

บทที่ 3

ขั้นตอนการดำเนินงาน

บทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานของโครงการปริญญาโท ซึ่งจะทำให้ทราบถึงการวิเคราะห์และการออกแบบระบบโดยละเอียดว่ามีแนวทางในการทำงานหรือขั้นตอนในการทำงานของระบบอย่างไร โดยขั้นตอนในการดำเนินงานมีรายละเอียดดังนี้

3.1 กรอบการดำเนินงาน

กรอบการทำงานนี้จะแสดงขั้นตอนการพัฒนาระบบ แอปพลิเคชันการจองอัตราแลกเปลี่ยนสกุลเงินไทย ซึ่งมีขั้นตอนการดำเนินงานดังนี้

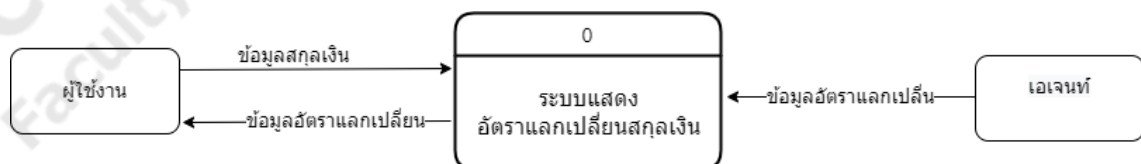
3.1.1 คำอธิบาย

- 1) เก็บข้อมูล ทำการเก็บข้อมูลทั้งหมดของผู้ดูแลระบบ บริษัทแลกเปลี่ยนสกุลเงิน อัตราแลกเปลี่ยนสกุลเงิน
- 2) ฐานข้อมูล นำข้อมูลจากการเก็บข้อมูลมาบันทึกลงในฐานข้อมูล
- 3) แอปเว็บพลิเคชัน
- 4) ทดสอบระบบ

3.2 การไหลของข้อมูล (Data Flow Diagram)

3.2.1 แผนภาพคอนแท็กซ์ (Context Diagram)

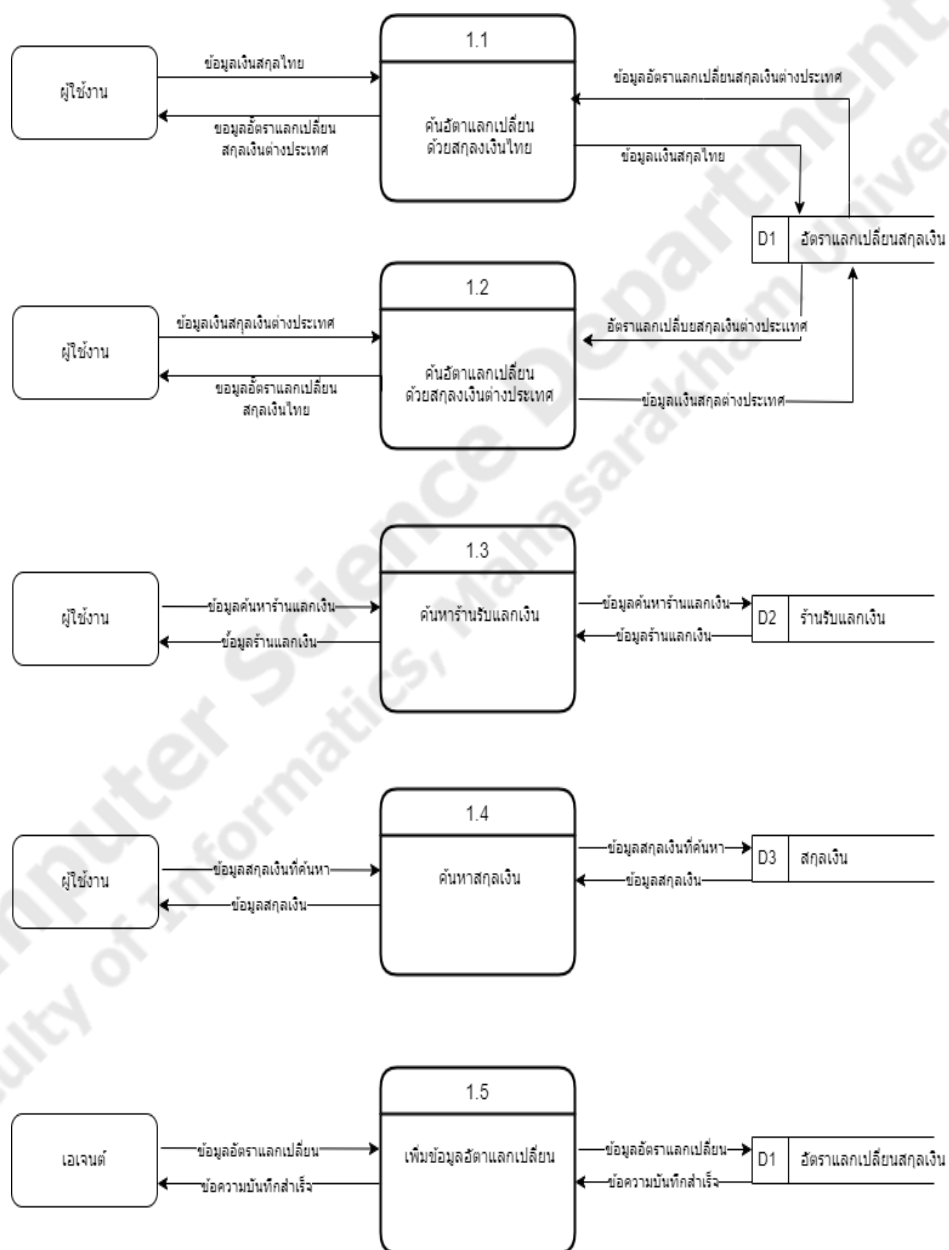
แผนภาพกระแสข้อมูลที่แสดงภาพรวมการทำงานของระบบที่มีความสัมพันธ์กับผู้ดูแลระบบ บริษัทแลกเปลี่ยน และผู้ใช้ เพื่อแสดงถึงขอบเขตของระบบการจองอัตราแลกเปลี่ยนสกุลเงินไทย



