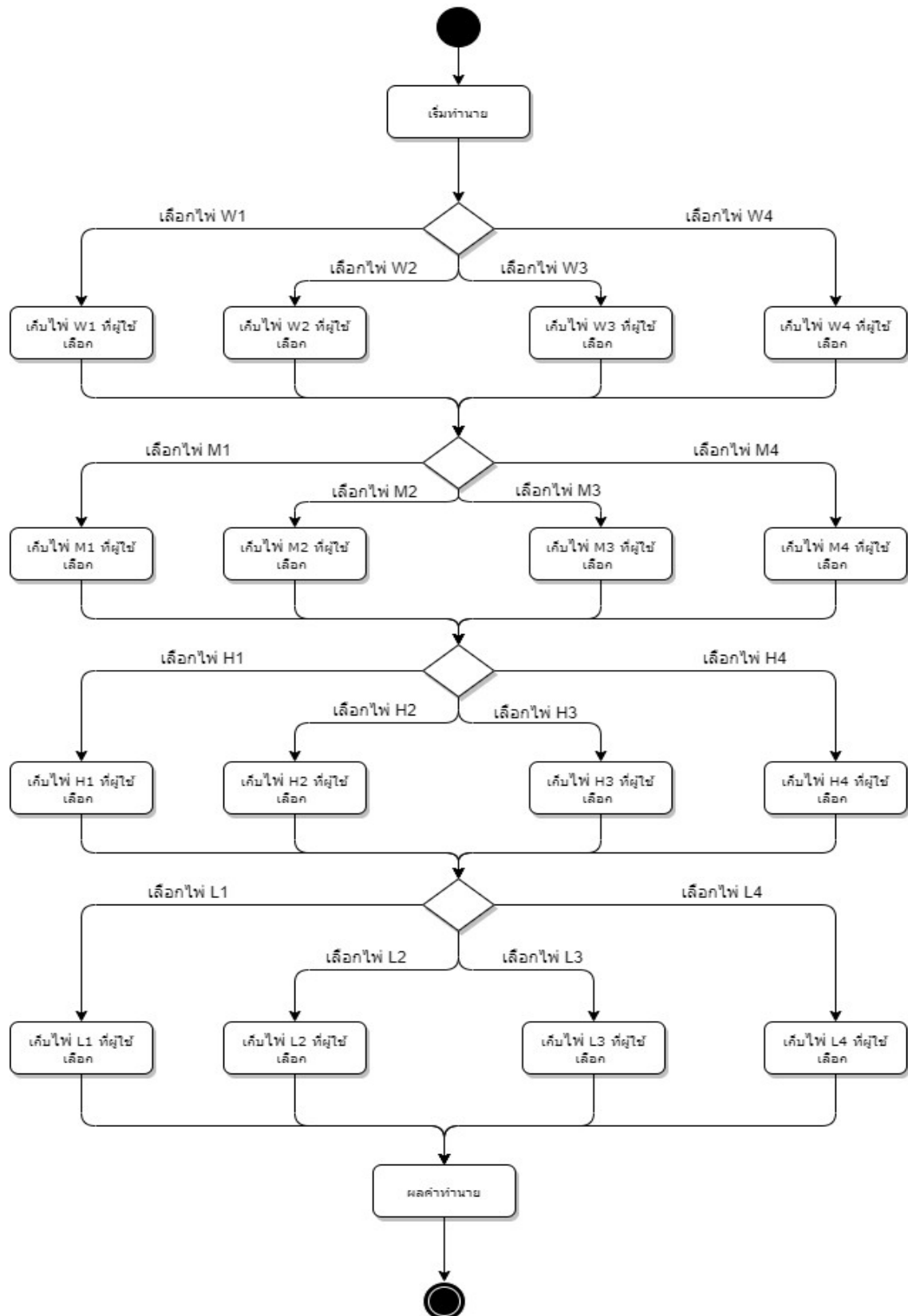


iii Activity Diagram

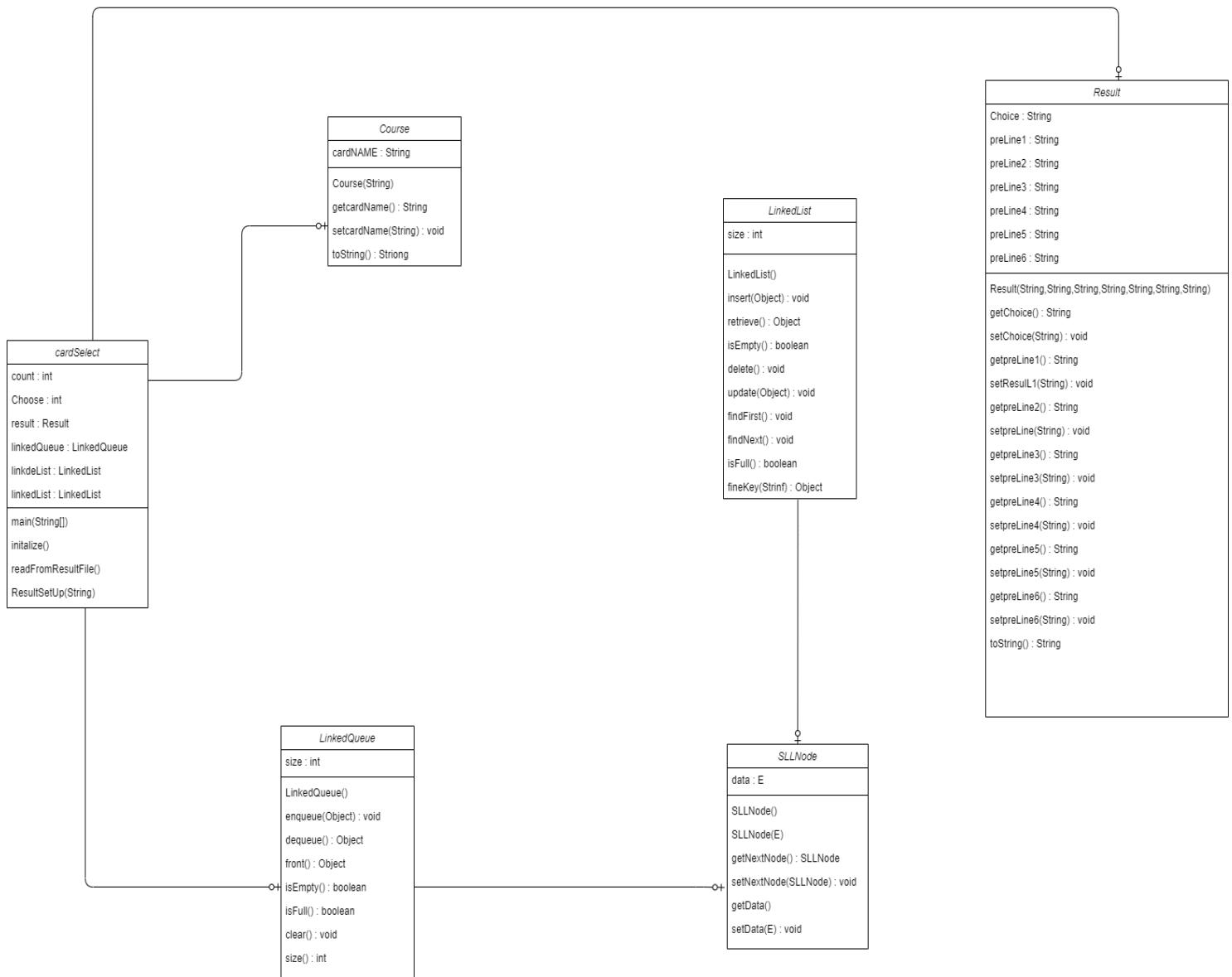


iv Use-Case Description

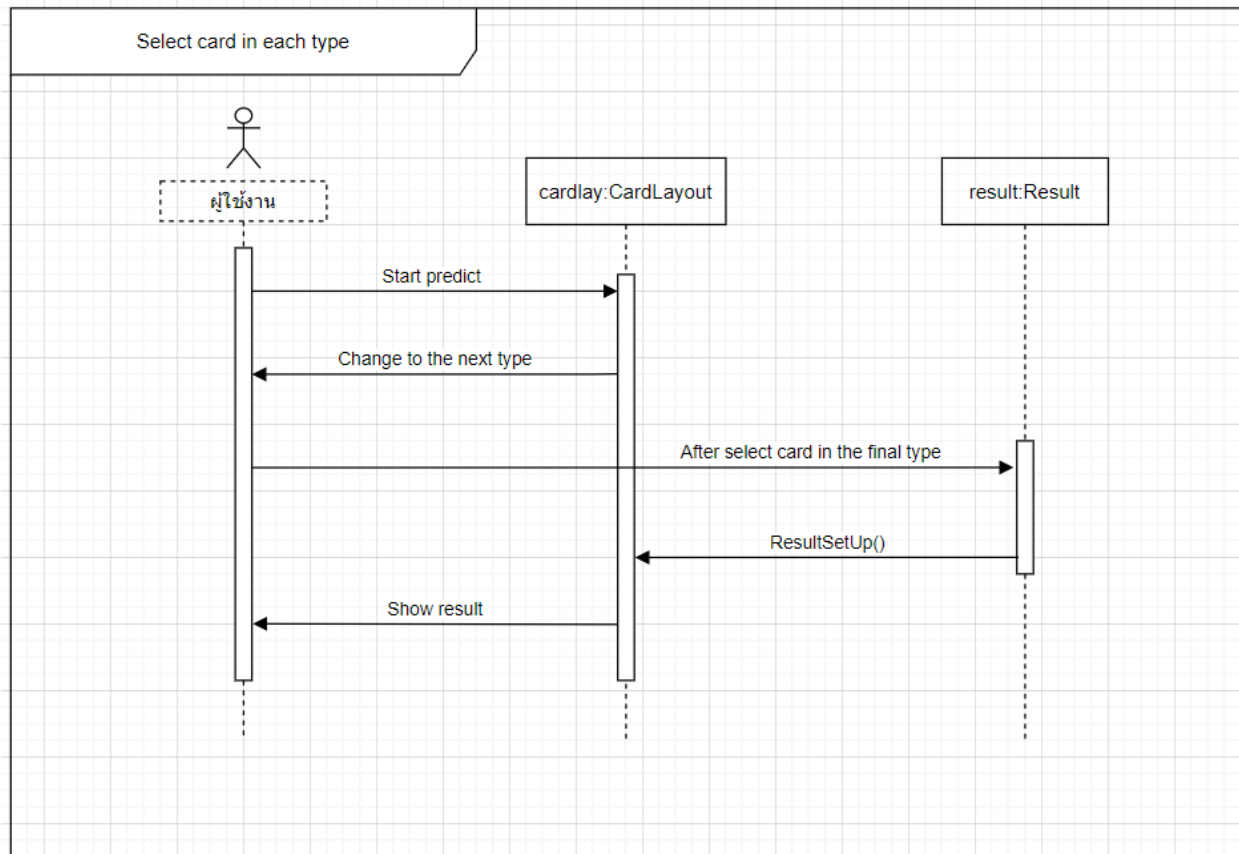
Use Case Name:	Select in each type
ID:	2
Importance Level:	Hight
Primary Actor:	ผู้ใช้งาน
Use Case Type:	Detail, Essential
Purpose:	เพื่อเลือกไฟในการทำนายดวงชะตา
