Architectural Design, UML diagrams & Implementation Plan

MONEY MOVEMENT

Group: ISTEP

นายพัสกร จุลพล 54010907

นายสุรพงศ์ เท่าเทียมตน 54011423

Problem Analysis

Abstraction

1. **ผู้ใช้งานระบบ**

   ประกอบไปด้วย Username Password และข้อมูลส่วนตัวอื่นๆของผู้ใช้เพื่อใช้ในการระบบและเป็นข้อมูลในการวิเคราะห์

2. **รายรับรายจ่ายของผู้ใช้**

   ประกอบไปด้วย เจ้าของ จำนวนเงิน หมวดหมู่ วันที่ ของรายรับรายจ่ายนั้น มีไว้เพื่อใช้ในการเป็นข้อมูลในการแสดงผลวิเคราะห์ เปรียบเทียบ ตามความต้องการของผู้ใช้

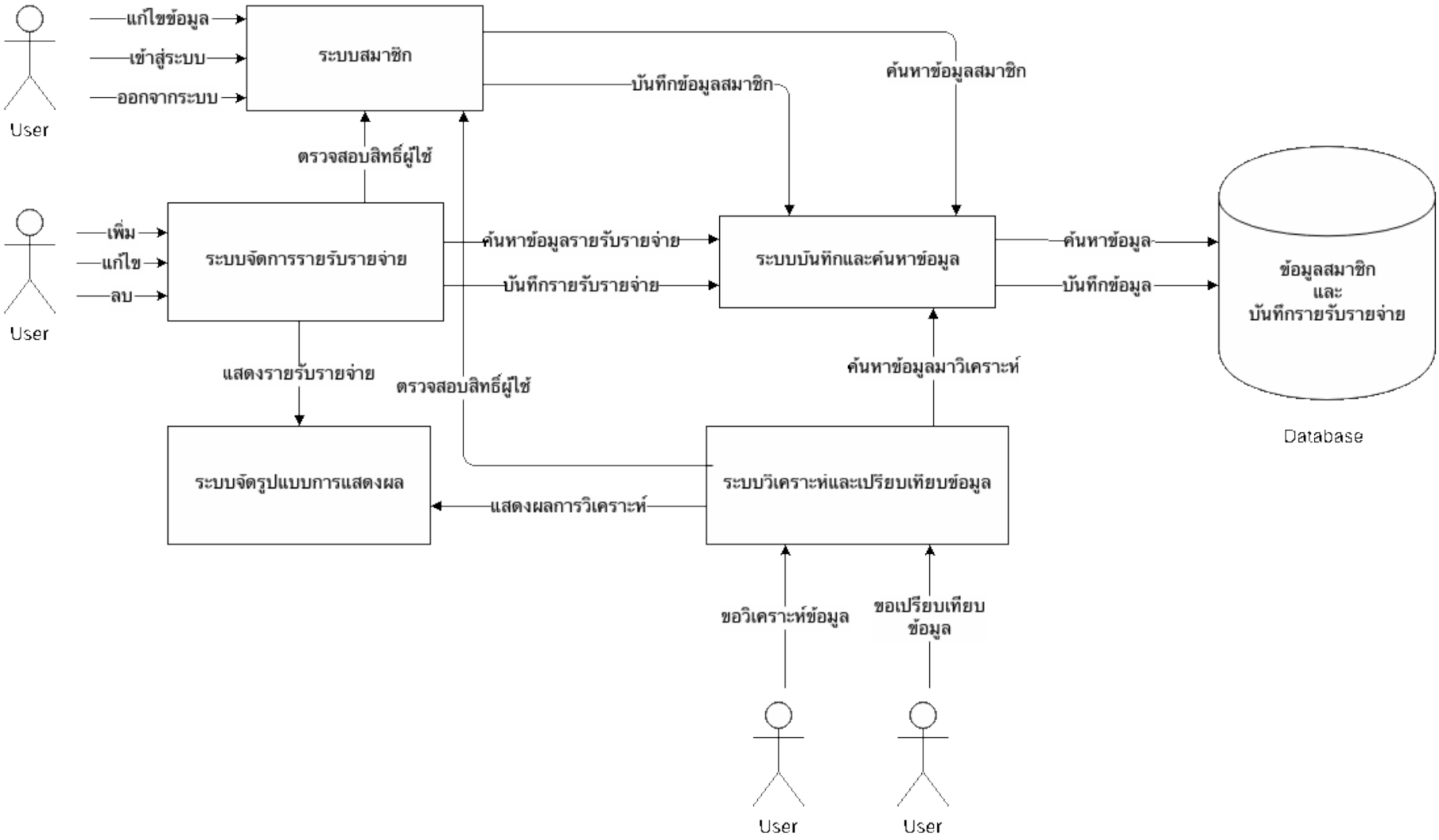
3. **วิเคราะห์ข้อมูลรายรับรายจ่าย**

   ประกอบด้วยส่วนของการวิเคราะห์ข้อมูล เปรียบเทียบข้อมูล โดยจะนำเอาข้อมูลของการบันทึกรายรับรายจ่ายของผู้ใช้มาทำการวิเคราะห์ เปรียบเทียบและนำผลลัพท์ไปแสดงผลให้กับผู้ใช้

4. **รูปแบบการแสดงผล**

   ในการแสดงผลให้กับผู้ใช้นั้นจะแสดงผลในรูปแบบของกราฟความสัมพันธ์ เช่น กราฟเส้น กราฟวงกลม หรืออาจจะแสดงในรูปแบบของลำดับของการจดบันทึกรายรับรายจ่าย (timeline)

Application Architecture



ระบบสมาชิก

มีหน้าที่ทำเกี่ยวกับการสมัครสมาชิก เข้าสู่ระบบ ออกจากระบบ การทำการตรวจสอบว่าผู้ใช้ทำการเข้าสู่ระบบแล้วหรือยัง เมื่อเรียกใช้ส่วนที่จำเป็นต้องผ่านการเข้าสู่ระบบจึงจะสามารถทำได้

ระบบจัดการรายรับรายจ่าย

มีหน้าที่ทำการจัดการเกี่ยวกับ เพิ่มข้อมูลรายรับรายจ่าย แก้ไขข้อมูลรายรับรายจ่าย ลบข้อมูลรายรับจ่าย

ระบบวิเคราะห์และเปรียบเทียบข้อมูล

มีหน้าที่จัดการการวิเคราะห์ข้อมูลรายรับรายจ่ายของผู้ใช้และเปรียบเทียบข้อมูลรายรับรายจ่ายกับค่าเฉลี่ยของผู้ใช้คนอื่นๆ