ภาพประกอบที่ 3.1 แผนภาพคอนแท็กซ์ (Context Diagram)

3.2.2 Data Flow Diagram Level 1

แผนภาพกระแสข้อมูลในระดับที่แสดงขั้นตอนการทำงานหลักทั้งหมด (Process หลัก) ของ ระบบ แสดงทิศทางการไหลของ Data Flow และแสดงรายละเอียดของแหล่งจัดเก็บข้อมูล (Data Store)



ภาพประกอบที่ 3.2 แผนภาพกระแสข้อมูลเลเวล-1

3.3 พจนานุกรมข้อมูล (Data Dictionary)

3.3.1 อธิบายอินทิตีภายนอก (External Entity Description)

ตารางที่ 3.1 External Entity Description

Name	Description	Input Data Flow	Output Data Flow
ผู้ใช้	เป็นผู้ใช้งานระบบที่มี ความสามารถในการ ค้นหา คำนวณอัตรา แลกเปลี่ยนสกุลเงิน	- ข้อมูลค้นหาอัตรา แลกเปลี่ยนสกุลเงินไทย -ข้อมูลค้นหาอัตรา แลกเปลี่ยนสกุลเงิน ต่างประเทศ -ข้อมูลค้นหาร้านแลกเปลี่ยน -ข้อมูลค้นหาสกุลเงิน	- ข้อมูลอัตราการแลกเปลี่ยน สกุลเงิน - ข้อมูลร้านแลกเปลี่ยน -ข้อมูลสกุลเงิน
เอเจนต์	เป็นโมดูลโปรแกรมที่ ทำการดึงข้อมูลจาก ร้านที่แสดงอัตราการ แลกเปลี่ยน	- อัตราการแลกเปลี่ยนของแต่ละ ร้านแลกเปลี่ยน	- อัตราการแลกเปลี่ยนของแต่ละ ร้านแลกเปลี่ยน

3.3.2 อธิบายการไหลของข้อมูลและโครงสร้าง (Data Flow and Data Structure Description)

ตารางที่ 3.2 Data Flow and Data Structure Description

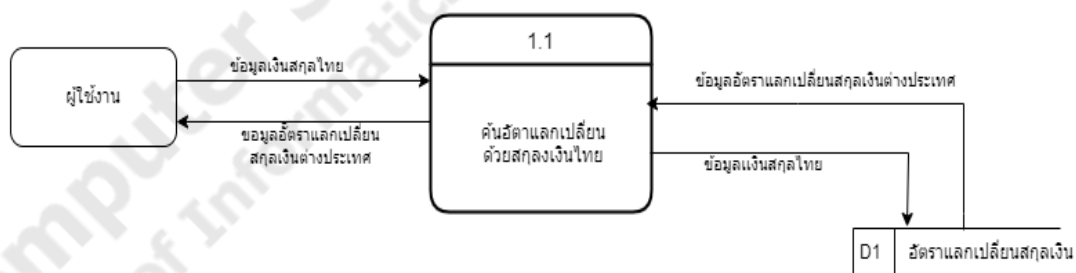
Name	Description	Source	Destination	Data Structure
ข้อมูลอัตรา แลกเปลี่ยน สกุลเงินไทย	ค้นหาข้อมูล อัตรา แลกเปลี่ยน สกุลเงินไทย	ผู้ใช้งาน	Process 1.1 ค้นหาข้อมูลผู้ใช้	รหัสสกุลเงินไทย + รหัสสกุลเงิน
		Process 1.1	D1 ข้อมูลอัตรา แลกเปลี่ยน	ต่างประเทศ + จำนวนแลกเปลี่ยน
ข้อมูลอัตรา แลกเปลี่ยน สกุลเงิน ต่างประเทศ	ค้นหาข้อมูล อัตรา แลกเปลี่ยนสกุล เงินต่างประเทศ	ผู้ใช้งาน	Process 1.2 ข้อมูลผู้ใช้	รหัสสกุลเงินไทย + รหัสสกุลเงิน
		Process 1.1	D1 ข้อมูลอัตรา แลกเปลี่ยน D3 ข้อมูลสกุลเงิน	ต่างประเทศ + จำนวนแลกเปลี่ยน