Stakeholders and Interests:	ผู้ใช้งาน-ต้องการดูคำทำนายจากโปรแกรมเพื่อนำไป ประกอบการใช้ชีวิตอย่างระมัดระวัง
Brief Description:	เลือกไฟและแสดงผลคำทำนายจากไฟที่เลือก
Trigger:	ผู้ใช้เลือกไฟที่ต้องการ
Type:	External

Relationship:	<p>Association: ผู้ใช้งาน</p> <p>Include: Start predict, Show result</p> <p>Extend: -</p> <p>Generalization: -</p>
Normal Flow of Events:	<p>Actor</p> <p>1. ผู้ใช้งานเริ่มการนำนายดวงชะตาด้วยการเลือกไฟ</p> <ul style="list-style-type: none"> <li>- ไฟด้านการงาน</li> <li>- ไฟด้านการเงิน</li> <li>- ไฟด้านสุขภาพ</li> <li>- ไฟด้านความรัก</li> </ul> <p>Database System</p> <p>2. เก็บคำทำนายจากไฟแต่ละใบที่ผู้ใช้เลือก</p> <p>3. แสดงผลลัพธ์คำนายทั้งหมด</p>
Sub Flows:	-
Alternative/Exceptional Flows:	ระบบเก็บคำทำนายที่ผู้ใช้เลือกผิดพลาด

