62	69	2	8	1
ATM::changePin	71	85	3	15	2
ATM::printReceipt	87	94	2	7	0
ATMCompositeOperation.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMCompositeOperation::addOperation	9	11	1	3	1
ATMCompositeOperation::execute	14	18	2	5	0
ATMOperation.java					
AuthenticateUser.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
AuthenticateUser::AuthenticateUser	8	12	1	5	3
AuthenticateUser::execute	15	17	1	3	0
ChangePin.java					

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ChangePin::ChangePin	8	12	1	5	3
ChangePin::execute	15	17	1	3	0
CheckBalance.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
CheckBalance::CheckBalance	6	8	1	3	1
CheckBalance::execute	11	13	1	3	0
DepositMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
DepositMoney::DepositMoney	7	10	1	4	2
DepositMoney::execute	13	15	1	3	0
Main.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Main::main	4	16	1	11	1
PrintReceipt.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
PrintReceipt::PrintReceipt	6	8	1	3	1
PrintReceipt::execute	11	13	1	3	0
WithdrawMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
WithdrawMoney::WithdrawMoney	7	10	1	4	2

WithdrawMoney::execute	13	15	1	3	0
------------------------	----	----	---	---	---

Round2

ATM.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATM::ATM	9	14	1	6	3
ATM::authenticate	16	25	3	10	2
ATM::checkBalance	27	35	2	9	0
ATM::withdraw	37	51	3	15	1
ATM::deposit	53	60	2	8	1
ATM::changePin	62	76	3	15	2
ATM::printReceipt	78	85	2	7	0
ATMCompositeOperation.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMCompositeOperation::addOperation	8	10	1	3	1
ATMCompositeOperation::execute	13	17	2	5	0
ATMOperation.java					
AuthenticateUser.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
AuthenticateUser::AuthenticateUser	8	12	1	5	3
AuthenticateUser::execute	15	17	1	3	0

ChangePin.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ChangePin::ChangePin	8	12	1	5	3
ChangePin::execute	15	17	1	3	0
CheckBalance.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
CheckBalance::CheckBalance	6	8	1	3	1
CheckBalance::execute	11	13	1	3	0
DepositMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
DepositMoney::DepositMoney	7	10	1	4	2
DepositMoney::execute	13	15	1	3	0
Main.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Main::main	4	16	1	11	1
PrintReceipt.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
PrintReceipt::PrintReceipt	6	8	1	3	1
PrintReceipt::execute	11	13	1	3	0
WithdrawMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
WithdrawMoney::WithdrawMoney	7	10	1	4	2

wMoney					
WithdrawMoney::execute	13	15	1	3	0

Round3

ATM.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATM::ATM	10	16	1	7	3
ATM::authenticate	18	34	4	16	2
ATM::checkBalance	36	44	2	9	0
ATM::withdraw	46	60	3	15	1
ATM::deposit	62	69	2	8	1
ATM::changePin	71	85	3	15	2
ATM::printReceipt	87	94	2	7	0
ATMCompositeOperation.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMCompositeOperation::addOperation	9	11	1	3	1
ATMCompositeOperation::execute	14	18	2	5	0
ATMOperation.java					
AuthenticateUser.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
AuthenticateUser::AuthenticateUser	8	12	1	5	3
AuthenticateUser::execute	15	17	1	3	0

e					
ChangePin.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ChangePin::ChangePin	8	12	1	5	3
ChangePin::execute	15	17	1	3	0
CheckBalance.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
CheckBalance::CheckBalanc	6	8	1	3	1
CheckBalance::execute	11	13	1	3	0
DepositMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
DepositMoney::DepositMone	7	10	1	4	2
DepositMoney::execute	13	15	1	3	0
Main.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Main::main	4	16	1	11	1
PrintReceipt.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
PrintReceipt::PrintReceipt	6	8	1	3	1
PrintReceipt::execute	11	13	1	3	0
WithdrawMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)

