



LIBRARY SYSTEM

ระบบห้องสมุด 2018

King Mongkut's University of Technology north Bangkok
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

สารบัญ

เนื้อหา

Library	System
	ผิด
พลาด! ไม่ได้กำหนดบุ๊กมาร์ก	
Library System - Supplementary Specifications	2
Story Board ระบบ Library	3
Use Case Diagram ระบบ Library	4
Glossary	5
Use-Case Specification << Check Member >>	6
Use-Case Specification << Log In >>	8
Use-Case Specification << Borrow >>	10
Use-Case Specification << Save to Profile >>	12
Use-Case Specification << Check Profile Member >>	14
Use-Case Specification << Return>>	16
Use-Case Specification << Check Period >>	18
Use-Case Specification << Calculation Mulct >>	20
Analysis Class Diagram ระบบ Library	22

สารบัญ

Class Diagram ระบบ Library	23
Sequence Diagram การยืมหนังสือ	24
Sequence Diagram การคืนหนังสือ	25
Activity การยืมหนังสือ	26
Activity การคืนหนังสือ	27
State Diagram for Book	28
Collaboration Diagram for borrow Book	29
Collaboration Diagram for returning Book	30
Code	31
รายชื่อผู้จัดทำ	69

สารบัญ

LIBRARY SYSTEM

Library System หรือ ระบบห้องสมุด คือระบบจัดการห้องสมุดที่สามารถอำนวยความสะดวกให้กับผู้ใช้บริการห้องสมุด ซึ่งเป็นแหล่งสารสนเทศ บริการทรัพยากรสารสนเทศในรูปแบบต่างๆ เช่น หนังสือ วารสาร หนังสือพิมพ์ จุลสาร กฤตภาค วัสดุเทป และโทรทัศน์ ซีดีรอม วีซีดี ดีวีดี โดยมีบรรณารักษ์เป็นผู้ดำเนินงาน และบริหารงานต่างๆ ในห้องสมุด โดยจัดระบบเป็นหมวดหมู่ และระเบียบเรียบร้อย เพื่อให้ผู้ใช้ห้องสมุดมีความสะดวกสืบค้นได้ง่าย และตรงกับความต้องการ ซึ่งระบบห้องสมุดสามารถให้บริการผู้ใช้บริการได้ดังต่อไปนี้

ผู้ใช้บริการสามารถสมัครสมาชิกเพื่อเข้าสู่ระบบในการทำรายการยืม คืน หนังสือ โดยจะต้องกรอกรายละเอียด ชื่อ นามสกุล รหัส ระบบจะแสดงข้อมูล หากผู้ใช้บริการต้องการสมัครสมาชิกให้กดยืนยันเพื่อทำการ หรือยกเลิกการทำรายการ ตามต้องการใช้บริการสามารถเข้าสู่ระบบเพื่อตรวจสอบประวัติการยืมหนังสือ จากนั้นระบบจะแสดงข้อมูลการยืมหนังสือทั้งหมด รวมทั้งระยะเวลาที่ยืม ที่ผู้ใช้บริการยืมจากห้องสมุดไป

ผู้ใช้บริการสามารถเข้าสู่ระบบเพื่อทำการยืมหนังสือจากห้องสมุดได้ โดยเลือกรายการหนังสือที่ต้องการจะยืม ระบบจะแสดงข้อมูล หากผู้ใช้บริการต้องการทำการยืมให้กดยืนยันเพื่อทำการ หรือยกเลิกการทำรายการ ตามต้องการ

ผู้ใช้บริการสามารถเข้าสู่ระบบเพื่อทำการคืนหนังสือได้ โดยเลือกรายการจากหนังสือที่ต้องการจะคืน ระบบจะแสดงข้อมูล ระยะเวลาทั้งหมดที่ยืม รวมถึงหากยืมเกินระยะเวลาดังแสดงค่าปรับ หากผู้ใช้บริการต้องการคืนหนังสือให้กดยืนยันเพื่อทำการ หรือยกเลิกการทำรายการ ตามต้องการ

LIBRARY SYSTEM

Library System - Supplementary Specifications

1. ความง่ายของการใช้งาน (Usability)

- ผู้ใช้งานที่ไม่มีประสบการณ์สามารถทำความเข้าใจกับระบบและใช้งานระบบได้ไม่เกิน 15 นาที
- ผู้ใช้งานที่มีประสบการณ์สามารถทำความเข้าใจกับระบบและใช้งานระบบได้ไม่เกิน 5 นาที

2. ความคงเส้นคงวา (Reliability)

- ระบบสามารถรับรองการทำงานได้ตลอดเวลาห้องสมุดเปิดทำการ

3. ประสิทธิภาพในการทำงาน (Performance)