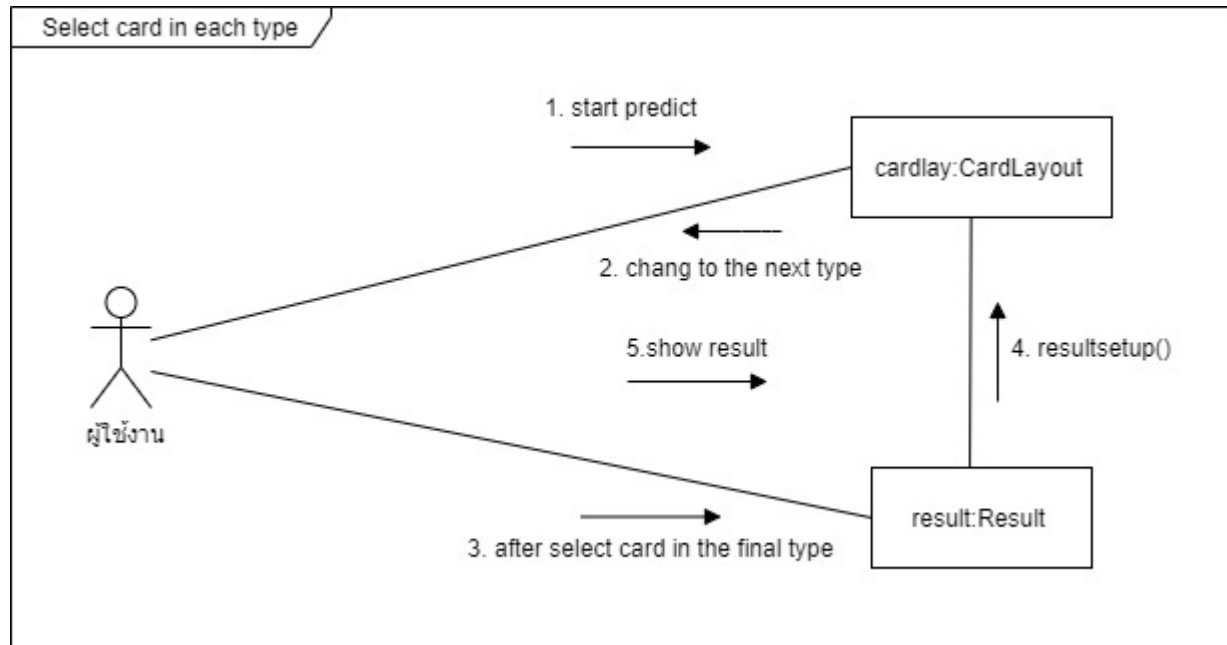
## v Analysis Class Diagram



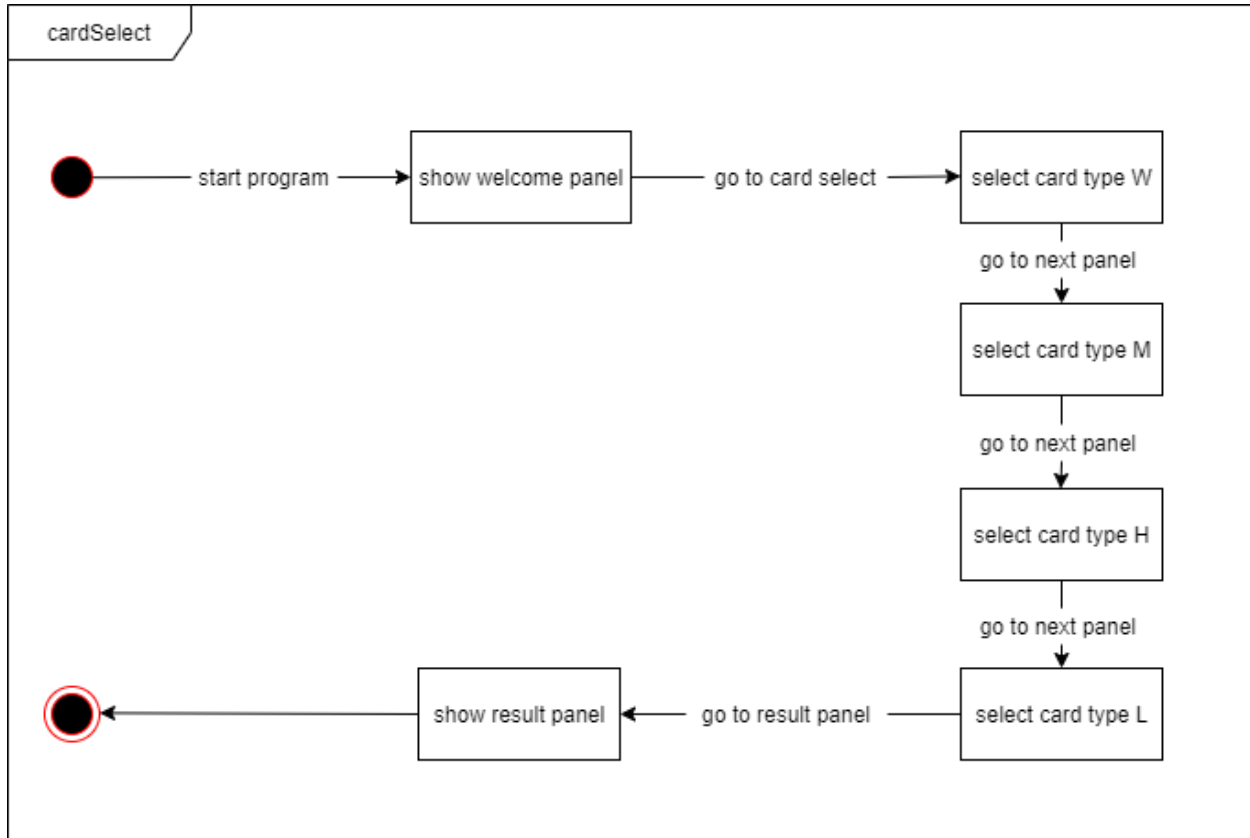
## vi Sequence Diagram



## vii Communication Diagram

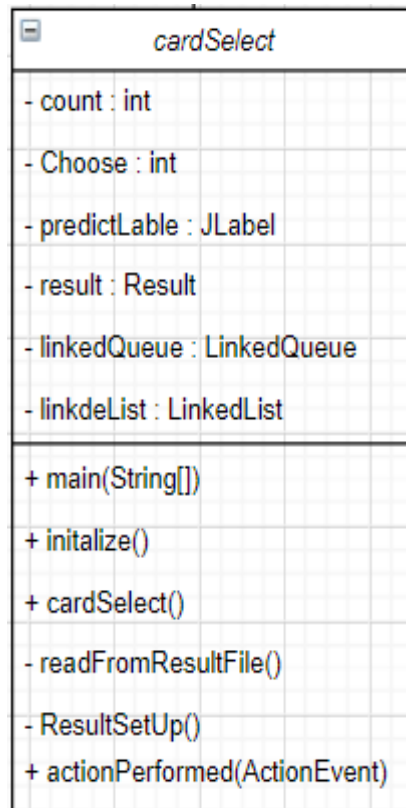


## viii Behavioral State Machine



#### 4. การออกแบบ (Design)

i Design Class Diagram



Method	การทำงาน
<code>main(String[])</code>	<code>cardSelect window = new cardSelect();</code> <code>window.J4.setVisible(true);</code>
<code>initialize()</code>	<code>J4 = new JFrame();</code> <code>J4.setTitle("ไม่ม่งมยทำนายดวงชะตา");</code> <code>J4.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);</code>
<code>cardSelect()</code>	<code>initialize();</code> <code>readFormResultFile();</code>

readFromResultFile()	<pre> File file = new File(path); br = new BufferedReader(new FileReader(file)); while ((Line = br.readLine()) != null) {     String[] Data = Line.split(",");     result = new Result(Data[0], Data[1], Data[2], Data[3],         Data[4], Data[5], Data[6]); </pre>
ResultSetUp()	<pre> if (key.equalsIgnoreCase("W1")) {     linkedlist.findFirst();     result = (Result)linkedlist.retrieve(); } else {     result = (Result)linkedlist.findKey(key); }if(key.equalsIgnoreCase("W1")  key.equalsIgnoreCase("W2")  key.eg ualsIgnoreCase("W3")  key.equalsIgnoreCase("W4")){     preWcard.setText(result.getChoice());     preW1.setText(result.getpreLine1());     preW2.setText(result.getpreLine2());     preW3.setText(result.getpreLine3());     preW4.setText(result.getpreLine4());     preW5.setText(result.getpreLine5());     preW6.setText(result.getpreLine6()); }else