ระบบบันทึกและค้นหาข้อมูล

เป็นส่วนที่มีหน้าที่ทำการไปค้นหาข้อมูลและบันทึกข้อมูลจากที่ที่เก็บข้อมูลไว้

ระบบจัดรูปแบบการแสดงผล

เป็นส่วนที่มีหน้าที่จัดการข้อมูลให้ตรงตามเงื่อนไขกับการนำไปแสดงผล

Subsystem / Component

1. ระบบสมาชิก

1.1 ส่วนดูแลการเข้าสู่ระบบมีหน้าที่จัดการเกี่ยวกับการเข้าสู่ระบบของผู้ใช้

1.2 ส่วนดูแลการออกจากระบบมีหน้าที่จัดการเกี่ยวกับออกจากระบบ

1.3 ส่วนดูแลการแก้ไขข้อมูลผู้ใช้มีหน้าที่ทำการแก้ไขข้อมูลของผู้ใช้ตามความต้องการของผู้ใช้ เช่น password ชื่อ นามสกุล อาชีพ อายุ

1.4 ส่วนดูแลการสมัครสมาชิกมีหน้าที่ดูแลการสมัครสมาชิกของผู้ใช้

1.5 ส่วนตรวจสอบการอยู่ในระบบและสิทธิ์ของผู้ใช้ มีหน้าที่ตรวจสอบว่าผู้ใช้ยังอยู่ในระบบหรือไม่เมื่อทำการขอให้งานความสามารถต่างๆระบบ

2. ระบบจัดการรายรับรายจ่าย

2.1 ส่วนเพิ่มข้อมูลรายรับรายจ่าย มีหน้าที่เพิ่มข้อมูลรายรับรายจ่ายเข้าไปสู่ระบบ

2.2 ส่วนแก้ไขข้อมูลรายรับรายจ่าย มีหน้าที่แก้ไขข้อมูลรายรับรายจ่ายที่มีอยู่ในระบบตามความต้องการของผู้ใช้

2.3 ส่วนลบข้อมูลรายรับรายจ่าย มีหน้าที่ลบข้อมูลรายรับรายจ่ายที่มีอยู่ในระบบของผู้ใช้

3. ระบบวิเคราะห์และเปรียบเทียบข้อมูล

3.1 ส่วนวิเคราะห์ข้อมูลรายรับรายจ่ายของลูกค้า มีหน้าที่วิเคราะห์ข้อมูลรายรับรายจ่ายของลูกค้าว่ามีแนวโน้มเป็นอย่างไรตามขั้นตอนวิธีการ (Algorithm) ต่างๆ

3.2 ส่วนเปรียบเทียบข้อมูลรายรับรายจ่ายของลูกค้า มีหน้าที่เปรียบเทียบข้อมูลรายรับรายจ่ายของลูกค้ากับค่าเฉลี่ยรายรับรายจ่ายของคนอื่นๆ ในระบบตามขั้นตอนวิธีการ (Algorithm) ต่างๆ

4. ระบบจัดรูปแบบการแสดงผล

4.1 ส่วนแสดงผลในส่วนของการวิเคราะห์ข้อมูลรายรับรายจ่าย มีหน้าที่ทำการแสดงผลลัพธ์ของการวิเคราะห์ออกมาในรูปแบบต่างๆให้ผู้ใช้เห็น

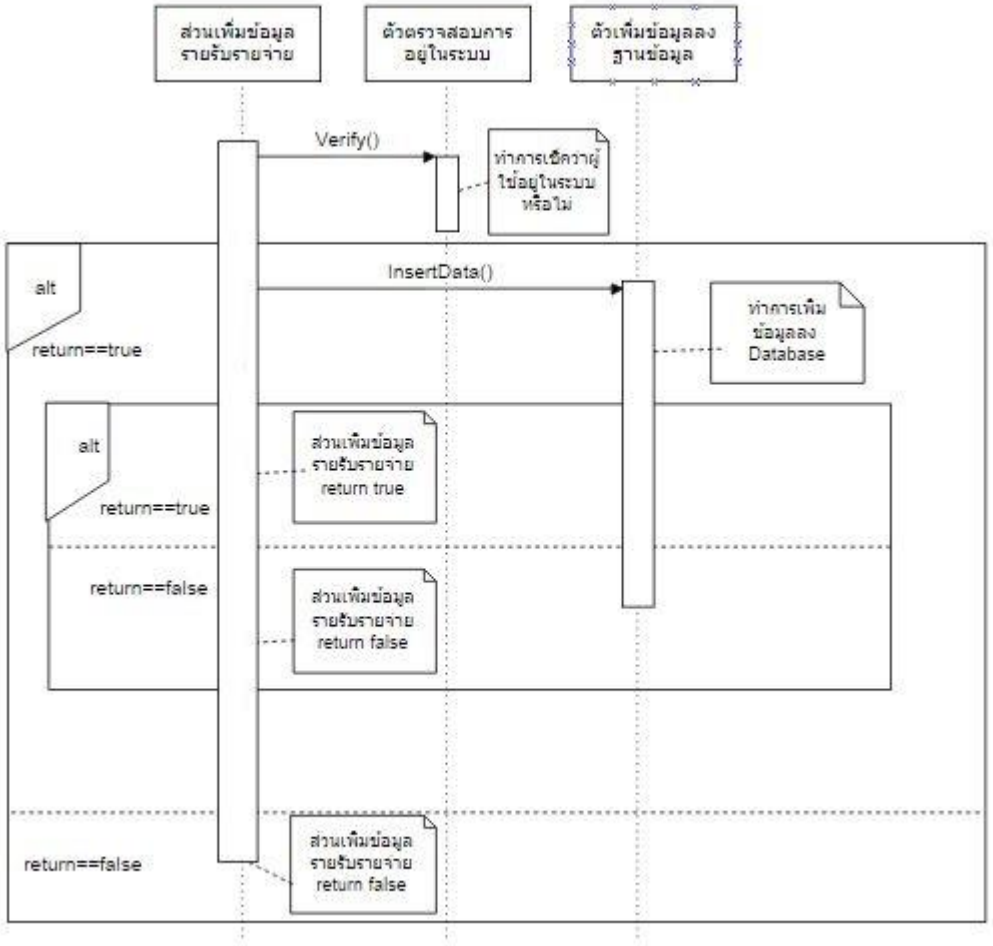
4.2 ส่วนแสดงผลในส่วนของการเปรียบเทียบข้อมูลรายรับรายจ่าย มีหน้าที่ทำการแสดงผลลัพธ์การวิเคราะห์ออกมาในรูปแบบต่างๆให้ผู้ใช้เห็น