3.3.3 อธิบายการเก็บข้อมูลและโครงสร้าง (Data Store and Data Structure Description)

ตารางที่ 3.3 Data Store and Data Structure

ID	Data Store Name	Description	Data Structure
D1	ข้อมูลอัตราแลกเปลี่ยน	เก็บข้อมูลข้อมูลอัตราแลกเปลี่ยน	<u>รหัสับอัตรา</u> + <u>รหัสร้านแลกเงิน</u> + <u>รหัสสกุลเงิน</u> + <u>อัตราแลกเปลี่ยน</u> + <u>ชื่อสกุลเงิน</u> + <u>ชื่อร้านแลกเงิน</u> + <u>ที่อยู่ร้านแลกเงิน</u> + <u>เบอร์โทร</u> + <u>อีเมล</u>
D2	ข้อมูลร้านแลกเงิน	เก็บข้อมูลร้านแลกเงิน	<u>รหัสร้านแลกเงิน</u> + <u>ชื่อร้านแลกเงิน</u> + <u>ที่อยู่ร้านแลกเงิน</u> + <u>เบอร์โทร</u> + <u>อีเมล</u>
D3	ข้อมูลสกุลเงิน	เก็บข้อมูลสกุลเงิน	<u>รหัสสกุลเงิน</u> + <u>ชื่อสกุลเงิน</u>

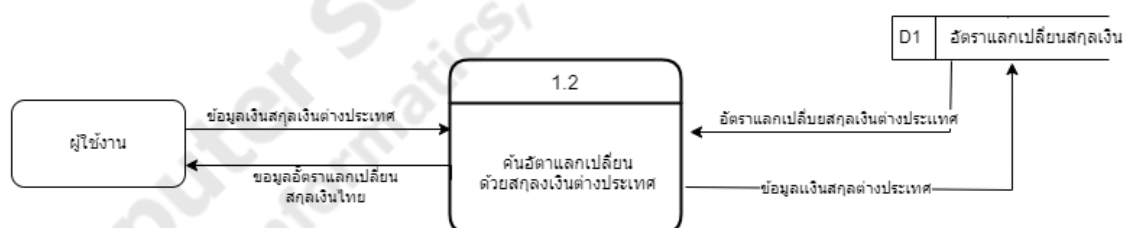
3.3.4 อธิบายการประมวลผล (Process Description)



ภาพประกอบที่ 3.3 แผนภาพการประมวลผลของการค้นหาอัตราแลกเปลี่ยนด้วยสกุลเงินไทย

ตารางที่ 3.4 อธิบายการประมวลผลการค้นหาอัตราแลกเปลี่ยนสกุลเงินไทย

ID	1.1
NAME	ค้นหาอัตราแลกเปลี่ยนสกุลเงินไทย
DESCRIPTION	เป็นกระบวนการค้นหาข้อมูลอัตราแลกเปลี่ยนสกุลเงินไทย
INPUT DATA FLOWS	ข้อมูลเงินสกุลไทย
OUTPUT DATA FLOWS	ข้อมูลอัตราแลกเปลี่ยนสกุลเงินต่างประเทศ
PROCESS DESCRIPTION	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลสกุลเงินไทย 2. คำนวณอัตราแลกเปลี่ยน ข้อมูลสกุลเงินไทย คูณ ข้อมูลอัตราการขาย 3. แสดงอัตราแลกเปลี่ยนสกุลเงินต่างประเทศโดยที่แสดงข้อมูลร้านที่ได้ราคาแลกเปลี่ยนที่ดีที่สุดขึ้นมาก่อนอันดับแรก <p>จบการทำงาน</p>



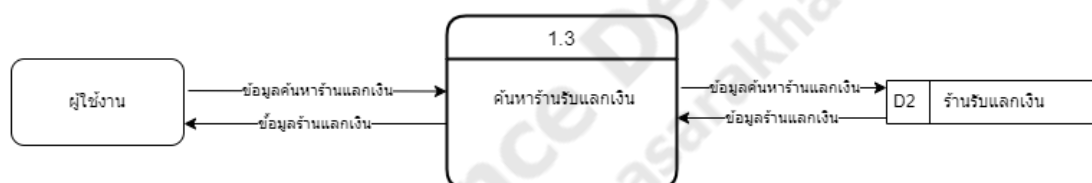
ภาพประกอบที่ 3.4 แผนภาพการประมวลผลของการค้นหาอัตราแลกเปลี่ยนด้วยสกุลเงิน

ตารางที่ 3.5 อธิบายการประมวลผลค้นหาอัตราแลกเปลี่ยนด้วยสกุลเงินต่างประเทศ

ID	1.2
NAME	ค้นหาอัตราแลกเปลี่ยนสกุลเงินต่างประเทศ
DESCRIPTION	เป็นกระบวนการค้นหาอัตราแลกเปลี่ยนด้วยสกุลเงินต่างประเทศ
INPUT DATA FLOWS	ข้อมูลสกุลเงินต่างประเทศ