if (key.equalsIgnoreCase("M1")  key.equalsIgnoreCase("M2")  key.equa lsIgnoreCase("M3")  key.equalsIgnoreCase("M4")){     preMcard.setText(result.getChoice());     preM1.setText(result.getpreLine1());     preM2.setText(result.getpreLine2());     preM3.setText(result.getpreLine3());     preM4.setText(result.getpreLine4());     preM5.setText(result.getpreLine5());     preM6.setText(result.getpreLine6()); }else if </pre>



	<pre> (key.equalsIgnoreCase("H1"))  key.equalsIgnoreCase("H2"))  key.equalsIgnoreCase("H3"))  key.equalsIgnoreCase("H4")){     preHcard.setText(result.getChoice()); preH1.setText(result.getpreLine1()); preH2.setText(result.getpreLine2()); preH3.setText(result.getpreLine3());     preH4.setText(result.getpreLine4()); preH5.setText(result.getpreLine5()); preH6.setText(result.getpreLine6()); }else if (key.equalsIgnoreCase("L1"))  key.equalsIgnoreCase("L2"))  key.equalsIgnoreCase("L3"))  key.equalsIgnoreCase("L4")){     preLcard.setText(result.getChoice()); preL1.setText(result.getpreLine1()); preL2.setText(result.getpreLine2()); preL3.setText(result.getpreLine3()); preL4.setText(result.getpreLine4()); preL5.setText(result.getpreLine5()); preL6.setText(result.getpreLine6()); } </pre>
actionPerformed()	<pre> int num = 0; while (num &lt; 6) {     if(num == 0) {         num++;         cardlay.next(panel);     }      Course text = new Course(buttonText);     num++;     queue.enqueue(text); }  for (int i = 0; i &lt; 5; i++) { </pre>

	<pre>course = (Course)queue.dequeue(); ResultSetUp(course.getcardName()); }</pre>
--	---