4.3 ส่วนแสดงผลข้อมูลรายรับรายจ่าย มีหน้าที่ทำการแสดงข้อมูลรายรับรายจ่ายออกมาในรูปแบบต่างๆเช่น กราฟเส้น กราฟวงกลม ตาราง เป็นต้น

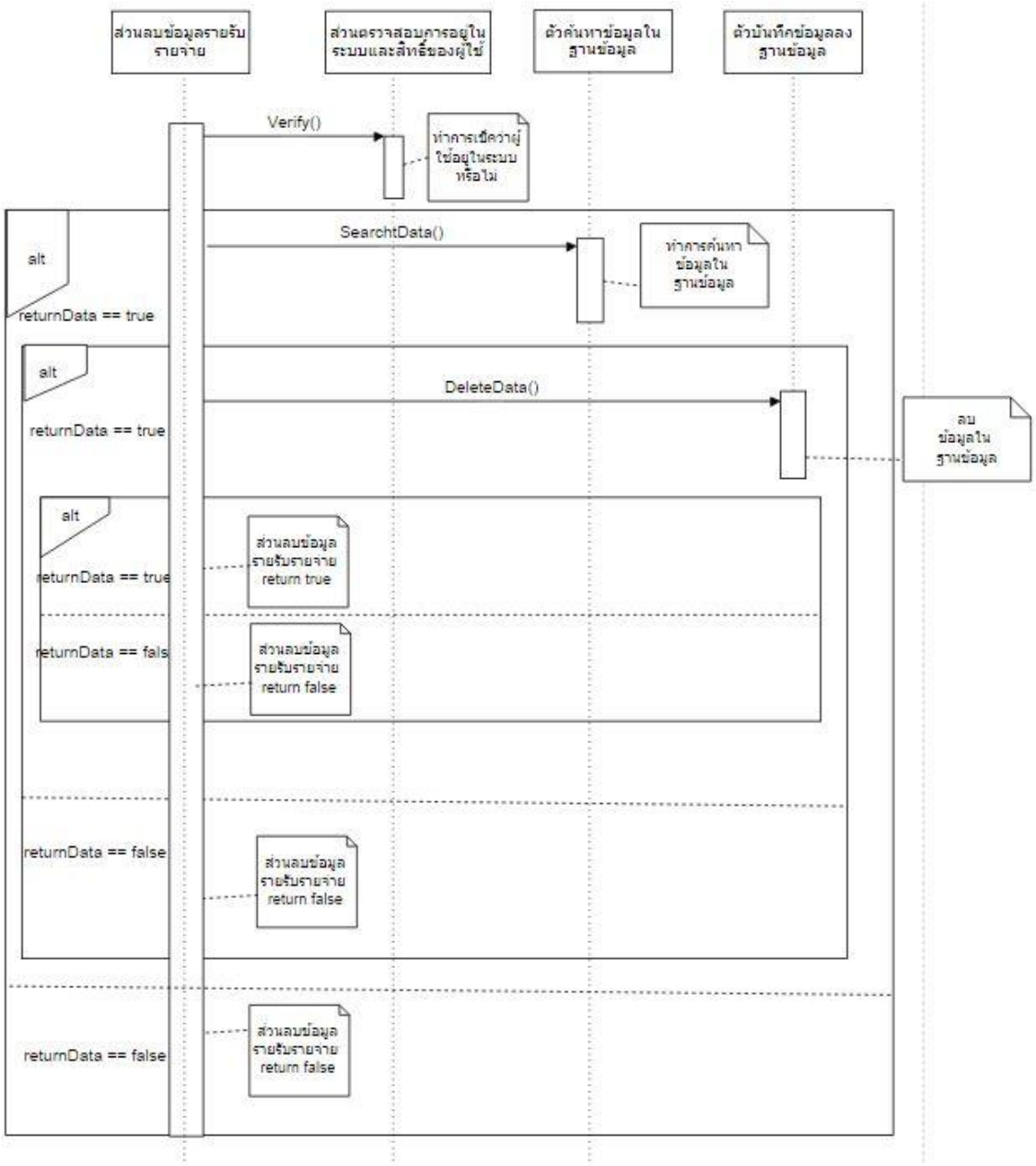
5. ระบบบันทึกและค้นหาข้อมูล

มีหน้าที่เป็นตัวกลางในการติดต่อกับแหล่งข้อมูลโดยมีหน้าที่หลักๆ คือ 1.ค้นหาข้อมูล 2.แก้ไขข้อมูล 3. เพิ่มข้อมูล 4. ลบข้อมูล