ตารางที่ 3.6 อธิบายการประมวลผลค้นหาอัตราแลกเปลี่ยนด้วยสกุลเงินต่างประเทศ(ต่อ)

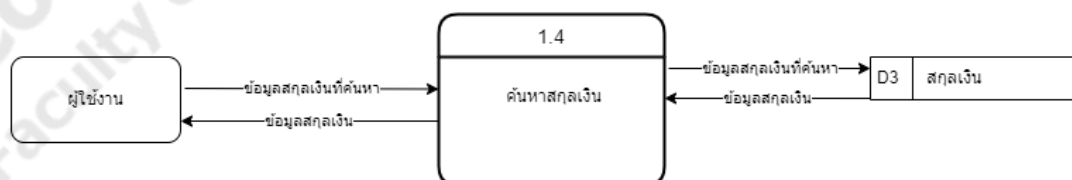
OUTPUT DATA FLOWS	ข้อมูลอัตราแลกเปลี่ยนสกุลเงินไทย
PROCESS DESCRIPTION	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลสกุลเงินต่างประเทศ 2. คำนวณอัตราแลกเปลี่ยน ข้อมูลสกุลเงินต่างประเทศ คุณ ข้อมูลอัตราการค้าซื้อ 3. แสดงอัตราแลกเปลี่ยนสกุลเงินต่างประเทศโดยที่แสดงข้อมูลร้านที่ได้ราคาแลกเปลี่ยนที่ดีที่สุดขึ้นมาก่อนอันอับแรก <p>จบการทำงาน</p>



ภาพประกอบที่ 3.5 แผนภาพการประมวลผลของการค้นหาร้านแลกเงิน

ตารางที่ 3.7 อธิบายการประมวลผลการค้นหาร้านแลกเงิน

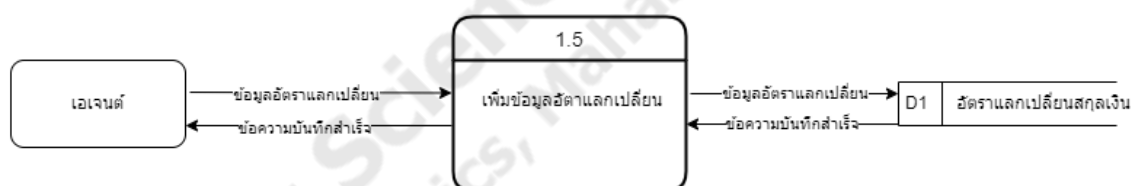
PROCESS DESCRIPTION	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลร้านแลกเงิน 2. แสดงข้อมูลร้านแลกเงิน <p>จบการทำงาน</p>
---------------------	---



ภาพประกอบที่ 3.6 แผนภาพประมวลผลของการค้นหาสกุลเงิน

ตารางที่ 3.8 อธิบายการประมวลผลการค้นหาสกุลเงิน

ID	1.4
NAME	ค้นหาสกุลเงิน
DESCRIPTION	เป็นกระบวนการค้นหาสกุลเงินต่างประเทศ
INPUT DATA FLOWS	ข้อมูลสกุลเงินที่ค้นหา
OUTPUT DATA FLOWS	ข้อมูลสกุลเงิน
PROCESS DESCRIPTION	เริ่มต้น 1. รับข้อมูลสกุลเงินที่ค้นหา 2. แสดงข้อมูลสกุลเงิน จบการทำงาน



ภาพประกอบที่ 3.7 แผนภาพการประมวลผลของเพิ่มข้อมูลอัตราแลกเปลี่ยน

ตารางที่ 3.9 อธิบายการประมวลผลการเพิ่มข้อมูลอัตราแลกเปลี่ยน

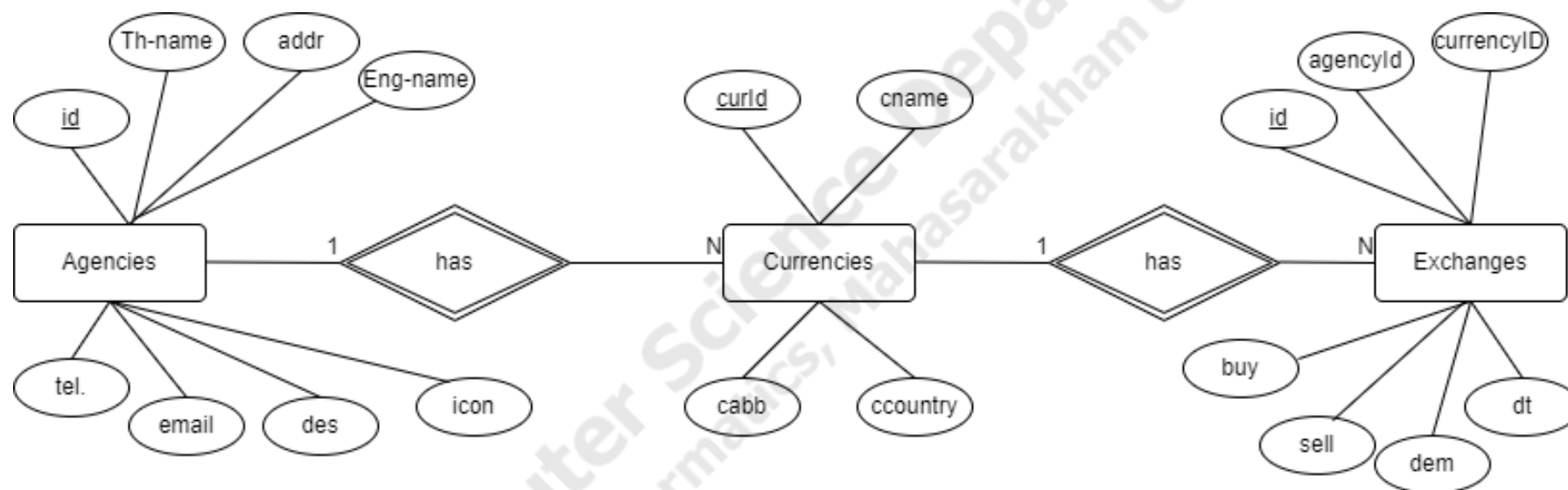
ID	1.5
NAME	เพิ่มข้อมูลอัตราแลกเปลี่ยน
DESCRIPTION	เป็นกระบวนการเพิ่มข้อมูลอัตราการแลกเปลี่ยนสกุลเงิน
INPUT DATA FLOWS	ข้อมูลอัตราแลกเปลี่ยนของร้าน 12 ร้าน
OUTPUT DATA FLOWS	ข้อมูลร้านทั้ง 12 ร้าน