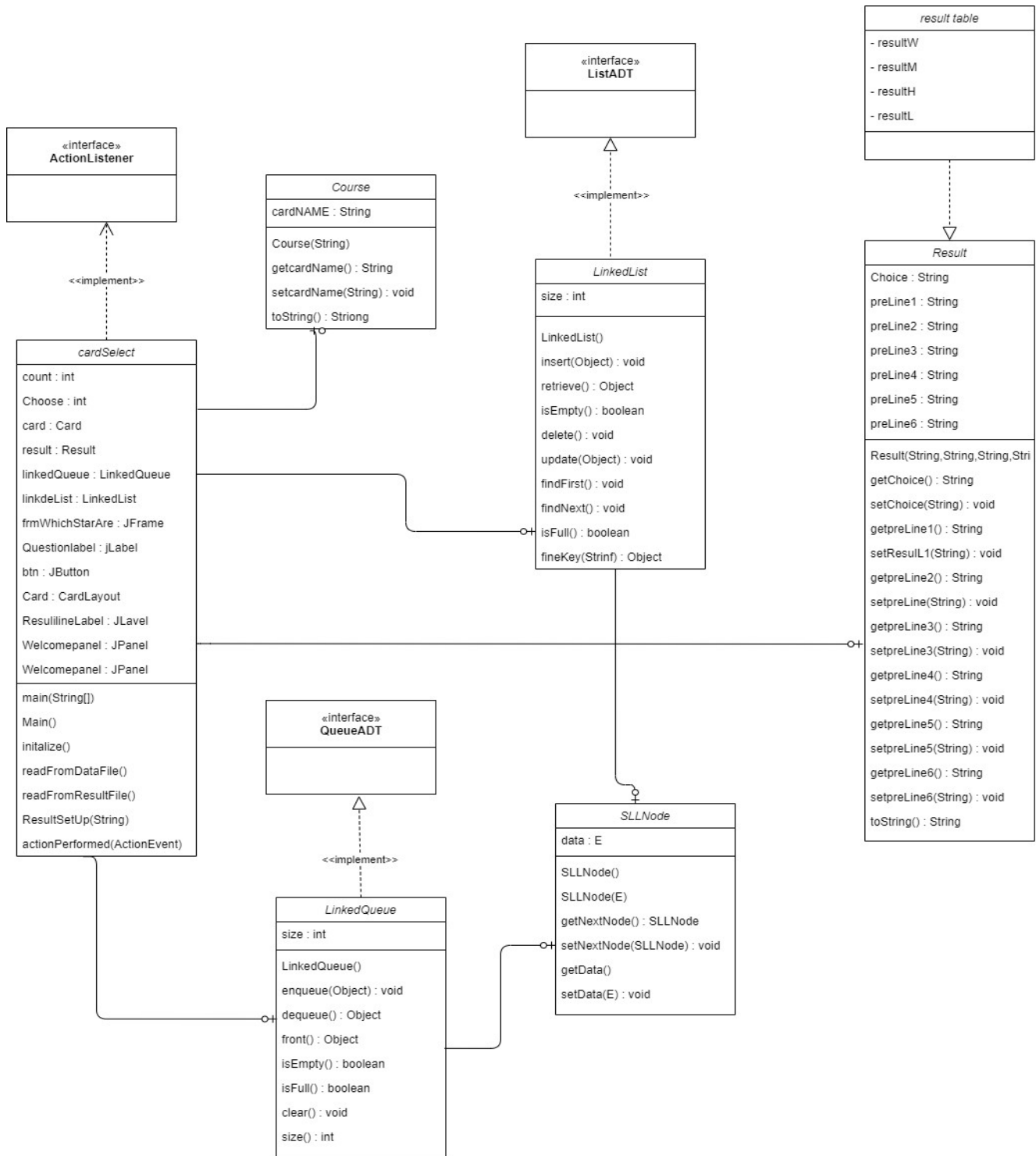
ii Method Contract

Method Name: <code>insert()</code>	Class Name: <code>LinkedList</code>	ID: <code>02</code>
Clients (Consumers): <code>Main</code>		
Associated Use Cases:		
Description of Responsibilities: คอยทำหน้าที่เก็บข้อมูลไฟที่ผู้ใช้เลือกเป็น List ไว้ เพื่อนำมา แสดงออกเป็นผลลัพธ์ภายหลัง		
Arguments Received: <code>ResultSetUp</code>		
Type of Value Returned: <code>void</code>		
Pre-conditions: <code>None</code>		
Post-conditions: <code>None</code>		