WithdrawMoney::WithdrawMoney	7	10	1	4	2
WithdrawMoney::execute	13	15	1	3	0

TEST

Round1\ATMTest.java			Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Function Name	Start Line	End Line			
ATMTest::setUp	13	15	1	3	0
ATMTest::testAuthenticate	18	21	1	4	0
ATMTest::testCheckBalance	24	27	1	4	0
ATMTest::testWithdraw	30	35	1	6	0
ATMTest::testDeposit	38	42	1	5	0
ATMTest::testChangePin	45	49	1	5	0
ATMTest::testPrintReceipt	52	58	1	7	0
Round2\ATMTest.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMTest::setUp	13	15	1	3	0
ATMTest::testAuthenticate	18	21	1	4	0
ATMTest::testCheckBalance	24	27	1	4	0
ATMTest::testWithdraw	30	35	1	6	0
ATMTest::testDeposit	38	42	1	5	0
ATMTest::testChangePin	45	49	1	5	0
ATMTest::testPrintReceipt	52	58	1	7	0

Round3ATMTest.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMTest::setUp	13	15	1	3	0
ATMTest::testAuthenticat e	18	25	1	8	0
ATMTest::testCheckBalan ce	28	33	1	6	0
ATMTest::testWithdraw	36	43	1	8	0
ATMTest::testDeposit	46	52	1	7	0
ATMTest::testChangePin	55	61	1	7	0
ATMTest::testPrintReceipt	64	70	1	7	0

4.2.4.2 Visitor Design Pattern

Round1

ATM.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATM::ATM	10	16	1	7	3
ATM::authenticate	18	34	4	16	2
ATM::checkBalance	36	44	2	9	0
ATM::withdraw	46	60	3	15	1
ATM::deposit	62	69	2	8	1
ATM::changePin	71	85	3	15	2
ATM::printReceipt	87	94	2	7	0
ATMOperation.java					
ATMVisitor.java					

ATMVisitorImpl.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMVisitorImpl::ATMVisito rImpl	6	8	1	3	1
ATMVisitorImpl::visit	11	13	1	3	1
ATMVisitorImpl::visit	16	18	1	3	1
ATMVisitorImpl::visit	21	23	1	3	1
ATMVisitorImpl::visit	26	28	1	3	1
ATMVisitorImpl::visit	31	33	1	3	1
ATMVisitorImpl::visit	36	38	1	3	1
AuthenticateUser.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
AuthenticateUser::Authen ticateUser	7	10	1	4	2
AuthenticateUser::getCar dNumber	12	14	1	3	0
AuthenticateUser::getPin	16	18	1	3	0
AuthenticateUser::accept	21	23	1	3	1
ChangePin.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ChangePin::ChangePin	7	10	1	4	2
ChangePin::getOldPin	12	14	1	3	0
ChangePin::getNewPin	16	18	1	3	0
ChangePin::accept	21	23	1	3	1
CheckBalance.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)

CheckBalance::accept	5	7	1	3	1
DepositMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
DepositMoney::DepositM oney	6	8	1	3	1
DepositMoney::getAmou nt	10	12	1	3	0
DepositMoney::accept	15	17	1	3	1
Main.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Main::main	4	20	2	15	1
PrintReceipt.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
PrintReceipt::accept	5	7	1	3	1
WithdrawMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
WithdrawMoney::Withdra wMoney	6	8	1	3	1
WithdrawMoney::getAmo unt	10	12	1	3	0
WithdrawMoney::accept	15	17	1	3	1

Round2

ATM.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)

ATM::ATM	10	16	1	7	3
ATM::authenticate	18	34	4	16	2
ATM::checkBalance	36	44	2	9	0
ATM::withdraw	46	60	3	15	1
ATM::deposit	62	69	2	8	1
ATM::changePin	71	85	3	15	2
ATM::printReceipt	87	94	2	7	0
ATMOperation.java					
ATMVisitor.java					
ATMVisitorImpl.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMVisitorImpl::ATMVisito rImpl	6	8	1	3	1
ATMVisitorImpl::visit	11	13	1	3	1
ATMVisitorImpl::visit	16	18	1	3	1
ATMVisitorImpl::visit	21	23	1	3	1
ATMVisitorImpl::visit	26	28	1	3	1
ATMVisitorImpl::visit	31	33	1	3	1
ATMVisitorImpl::visit	36	38	1	3	1
AuthenticateUser.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
AuthenticateUser::Authen ticateUser	7	10	1	4	2
AuthenticateUser::getCar dNumber	12	14	1	3	0
AuthenticateUser::getPin	16	18	1	3	0
AuthenticateUser::accept	21	23	1	3	1