- ระบบสามารถทำการเสร็จสมบูรณ์ภายใน 5 นาที

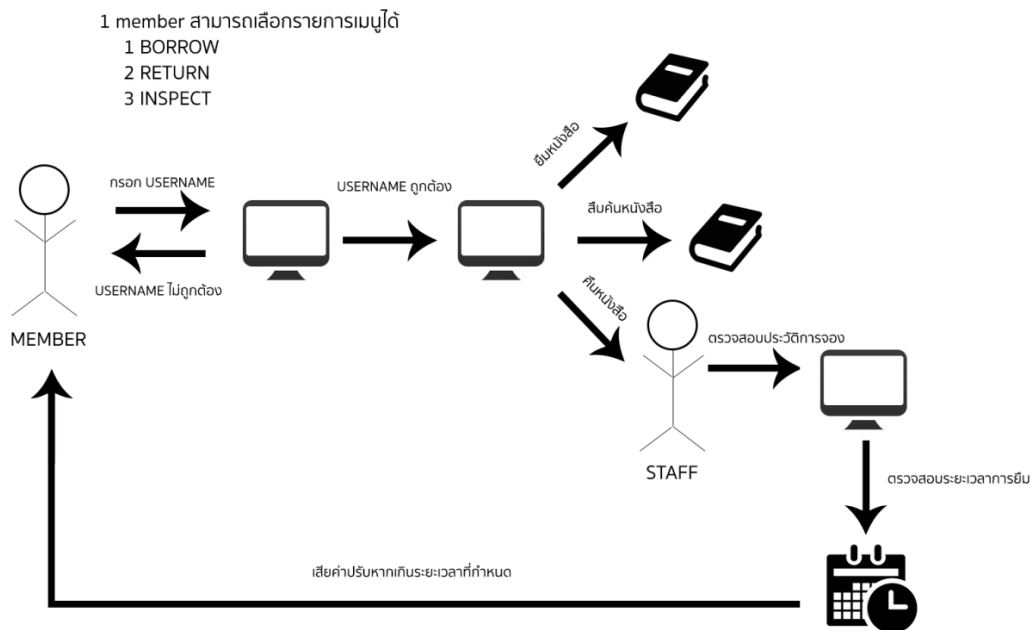
4. ความต้องการสนับสนุน (Supportability)

- สามารถทำงานได้เฉพาะระบบปฏิบัติการ Windows
- คอมพิวเตอร์ที่เชื่อมต่อกับเครือข่ายภายในของห้องสมุด

5. ข้อจำกัด/กรอบของการออกแบบ (Design constraints)

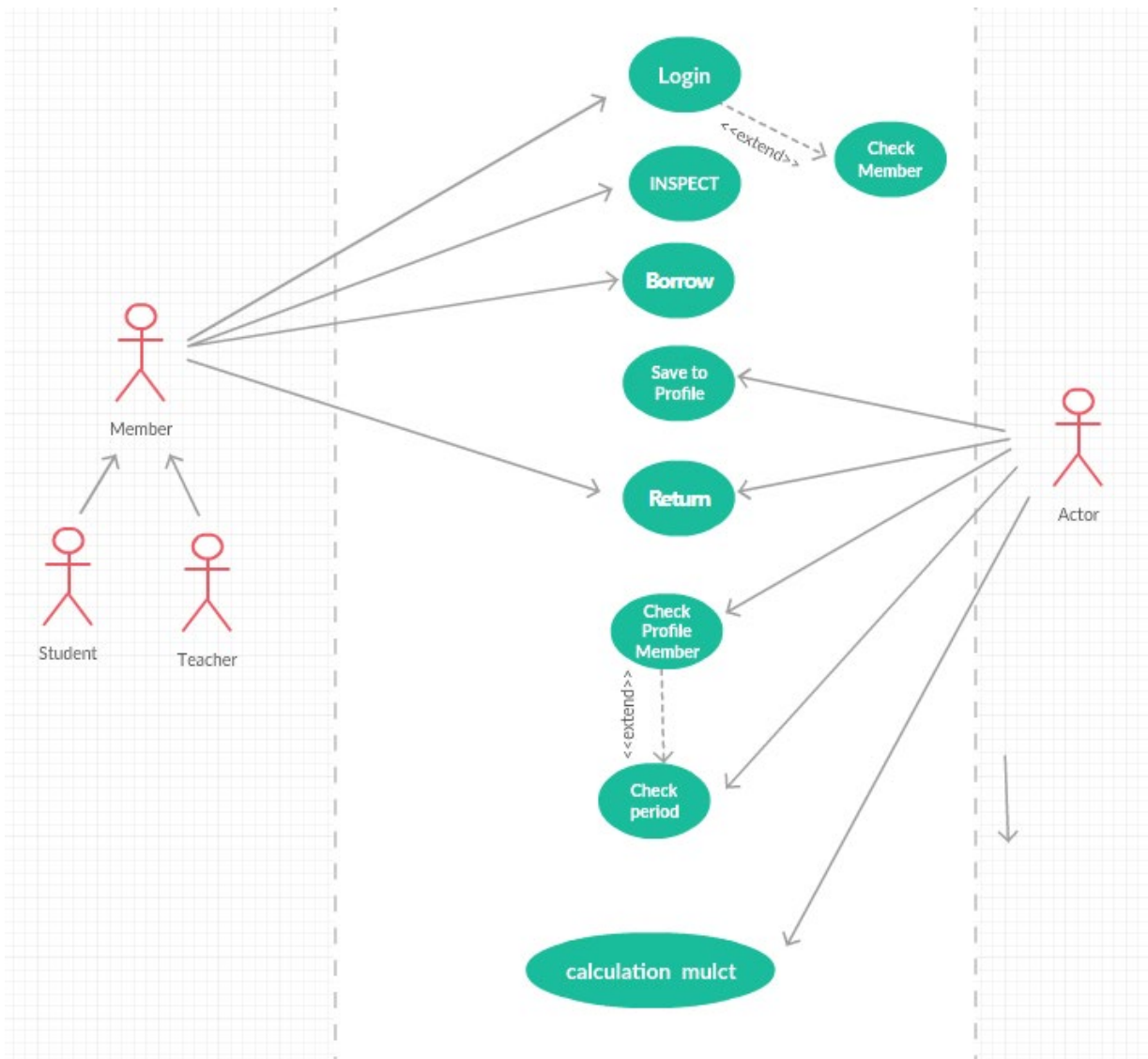
- ระบบจะทำการติดต่อสื่อสารกับผู้ใช้งานผ่าน window-based desktop เท่านั้น