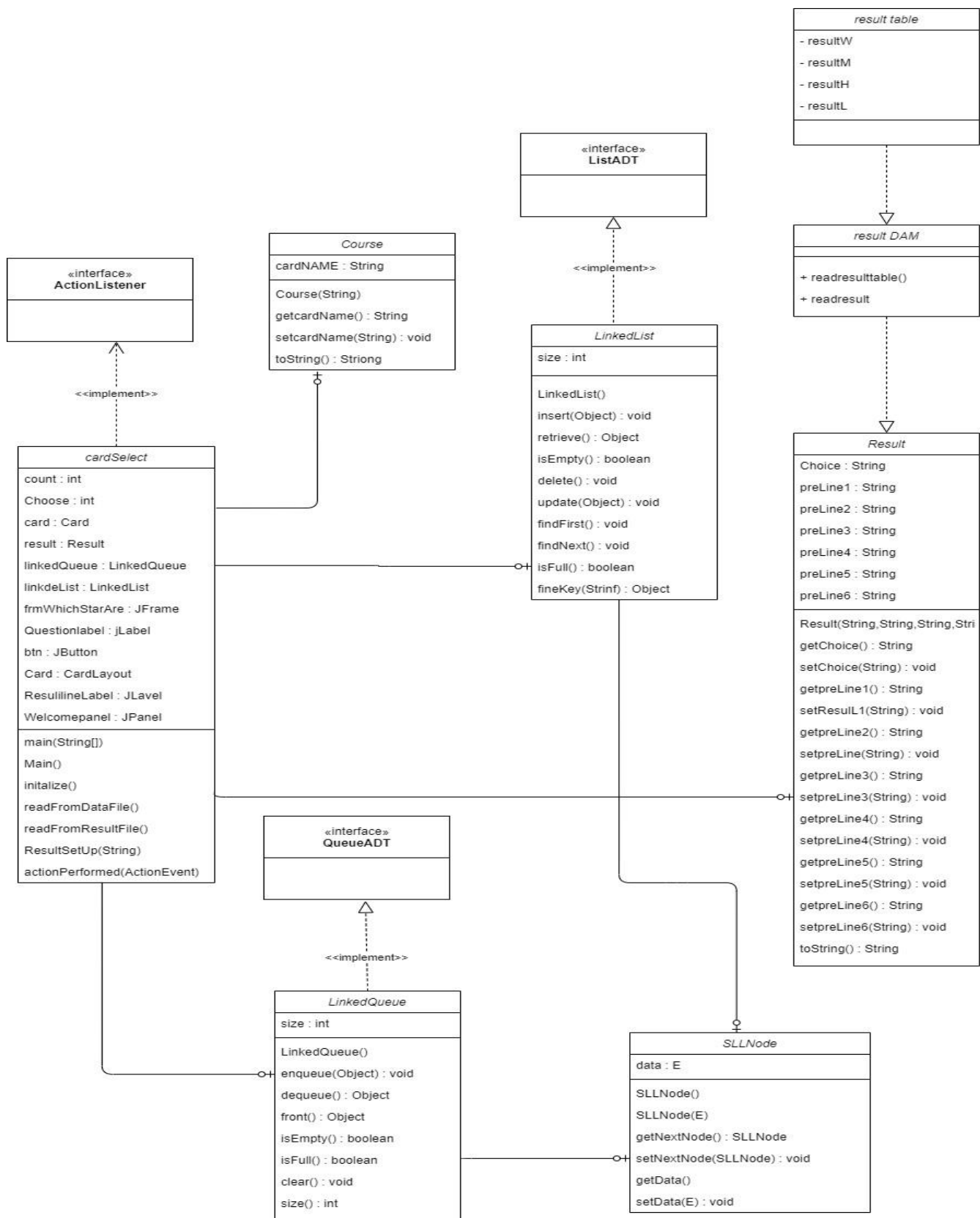
### iii Method Specification

Method Name: <code>insert()</code>	Class Name: <code>LinkedList</code>	ID: <code>02</code>
Contract ID: <code>138,426,430</code>	Programmer: <code>FourSeason</code>	Date Due: <code>25/04/2021</code>
Programming Language: <input type="checkbox"/> Visual Basic <input type="checkbox"/> Smalltalk <input type="checkbox"/> C++ <input checked="" type="checkbox"/> Java		
Triggers/Events: เมื่อผู้ใช้ทำการกดเลือกไฟ		
Arguments Received: Data Types:	Notes:	
<code>object</code>	<code>String</code> ของคำทำนายจากไฟที่ผู้ใช้เลือก	
Messages Sent & Arguments Passed: ClassName.MethodName:	Data Type:	Notes:
<code>linkedList.findFirst()</code>		
<code>linkedList.retrieve()</code>		
<code>linkedList.findKey(key)</code>		
Arguments Returned: Data Type:	Notes:	
<code>void</code>		
<b>Algorithm Specification:</b> <pre>         if (key.equalsIgnoreCase("W1")) {              linkedlist.findFirst();              result = (Result)linkedlist.retrieve();          } else {              result = (Result)linkedlist.findKey(key);          }       </pre>		
Misc. Notes: <code>None</code>		