การเพิ่มรายรับรายจ่าย



การลบรายรับรายจ่าย



Domain classes

Class ที่เกี่ยวข้องกับระบบสมาชิกจะประกอบไปด้วย 5 Class

1. LoginManager จะทำการ implement interface LoginAble เพื่อนำไปใช้เป็นตัวที่ทำการเข้าสู่ระบบ

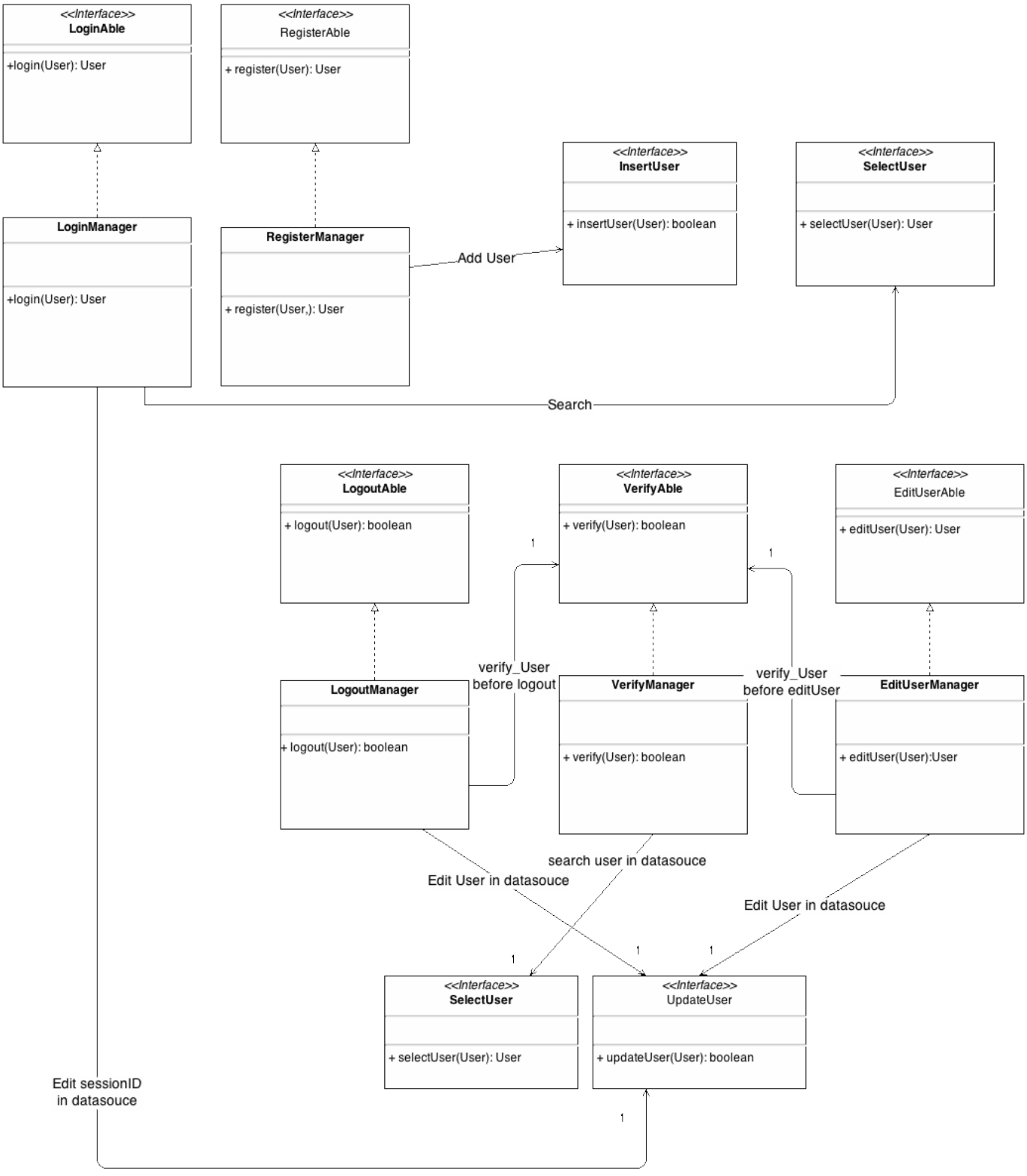
2. LogoutManager จะทำการ implement interface LogoutAble เพื่อนำไปใช้ในการออกจากระบบ โดย class เรียกใช้ class ที่ implement interface VerifyAble เพื่อตรวจสอบว่าผู้ใช้ที่จะสั่ง Logout นั้นมีสิทธิ์และอยู่ในระบบหรือไม่

3. VerifyManager จะทำการ implement interface VerifyAble เพื่อนำไปใช้ในการตรวจสอบผู้ใช้ว่าอยู่ในระบบหรือไม่มีสิทธิ์ที่จะทำการใดๆที่กำลังจะทำหรือไม่

4. EditUserManager จะทำการ implement interface EditUserManager เพื่อนำไปใช้ในการในการแก้ไขข้อมูลและบัญชีของผู้ใช้ โดย class นี้จะทำการเรียกใช้ interface VerifyAble เพื่อตรวจสอบสิทธิ์และสถานะการอยู่ในระบบ

5. RegisterManager จะทำการ implement interface RegisterAble เพื่อนำไปใช้ในการสมัครสมาชิกของผู้ใช้

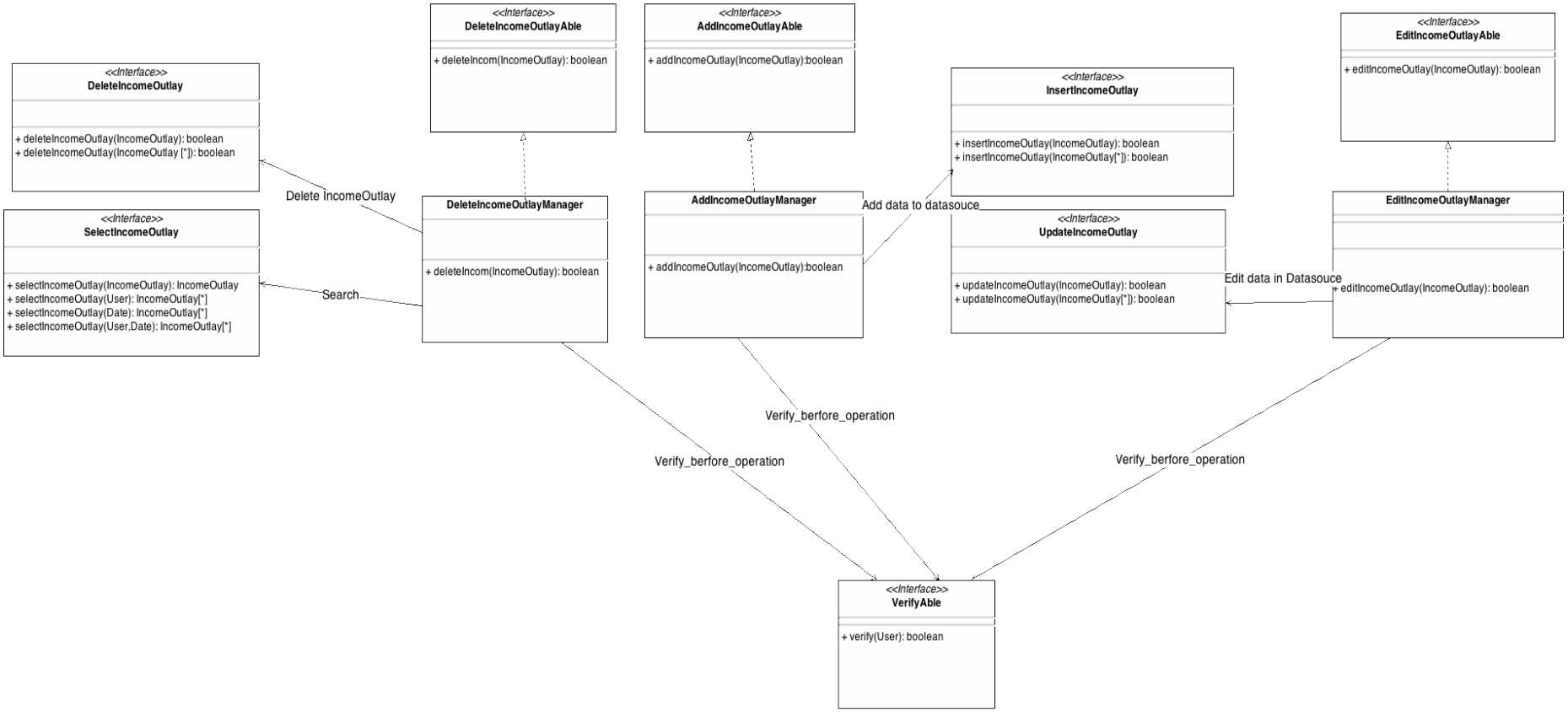
ซึ่งทั้ง 5 class นี้จะเรียกใช้ class ที่ implement interface ที่เกี่ยวกับการติดต่อกับแหล่งข้อมูลเพื่อทำการค้นหาจากแหล่งข้อมูลโดยที่ทำการออกแบบให้ใช้เรียกใช้ผ่าน class ที่ implement interface แทนการเขียนติดต่อกับแหล่งข้อมูลใน class เลย เพราะต้องการความยืดหยุ่นเวลาการแก้ไขปรับเปลี่ยนการติดต่อกับแหล่งข้อมูล จะได้แก้ไขที่คลาสที่เรียกใช้หรือทำการเรียกใช้ class ใหม่