ตารางที่ 3.10 อธิบายการประมวลผลการเพิ่มข้อมูลอัตราแลกเปลี่ยน

PROCESS	เริ่มต้น
DESCRIPTION	<ol style="list-style-type: none"> 1. ดึงข้อมูลของร้านรับแลกเปลี่ยนเงิน 2. แสดงข้อความแจ้งเตือน <ol style="list-style-type: none"> 2.1 ถ้าดึงข้อมูลสำเร็จ แสดงข้อความ “ดึงข้อมูลสำเร็จ” 2.2 ถ้าดึงข้อมูลไม่สำเร็จ แสดงข้อความ “ดึงข้อมูลไม่สำเร็จ” <p>จบการทำงาน</p>

3.4 ความสัมพันธ์ (Entity Relationship Diagram)

3.4.1 Entity Relationship Diagram (ER-Diagram)



ภาพประกอบที่ 3.8 ER Diagram

3.5 การออกแบบฐานข้อมูล (Database Design)

ตารางที่ 3.11 Agencies

ID	Column	Type	Description	Example Data	Constraint
1	agencyId	varchar(11)	รหัสสมาชิก	ID1	PK
2	thainame	varchar(50)	ชื่อบริษัท	บริษัทเขียว	Not Null
3	engname	varchar(50)	ชื่อบริษัท	SuperRich Currency	Not Null
3	username_agent	varchar(50)	ชื่อบัญชี	agent01	Not Null
4	email	varchar(30)	อีเมล	agent01@gmail .com	Not Null
5	tel	int(10)	เบอร์	0935462770	Not Null
6	add	varchar(100)	ที่อยู่	451 ถนน พระราม ที่ 1 แขวงปทุมวัน เขตปทุมวัน กรุงเทพมหานคร 10330	Not Null
8	desp	varchar(500)	ผู้ให้บริการ แลกเปลี่ยน ข้อมูล	Thailand	Not Null
9	icon	varchar(255)	ไอคอน	SAO	Not Null

ตารางที่ 3.12 Currencies

ID	Column	Type	Description	Example Data	Constraint
1	currencyId	varchar(11)	รหัสสมาชิก	ID1	PK
2	cname	varchar(50)	ชื่อสกุลเงิน	USD - US Dollar	Not Null
3	ccountry	varchar(50)	ชื่อประเทศ	USA	Not Null
4	cabb	varchar(20)	ชื่อย่อสกุลเงิน	USD	Not Null

ตารางที่ 3.13 Exchange

ID	Column	Type	Description	Example Data	Constraint
1	id	varchar(255)	รหัสสมาชิก	ID1	PK
2	currencyId	varchar(255)	รหัสสมาชิก โดยจะอ้างอิง มาจากtable currencies	ID2	FK
3	agencyId	varchar(11)	รหัสสมาชิก โดยจะอ้างอิง มาจากtable agency	ID3	FK
4	buy	varchar(50)	ราคาซื้อ	36.25	Not Null
5	sell	varchar(255)	ราคาขาย	36.65	Not Null
6	dem	varchar(255)	หน่วยเงิน ค่าเงิน	100	Not Null
7	dem2	varchar(255)	หน่วยเงิน ค่าเงิน	500	Not Null
8	flag	int	คือเป็นตัว เอาไว้ใช้ว่ามี รูปหรือไม่มี รูป มีคือ 1 ไม่ มีคือ 0	1	Not Null
9	dt	varchar(255)	เก็บค่าวัน เดือนปีเวลา	"2023-01-16 23:06:14.743464"	Not Null
10	cpbuy	varchar(255)	ราคาซื้อ ณ ตอนนั้น	0	Not Null
11	cpsell	varchar(255)	ราคาขาย ณ ตอนนั้น	0	Not Null