ChangePin.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ChangePin::ChangePin	7	10	1	4	2
ChangePin::getOldPin	12	14	1	3	0
ChangePin::getNewPin	16	18	1	3	0
ChangePin::accept	21	23	1	3	1
CheckBalance.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
CheckBalance::accept	5	7	1	3	1
DepositMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
DepositMoney::DepositM oney	6	8	1	3	1
DepositMoney::getAmou nt	10	12	1	3	0
DepositMoney::accept	15	17	1	3	1
Main.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Main::main	4	20	2	15	1
PrintReceipt.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
PrintReceipt::accept	5	7	1	3	1
WithdrawMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)

WithdrawMoney::WithdrawMoney	6	8	1	3	1
WithdrawMoney::getAmount	10	12	1	3	0
WithdrawMoney::accept	15	17	1	3	1

Round3

ATM.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATM::ATM	10	16	1	7	3
ATM::authenticate	18	34	4	16	2
ATM::checkBalance	36	44	2	9	0
ATM::withdraw	46	60	3	15	1
ATM::deposit	62	69	2	8	1
ATM::changePin	71	85	3	15	2
ATM::printReceipt	87	94	2	7	0
ATMOperation.java					
ATMVisitor.java					
ATMVisitorImpl.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMVisitorImpl::ATMVisitorImpl	6	8	1	3	1
ATMVisitorImpl::visit	11	13	1	3	1
ATMVisitorImpl::visit	16	18	1	3	1
ATMVisitorImpl::visit	21	23	1	3	1
ATMVisitorImpl::visit	26	28	1	3	1
ATMVisitorImpl::visit	31	33	1	3	1

ATMVisitorImpl::visit	36	38	1	3	1
AuthenticateUser.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
AuthenticateUser::Authen ticateUser	7	10	1	4	2
AuthenticateUser::getCar dNumber	12	14	1	3	0
AuthenticateUser::getPin	16	18	1	3	0
AuthenticateUser::accept	21	23	1	3	1
ChangePin.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ChangePin::ChangePin	7	10	1	4	2
ChangePin::getOldPin	12	14	1	3	0
ChangePin::getNewPin	16	18	1	3	0
ChangePin::accept	21	23	1	3	1
CheckBalance.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
CheckBalance::accept	5	7	1	3	1
DepositMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
DepositMoney::DepositM oney	6	8	1	3	1
DepositMoney::getAmou nt	10	12	1	3	0
DepositMoney::accept	15	17	1	3	1
Main.java					

Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
Main::main	4	20	2	15	1
PrintReceipt.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
PrintReceipt::accept	5	7	1	3	1
WithdrawMoney.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
WithdrawMoney::WithdrawMoney	6	8	1	3	1
WithdrawMoney::getAmount	10	12	1	3	0
WithdrawMoney::accept	15	17	1	3	1

TEST

Round1\ATMTest.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMTest::setUp	13	16	1	4	0
ATMTest::testAuthenticateUser	19	27	1	8	0
ATMTest::testCheckBalance	30	35	1	6	0
ATMTest::testWithdrawMoney	38	47	1	9	0
ATMTest::testDepositMoney	50	55	1	6	0
ATMTest::testChangePin	58	67	1	9	0

ATMTest::testPrintReceipt	70	75	1	5	0
Round2\ATMTest.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMTest::setUp	13	16	1	4	0
ATMTest::testAuthenticateUser	19	27	1	8	0
ATMTest::testCheckBalance	30	35	1	6	0
ATMTest::testWithdrawMoney	38	47	1	9	0
ATMTest::testDepositMoney	50	55	1	6	0
ATMTest::testChangePin	58	67	1	9	0
ATMTest::testPrintReceipt	70	75	1	5	0
Round3\ATMTest.java					
Function Name	Start Line	End Line	Cyclomatic Complexity (Threshold: 10)	Lines of Code (Threshold: 50)	Parameter Count (Threshold: 4)
ATMTest::setUp	13	16	1	4	0
ATMTest::testAuthenticateUser	19	27	1	8	0
ATMTest::testCheckBalance	30	35	1	6	0
ATMTest::testWithdrawMoney	38	47	1	9	0
ATMTest::testDepositMoney	50	55	1	6	0
ATMTest::testChangePin	58	67	1	9	0
ATMTest::testPrintReceipt	70	75	1	5	0