<<Interface>>
LoginAble
+login(User): User
<<Interface>>
RegisterAble
+ register(User): User
<<Interface>>
InsertUser
+ insertUser(User): boolean
<<Interface>>
SelectUser
+ selectUser(User): User
LoginManager
+login(User): User
RegisterManager
+ register(User,): User
Add User
Search
<<Interface>>
LogoutAble
+ logout(User): boolean
<<Interface>>
VerifyAble
+ verify(User): boolean
<<Interface>>
EditUserAble
+ editUser(User): User
1
1
verify_User
before logout
verify_User
before editUser
LogoutManager
+ logout(User): boolean
VerifyManager
+ verify(User): boolean
EditUserManager
+ editUser(User):User
search user in datasouce
Edit User in datasouce
Edit User in datasouce
1
1
1
<<Interface>>
SelectUser
+ selectUser(User): User
<<Interface>>
UpdateUser
+ updateUser(User): boolean
Edit sessionID
in datasouce
1

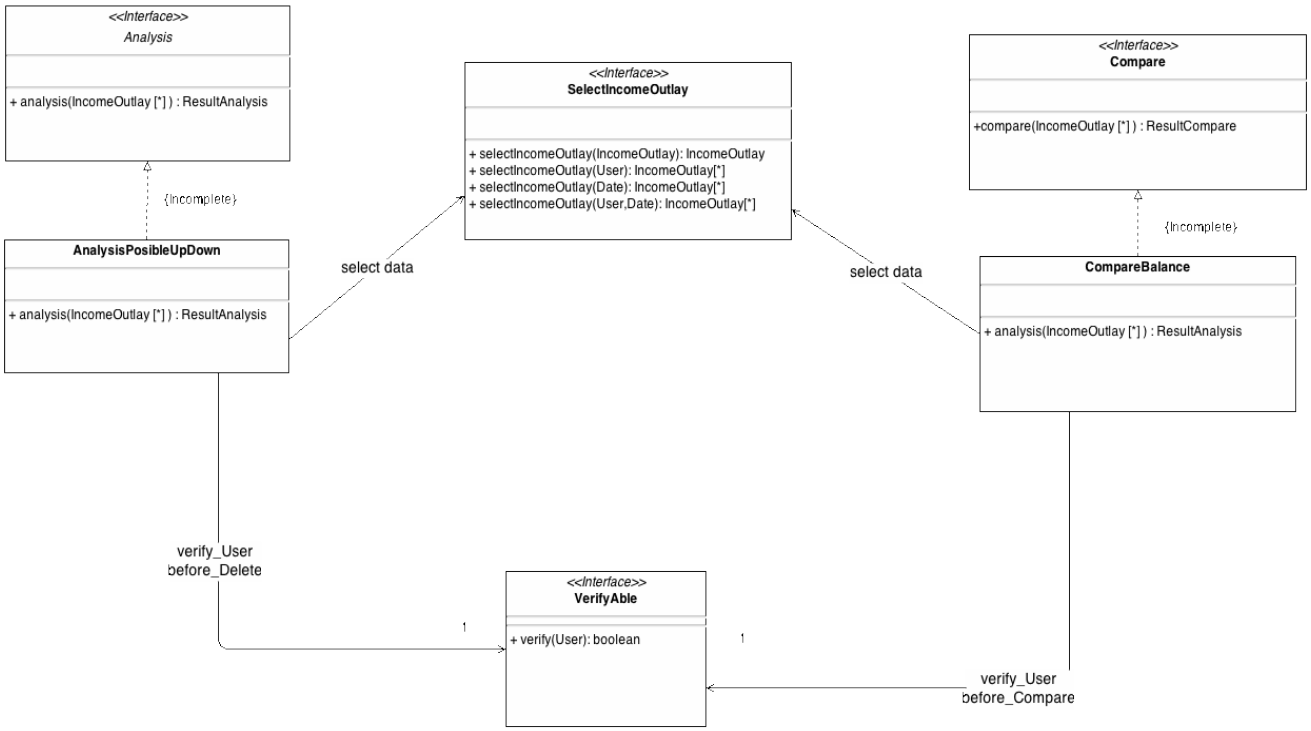
Class ที่เกี่ยวข้องกับส่วนจัดการรายรับรายจ่ายจะประกอบด้วย 3 class คือ

1. **AddIncomeOutlayManager** จะทำการ implement AddIncomeOutlayAble เพื่อใช้ในการเพิ่มรายรับรายจ่าย
2. **EditIncomeOutlayManager** จะทำการ implement EditIncomeOutlayAble เพื่อใช้ในการแก้ไขรายรับรายจ่าย
3. **DeleteIncomeOutlayManager** จะทำการ implement DeleteIncomeOutlayAble เพื่อใช้ในการลบรายรับรายจ่าย