3.6 ขั้นตอนการใช้งาน Python Web Scraping

ในการทำ Web Scraping โดยจะทำการเรียก library ที่ชื่อว่า BeautifulSoup และ Selenium ในการใช้ทำ Web Scraping โดยเริ่มต้นจะกำหนด url link ของเว็บไซต์ที่ต้องการจะใช้ Scraping จากเว็บไซต์นั้นๆ

```

2 from bs4 import BeautifulSoup
3 from selenium import webdriver
4 from selenium.webdriver.common.keys import Keys
5 import time
6
7 url = "https://www.superrich1965.com/currency.php"

```

ภาพประกอบที่ 3.9 ขั้นตอนการใช้งาน Python Web Scraping

```

9 driver = webdriver.Edge(executable_path='C:/Users/shipd/Desktop/project file/Code project
10 driver.get(url)
11
12 time.sleep(5)
13
14 html = driver.page_source

```

ภาพประกอบที่ 3.10 ขั้นตอนการใช้งาน Python Web Scraping (ต่อ)

ขั้นตอนต่อมา จะตั้งตัวแปร soup โดยที่เรียกใช้จาก Library BeautifulSoup โดยใส่ตัว parameter คือ soup(html, "html.parser") แล้วต่อมาก็จะสร้างตัวแปรที่ชื่อว่า all_divs

ซึ่งที่เขียนเป็นคำสั่งคือ all_divs = soup.find('div', {'class': 'container'}).text ซึ่งตัวแปรนี้จะทำการเก็บค่าจากตัวแปร soup ที่ไปเก็บค่าต่างๆจาก Tag div อยู่ที่ class ที่มีชื่อว่า container โดยแปลงค่าต่างๆเป็น Text ซึ่งในตัวแปร li และ tr ก็จะทำคล้ายๆกันแต่จะใช้ Method ที่เป็น find_all

```
li = soup.find_all('div', {'class': 'table shadow'})
```

```
tr = soup.find_all('table', {'class': 'table'})
```

ทั้งสองตัวแปรนี้ จะทำการหาข้อมูลทั้งหมดที่ได้กำหนดใน parameter ก็คือ tag div แล้วเอาข้อมูลที่ class ซึ่งอยู่ที่ 'table shadow' ซึ่งอีกตัวก็จะทำคล้ายๆกันแต่จะไปดึงข้อมูลที่ Tag table อยู่ที่ class ข้อมูลที่ table

```

16 soup = BeautifulSoup(html, "html.parser")
17 all_divs = soup.find('div', {'class': 'container'}).text
18 # transcript = all_divs.find('table', class_='table')
19 li = soup.find_all('div', {'class': 'table shadow'})
20 tr = soup.find_all('table', {'class': 'table'})

```

ภาพประกอบที่ 3.11 ขั้นตอนการใช้งาน Python Scraping (ต่อ)

หลังจากนั้นก็สร้างที่แปรที่เป็น array เพื่อที่จะเอาไว้เก็บค่าที่ได้ Web Scraping จากนั้นก็จะทำการสร้าง for loop เพื่อที่จะเก็บค่าจากการ Web Scraping ทุกๆตัวจากแต่ละ tag แต่ละ div ในเว็บไซต์ที่เราต้องการที่จะ Scraping โดยเราจะกำหนดค่า parameter ต่างๆ คือ จะนำตัวแปร tr มา for loop แล้วเก็บค่าไว้ใน row ที่นำมา for loop คือ จะให้หาตัวค่าที่ต้องการใน tag tr โดยใน for loop จะสร้างตัวแปร td ซึ่งก็ได้เก็บค่า tr ไว้ใน row ก็จะเรียกใช้ method findAll เพื่อหาค่าที่ต้องการคือ หาค่าจาก td ดังบรรทัด 27 ตามรูปประกอบ ตัวแปร k=0 เพื่อที่จะนำไปหาค่าใน for loop ต่อไป โดยใน for นี้จะเอาไว้หาค่าสกุลเงินที่อยู่เว็บไซต์ที่ต้องการ Web Scraping ว่ามีกี่ค่าสกุลเงินทั้งหมดใน เว็บไซต์ที่กำหนด

```

21 td = []
22 cur = []
23 dem = []
24 buy = []
25 sel = []
26 for row in tr:
27     td = row.findAll('td')
28     k = 0
29     for t in td:
30         if(k%5 == 0 ):
31             cur.append(t.text[0:3])
32             k+=1
33     k=0
34     p=0

```

ภาพประกอบที่ 3.12 ขั้นตอนการใช้งาน Python Web Scraping (ต่อ)

ต่อมาใน for loop จะทำการหาค่า dem, buy, sel และการตัดตัวอักษร ที่ติดมาจาก text html นั่นก็คือ '
' เริ่มแรกก็จะทำการสร้าง for t in td โดยที่ t จะเก็บค่าข้อมูลจาก td เริ่มแรกจะกำหนด ตัวแปร x ที่เอาไว้จัดการกับ text html ที่ติดมาด้วย โดยจะใช้ method split เพื่อทำการแยกอักษรที่ไม่ได้ต้องการ หลังจากนั้นก็จะใช้ method strip เพื่อทำการลบช่องว่าง

```

35     for t in td:
36         x = str(t).split('<br/>')
37         x[0] = x[0].split('>')[1].strip()
38         x[len(x)-1] = x[len(x)-1].split('<')[0].strip()