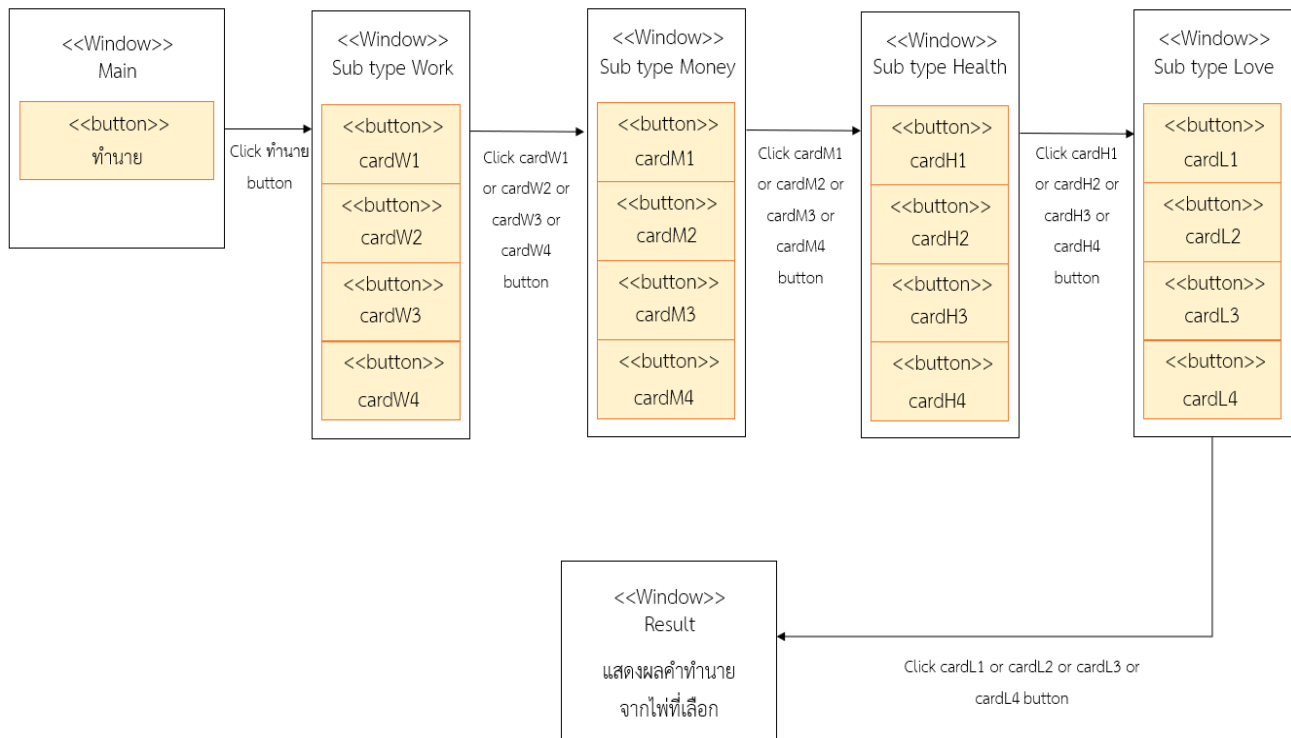
#### iv Data Base: Mapping Class Diagram



## Data Base: Data Access and Manipulation Classes



## v User Interface



## 5. บทสรุป

จากการทำโครงการนี้สิ่งที่ได้เรียนรู้ คือ การออกแบบและนำมาต่อยอดจากแอปพลิเคชันที่มีอยู่ให้มีความสมบูรณ์มากขึ้น อุปสรรคของการวิเคราะห์และออกแบบ อยู่ที่ส่วนของงานบางหัวข้อต้องเรียนรู้หลายรอบเพื่อความละเอียดในการเขียน และหัวข้อบางอย่างที่คิดไว้พอออกแบบแล้วดูไม่มีความเหมาะสมจึงจำเป็นต้องตัดออก นอกจากนั้นในกลุ่มมีการจัดการเวลาให้ตรงกันยากเนื่องจากการระบาดของเชื้อโควิด-19 สมาชิกในกลุ่มมีงานที่ต้องทำคนละกลุ่มทำให้เจอกันยาก ส่งผลให้การพูดคุย ทำงาน ประชุมเนื้อหาเป็นได้ยากและไม่สะดวก

## 6. อ้างอิง

[1] อ.สิงห์โต สุริยาอารักษ์. (2003). **โหราเวทย์**. (สืบค้นเมื่อ 21 เมษายน 2564).

<http://www.horawej.com/index.php?lay=show&ac=article&id=189124&Ntype=34>