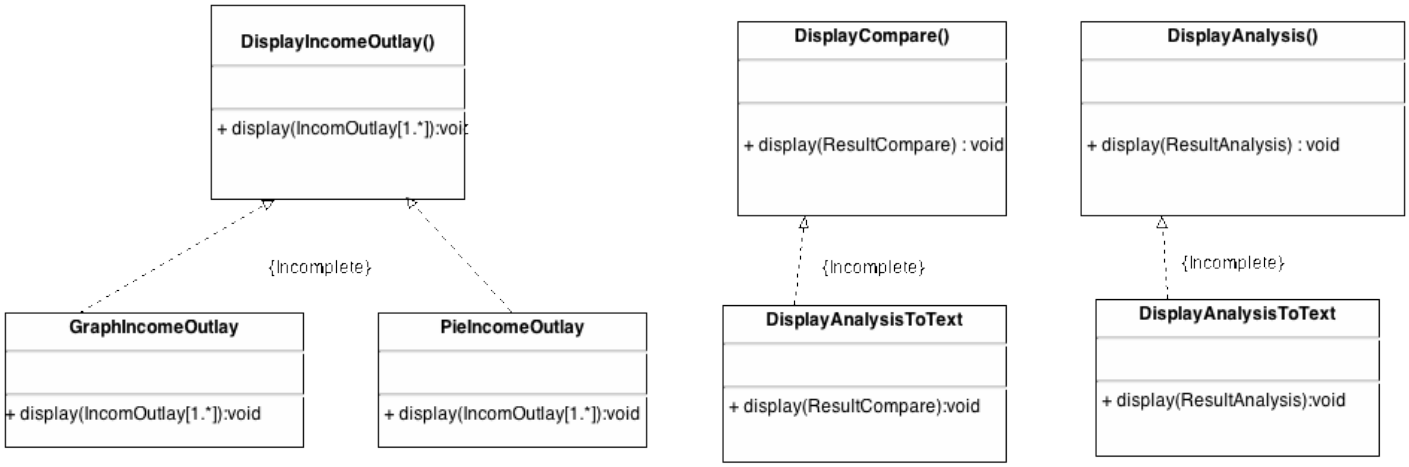
ซึ่งทั้ง 3 class นี้จะเรียกใช้ class ที่ implement interface VerifyAble และเรียกใช้ class ที่ implement interface ที่เกี่ยวกับการติดต่อกับแหล่งข้อมูลที่ทำอย่างนี้นั้นแทนการเขียนทำการตรวจสอบสิทธิ์ที่ตัว class เลยก็เพราะ ถ้ามีการเปลี่ยนรูปแบบการตรวจสอบสิทธ์ จะได้ไม่ต้องทำการแก้ไขที่ class นี้แต่ไปแก้ไขที่ class ที่ทำการ implement VerifyAble แทน หรือ ให้ class นี้เรียกใช้ class ที่ทำการ implement VerifyAble ที่มีการตรวจสอบสิทธิ์แบบใหม่



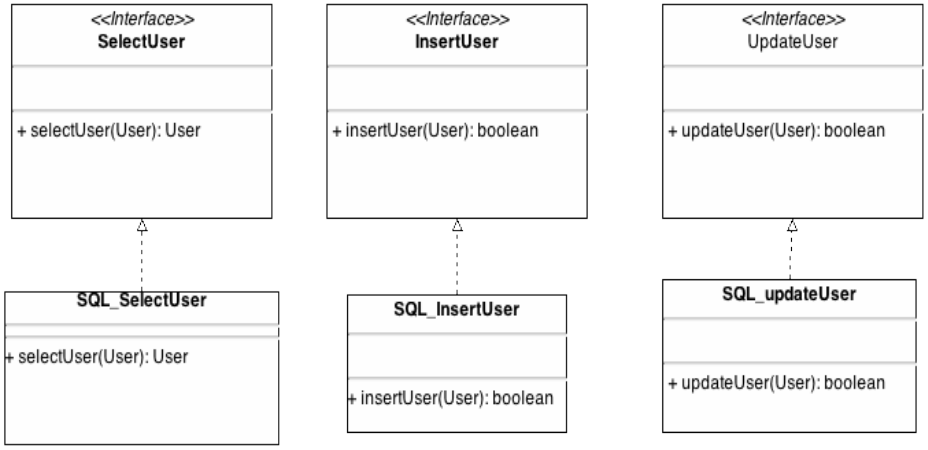
Class ที่เกี่ยวข้องกับส่วนวิเคราะห์ข้อมูล นั้นจะเป็น class ที่ทำการ implement interface Analysis และ Compare ซึ่งที่ออกแบบอย่างนี้เพราะอาจมีวิธีการวิเคราะห์ข้อมูลและเปรียบเทียบข้อมูลแบบอื่นๆอีกหลากหลายรูปแบบจึงทำ interface ให้คนภายนอกส่งข้อมูลมาวิเคราะห์หรือเปรียบเทียบอย่างเดียว ส่วนภายในเราจะใช้ขั้นตอนวิธีการ (Algorithm) ใหม่อย่างไรก็ได้เพียงแต่ต้อง implement interface Analysis หรือ Compare ก็พอ

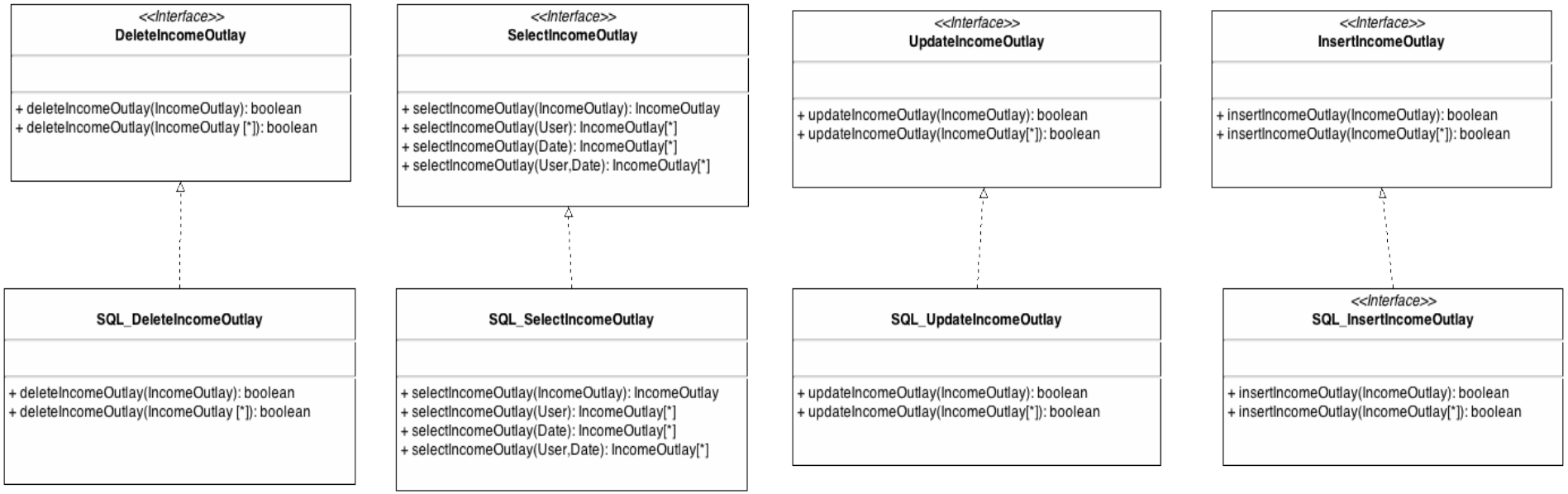


Class ที่เกี่ยวข้องกับส่วนแสดงผล นั้นจะเป็น class ที่ทำการ implement interface DisplayIncomeOutlay, DisplayAnalysis และ DisplayCompare โดยเหตุผลที่ทำอย่างนี้ก็คล้ายๆกับส่วนวิเคราะห์ข้อมูลคือ การแสดงผลนั้นอาจทำหลากหลายรูปแบบ เช่น ข้อมูล กราฟวงกลม กราฟเส้น ดังนั้นจึงเปิดให้ผู้ใช้เรียกใช้ interface