```

ภาพประกอบที่ 3.13 ขั้นตอนการใช้งาน Python Web Scraping (ต่อ)

หลังจากนั้นก็เข้าเงื่อนไข if คือ ถ้า $(\text{len}(x[0]) > 0)$ จะให้ทำเงื่อนไข if ต่อไปคือ ถ้า $(x[0].find('<') < 0)$ จะให้ $k += 1$ โดยที่ถ้า $k == 1+p$ ก็จะทำให้ ค่า dem ที่ได้กำหนดไว้ในตอนแรก append ไปที่ x หลังจากนั้นก็จะทำการ print ค่าที่ได้ทำการ Scraping มาดูทั้งหมด ว่าก็รายการทั้งหมด

```

39         if(len(x[0]) > 0):
40             if(x[0].find('<') < 0):
41                 k+=1
42                 if k == 1+p :
43                     dem.append(x)
44                 elif k == 2+p:
45                     buy.append(x)
46                 elif k== 3+p:
47                     sel.append(x)
48                     p+=3
49                 break
50     rate = []
51
52     print(len(cur), len(sel), len(dem), len(buy))
53

```

ภาพประกอบที่ 3.14 ขั้นตอนการใช้งาน Python Web Scraping (ต่อ)

จะเห็นได้ว่าในบรรทัดที่ 50 จะสร้างตัวแปร rate ที่เป็น array ซึ่งจะเอาไว้เก็บค่า หน่วยเงิน (denomination) โคนจะใช้ for loop ในการหาจากค่าที่ได้จากตัวแปร dem ที่ได้เก็บมาจากตอนต้น เริ่มแรกจะทำการ for loop โดยเก็บค่า dem ไว้ที่ i แล้วสร้างตัวแปร c ที่เอาไว้เก็บค่า cur หลังจากนั้นก็จะสร้าง for loop อีกตัว โดยจะเก็บค่า dem[i] ที่ได้จาก for loop แรกซึ่งจะเก็บไว้ที่ตัวแปร j โดยจะทำการสร้างตัวแปร d ที่เอาไว้ใช้ในการเก็บค่าข้อมูล โดยบรรทัดที่ 58 นั้นจะเป็นการให้ d ที่ข้อมูลที่ชื่อว่า ['cur'] = c.strip() ก็คือให้ข้อมูลที่มีชื่อดังกล่าวนั้นลบช่องว่างออก จากนั้นก็เข้าเงื่อนไข if คือ ถ้า $c == \text{'VND'}$ หรือ $c == \text{'IDR'}$ ให้ $d[\text{dem}1] = \text{dem}[\text{i}][\text{j}][2:6]$ สามารถอธิบายได้ว่า เมื่อในตัวแปร c มีข้อมูลที่

ชื่อว่า VND หรือ IDR จะให้ตัวแปร d สร้าง Object ที่ชื่อ dem1 และ dem2 นั้นหาตั้งแต่ตัวที่ l และ j จากตำแหน่งที่ 2 ถึง 6

```

54 for i in range(len(dem)):
55     c = cur[i]
56     for j in range(len(dem[i])):
57         d = {}
58         d['cur'] = c.strip()
59         if c == 'VND' or c == 'IDR':
60             d['dem1'] = dem[i][j][2:6]
61             d['dem2'] = dem[i][j][2:6]

```

ภาพประกอบที่ 3.15 ขั้นตอนการใช้งาน Python Web Scraping (ต่อ)

หรือจะเข้าเงื่อนไข else ในบรรทัดที่ 63 ว่าถ้า `if(dem[i][j].find('-') > 0)` คือถ้าตัวแปร dem ตัวที่ l และ j หาเครื่องหมาย - เจอจะมีค่ามากกว่า 0 แล้วจะให้ `d['dem1'] = dem[i][j].split('-')[1].strip()` จะทำการให้ตัวแปร dem ตัวที่ l และ j แยกเครื่องหมาย - ออกจากตำแหน่งที่ 1 แล้วก็ใช้คำสั่ง `strip()` เพื่อลบช่องว่างออก

`d['dem2'] = dem[i][j].split('-')[0].strip()` .ในคำสั่งนี้ก็ทำเหมือนกันแต่จะเป็นในตำแหน่งที่ 0 หรือเข้าเงื่อนไข else อีกรอบว่า

`d['dem1'] = dem[i][j].strip()` ให้ลบช่องว่างข้อมูลที่เก็บไว้ในตัวแปร dem ตัวที่ l และ j ออก โดยใช้คำสั่ง `strip()`

`d['dem2'] = dem[i][j].strip()` ให้ลบช่องว่างข้อมูลที่เก็บไว้ในตัวแปร dem ตัวที่ l และ j ออก โดยใช้คำสั่ง `strip()`