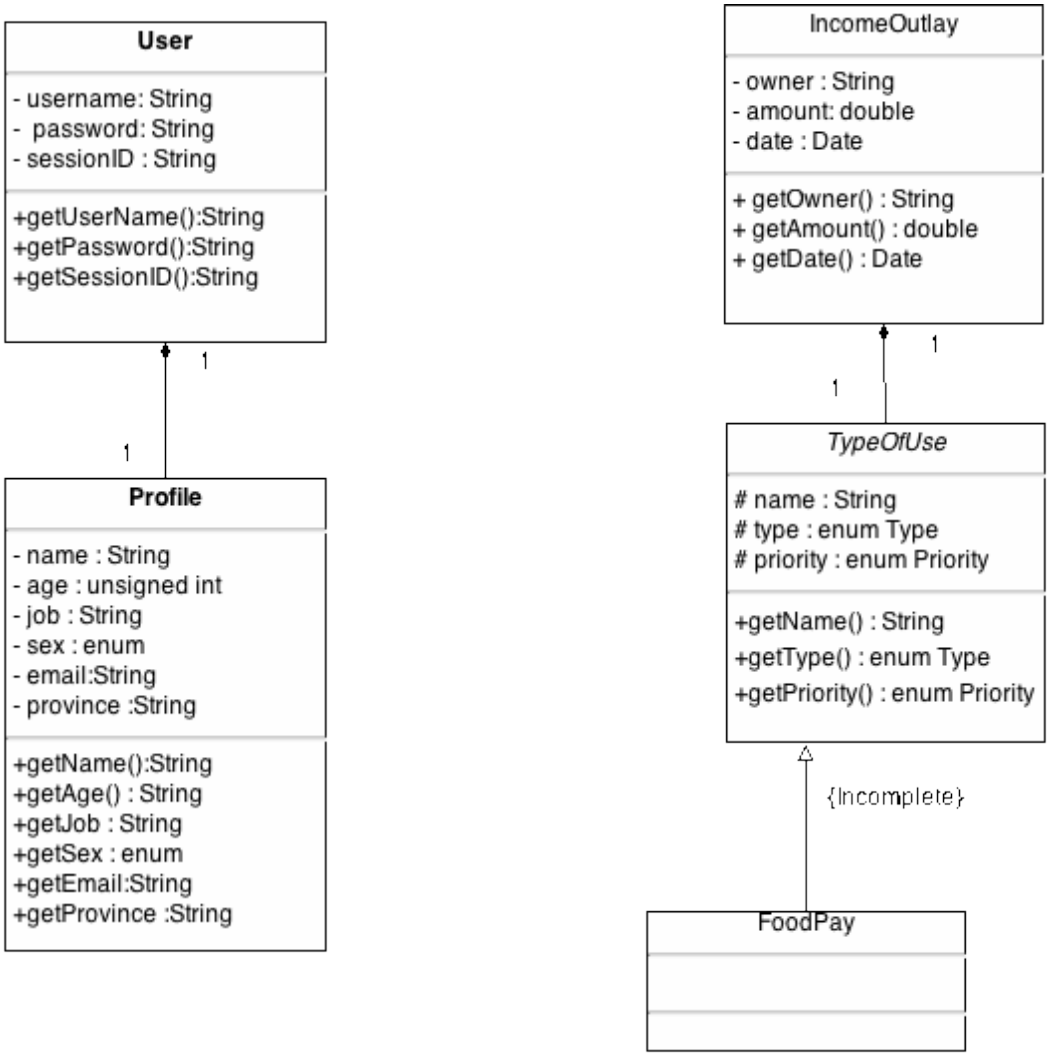
Class ที่เกี่ยวข้องกับส่วนบันทึกและค้นหาข้อมูล นั้นจะเป็น class ที่ทำการ implement interface ที่อยู่ในภาพ โดยที่ออกแบบอย่างนี้ก็เพราะต้องการให้ยืดหยุ่น เราสามารถปรับเปลี่ยนการทำงานในการไปค้นหาหรือลบข้อมูลตามความต้องการที่เปลี่ยนแปลงได้





Class User และ Profile (ซ้าย) นั้น Class user จะประกอบไปด้วย Profile โดยมีความสัมพันธ์แบบ 1 ต่อ 1 คือ User หนึ่งตัวมีได้ Profile เดียวและ Profile นั้นมีเจ้าของได้คนเดียวซึ่งในที่นี้สามารถเพิ่ม operation ในการ get หรือ set ข้อมูลที่เป็น private ได้ตามความเห็นสมควรว่าจะให้เอาข้อมูลออกมาได้หรือไม่

Class IncomeOutlay และ TypeOfUse (ขวา) นั้น IncomeOutlay นั้นจะประกอบไปด้วย TypeOfuse โดยมีความสัมพันธ์แบบ 1 ต่อ 1 โดยที่ออกแบบอย่างนี้เพราะต้องการให้ TypeOfUse มีได้หลายชนิดไม่จำกัดอยู่เพียงไม่กี่ชนิดและสามารถสร้างขึ้นใหม่ได้เรื่อยๆโดยสร้าง class ที่สืบทอดจาก TypeOfUse



Deployment

ส่วนที่เป็นหน้าแสดงผล

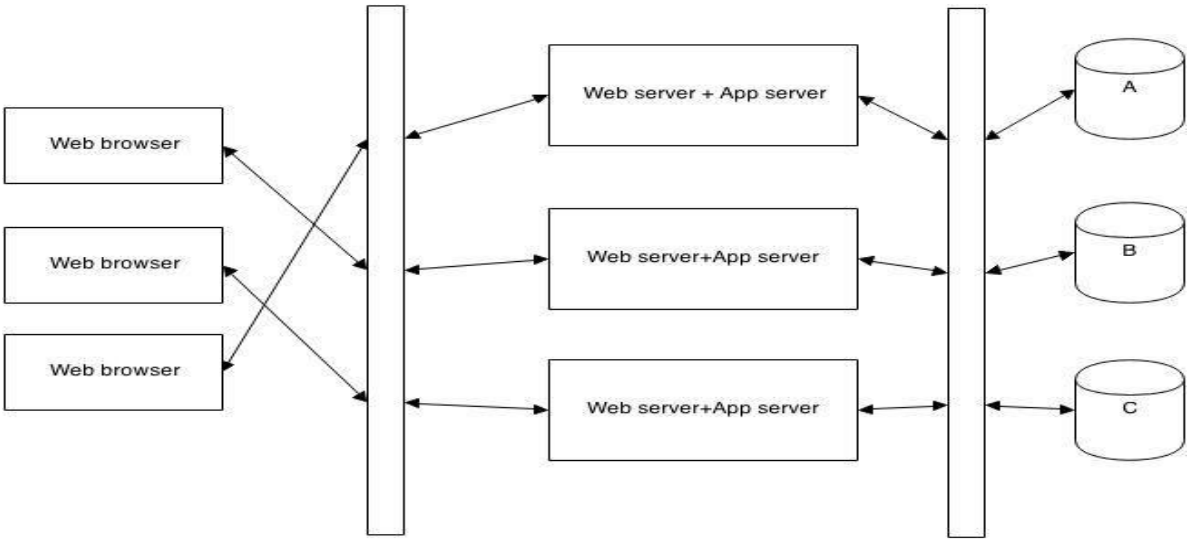
ในส่วนนี้จะใช้ HTML, CSS, Java Script ในการทำส่วนการแสดงผล(Presentation) ที่จะนำไปใช้กับ Web-browser ของผู้ใช้โดยจะเลือกใช้ Bootstrap เพื่อช่วยในการจัดหน้าจอเพื่อให้ใช้แสดงได้ในหลายอุปกรณ์ เช่น คอมพิวเตอร์ มือถือ ที่มีขนาดหน้าจอแตกต่างกัน ใช้ JQuery ที่เป็น API ของภาษา Java Script ซึ่งช่วยทำให้เขียนโปรแกรมได้สะดวกและง่ายขึ้นไม่ต้องทำการเขียนในหลายๆ ส่วนเอง

Web server

ในส่วนนี้จะใช้บริการ Web site ของ Microsoft Azure ที่มีไว้ให้ โดยทาง Azure จะสร้างเครื่องที่ทำการลงโปรแกรมที่จำเป็นต่างๆ ที่จำเป็นในการเป็นเครื่อง Server ไว้ให้ ตัวเราเพียงแค่ทำการส่งชุดคำสั่งของโปรแกรมของเราขึ้นไปบนตัวเครื่องเท่านั้น อีกทั้งตัวบริการนี้สามารถทำการเพิ่ม-ลดขนาด(Scale) โดยสร้างเครื่องใหม่ขึ้นมารับงานเมื่อเครื่อง Server ตัวเดียวเริ่มรับงานไม่ไหว ซึ่งสามารถตั้งกฎในการเพิ่มจำนวนเครื่องที่เป็น Instances ได้ง่ายโดยไม่จำเป็นต้องเขียนโปรแกรม โดยเราเลือกให้ตัว Web server ทำการลงตัว Apache tomcat และใช้ภาษา Java ในการเขียนส่วนที่เป็น Business logic และใช้ตัว Spring framework เป็นตัวช่วยในการติดต่อ

ส่วนของ Database

ในส่วนนี้จะใช้บริการของ SQL Database ของ Microsoft Azure ซึ่งมีให้อยู่แล้วอีกทั้งยังรับประกันความสามารถในการให้บริการ (Availability) อีกทั้งยังสามารถทำการ Scale up หรือ Scale down ได้ง่าย โดยจะเอา SQL Database ที่ได้นั้นมาทำเป็น Distribute database โดยกระจายข้อมูลแบบ Horizontal ไปให้ทุก Database เพื่อแบ่งภาระงานในการเขียนและอ่านข้อมูล โดยส่วนที่ทำการเลือกที่จะส่งข้อมูลลง Database ว่าจะลง Database ไหนนั้นจะทำการดำเนินการสร้าง (Implement) ขึ้นมาเอง



Implementation plan

งานที่ทำ	ระยะเวลา/วัน	ผู้รับผิดชอบ
1. ออกแบบส่วนที่เป็น business logic ว่าแต่ละระบบนั้นจะมี Class อะไรบ้าง แต่ละ Class ติดต่อกันอย่างไร	5 (20-24 ก.ย. 57)	นายสุรพงศ์ เท่าเทียมตน
2. ออกแบบส่วนที่เป็นส่วนแสดงผลให้กับผู้ใช้ว่าควรจะจัดวางอย่างไร และทดลองใช้ Bootstrap สร้างส่วนแสดงผลขึ้นมา	6 (25-30 ก.ย. 57)	นายพัสกร จุลพล
3. สร้างและออกแบบฐานข้อมูล	2 (9-10 ต.ค. 57)	นายสุรพงศ์ เท่าเทียมตน
4. เขียนส่วนที่เป็นส่วนจัดรูปแบบแสดงผล และ ส่วนระบบสมาชิก	4 (11-14 ต.ค. 57)	นายพัสกร จุลพล
5. เขียนนส่วนแสดงผลที่เกี่ยวกับระบบสมาชิก เช่น หน้าเข้าสู่ระบบ หน้าประวัติสมาชิก	4 (15-18 ต.ค. 57)	นายสุรพงศ์ เท่าเทียมตน
6. เขียนส่วนจัดการรายรับรายจ่าย ส่วนบันทึกและค้นหาข้อมูล	4 (19-22 ต.ค. 57)	นายสุรพงศ์ เท่าเทียมตน
7. เขียนส่วนแสดงผลที่เกี่ยวกับการจัดการรายรับรายจ่าย เช่น หน้าบันทึกรายรับรายจ่าย หน้าค้นหา หน้าลบรายรับรายจ่าย	4 (23-26 ต.ค. 57)	นายพัสกร จุลพล
8. เขียนส่วนวิเคราะห์และเปรียบเทียบข้อมูล	4 (27-30 ต.ค. 57)	นายสุรพงศ์ เท่าเทียมตน
9. เขียนส่วนแสดงผลการวิเคราะห์และเปรียบเทียบ	4 (31 ต.ค.-3 พ.ย. 57)	นายพัสกร จุลพล
10. เริ่มทำการสร้าง Distribute database และทำส่วนที่ทำการแยกว่าจะเลือกค้นหาหรือบันทึกลง Database ตัวไหน	4 (4-7 พ.ย. 57)	นายสุรพงศ์ เท่าเทียมตน
11. การ Test แต่ละส่วนของระบบ	7 (8-14 พ.ย. 57)	นายพัสกร จุลพล