ในส่วนของ บรรทัดที่ 69-70 ให้ตัวแปร d สร้าง Object ชื่อ buy ,sell เพื่อให้เอาไว้เก็บค่าจากตัวแปร buy และ sel ที่เป็น array ด้วยจะให้ buy ตัวที่ l และ j ใช้คำสั่ง `strip()` เพื่อลบค่าช่องว่าง ในส่วนตัวแปร sel ก็ทำเหมือนกัน แล้วย้ายไปเก็บไว้ที่ `d['buy']`, `d['sell']` หลังจากนั้นก็ทำการ append ค่า d ทั้งหมดลงตัวแปร rate

```

62         else:
63             if(dem[i][j].find('-') > 0):
64                 d['dem1'] = dem[i][j].split('-')[1].strip()
65                 d['dem2'] = dem[i][j].split('-')[0].strip()
66             else:
67                 d['dem1'] = dem[i][j].strip()
68                 d['dem2'] = dem[i][j].strip()
69         d['buy'] = buy[i][j].strip()
70         d['sell'] = sel[i][j].strip()
71
72         rate.append(d)
73

```

ภาพประกอบที่ 3.16 ขั้นตอนการใช้งาน Python Web Scraping (ต่อ)

ในส่วนต่อมาจะทำการนำข้อมูลที่ได้มานั้น export ออกมาเป็นไฟล์เพื่อให้ง่ายต่อการนำไปใช้งานคืออย่างแรกจากเรียกใช้ method datetime เพื่อให้รู้ว่าข้อมูลที่เราได้มานั้นมาจากช่วงเวลาไหน จากนั้นก็จะสร้างตัวแปร fn ที่เอาไว้เก็บ path ที่จะ export file ออกไปก็กำหนด path ที่ต้องการ หลังจากนั้น ก็จะใช้คำสั่ง with open(fn,'w') as data: โดยด้านในจะมี for loop ที่เรียกค่าจาก ตัวแปร rate ที่ได้เก็บข้อมูลต่างๆจากการ Scraping โดยจะใช้คำสั่ง write แล้วก็จะใส่ parameter ที่ต้องการจะเขียนข้อมูลลงให้ครบหลังจากนั้นก็ print('done') เพื่อทำการเช็คว่าการทำงานสำเร็จ

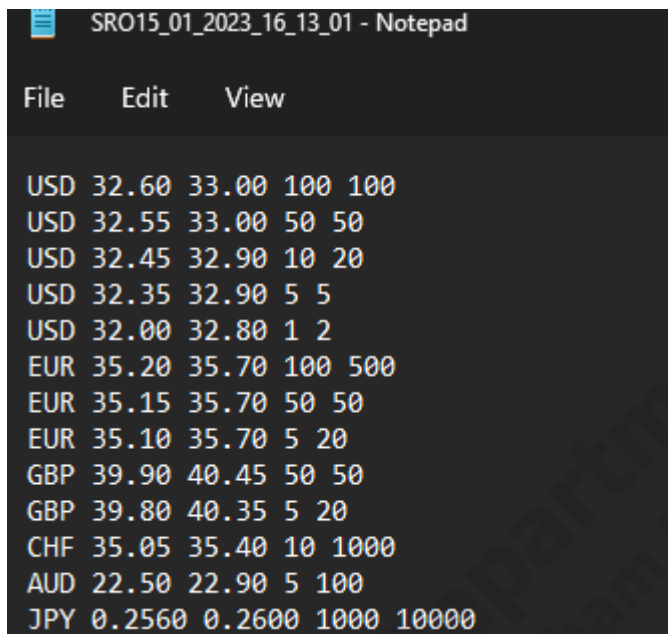
```

74 from datetime import datetime
75
76 fn = 'C:\\Users\\shipd\\Desktop\\project file\\Code project\\currency\\SRO'+str(datetime.now())
77
78 with open(fn,'w') as data:
79     for v in rate:
80         data.write('%s %s %s %s %s\n' % (v['cur'], v['buy'], v['sell'], v['dem1'], v['dem2']))
81 driver.close()
82 print('done')

```

ภาพประกอบที่ 3.17 ขั้นตอนการใช้งาน Python Web Scraping (ต่อ)

ซึ่งข้อมูลที่ได้มานั้นจะอยู่ใน File text ใน path ที่ได้กำหนดและนี่คือตัวอย่างของข้อมูลที่ได้มา



The image shows a Notepad window titled "SRO15_01_2023_16_13_01 - Notepad". The window contains a list of currency exchange rates. The data is as follows:

Currency	Rate 1	Rate 2	Rate 3	Rate 4
USD	32.60	33.00	100	100
USD	32.55	33.00	50	50
USD	32.45	32.90	10	20
USD	32.35	32.90	5	5
USD	32.00	32.80	1	2
EUR	35.20	35.70	100	500
EUR	35.15	35.70	50	50
EUR	35.10	35.70	5	20
GBP	39.90	40.45	50	50
GBP	39.80	40.35	5	20
CHF	35.05	35.40	10	1000
AUD	22.50	22.90	5	100
JPY	0.2560	0.2600	1000	10000

ภาพประกอบที่ 3.18 ขั้นตอนการใช้งาน Python Web ScraPi