LIBRARY SYSTEM



Storyboard ระบบ Library

LIBRARY SYSTEM



Use Case Diagram ๙๕๒๒ Library

LIBRARY SYSTEM

Glossary

การทำงานโดยย่อ (Brief Description)

Glossary ประกอบด้วยคำนิยามต่างๆ ที่ใช้ในห้องสมุด เพื่อช่วยให้ง่ายในการทำ
เข้าใจการทำงานของระบบห้องสมุดโดยรวม

คำนิยาม (Terms)

Member	คือผู้ใช้งานห้องสมุด
Librarian	คือผู้ดูแลห้องสมุด
Borrow	คือการยืมหนังสือในห้องสมุด
Return	คือการคืนหนังสือที่ยืมมา
Check Member	เป็นการตรวจสอบข้อมูลของผู้ยืม
Check Book	เป็นการตรวจสอบข้อมูลหนังสือที่ยืม
Check Period	เป็นการตรวจสอบระยะเวลาที่ยืม
Calculation Mulct	เป็นการตรวจสอบค่าธรรมเนียมการยืมหนังสือ
Check Profile Member	เป็นการตรวจสอบประวัติการยืมหนังสือของสมาชิก

LIBRARY SYSTEM

Use-Case Specification

<< Check Member >>

1. การทำงานโดยย่อของยูสเคส (Brief Description)

- เริ่มต้นโดยผู้ใช้งานกรอกข้อมูลลงในหน้า LOG IN แล้ว

2. ลำดับเหตุการณ์ (Flow Events)

2.1 เหตุการณ์ปกติ (Main Flows)

- มีชื่อผู้ใช้งานในระบบ

2.2 เหตุการณ์ไม่ปกติ (Alternate Flows)

- ไม่พบชื่อผู้ใช้งาน อาจเกิดจาก กรอก username หรือ password ผิด จะแจ้งให้กรอกใหม่

3. ความสัมพันธ์ (Associations)

3.1 Actors

- Actors ที่เริ่มยูสเคสนี้
- Member
- Actors ที่สัมพันธ์กับยูสเคส
- ไม่มี

LIBRARY SYSTEM

3.2 ความสัมพันธ์กับยูสเคสอื่น

- ไม่มี

4. เงื่อนไขก่อนทำงานของยูสเคส (Pre-Conditions)

- ต้องผ่านการ Log In ก่อน

5. เงื่อนไขหลังทำงานของยูสเคส (Post-Conditions)

- ไม่มี

6. ความต้องการอื่นๆ (Special Requirement)

- ไม่มี

LIBRARY SYSTEM

Use-Case Specification

<< Log In >>

1. การทำงานโดยย่อของยูสเคส (Brief Description)

- เริ่มต้นโดยผู้ใช้งานกรอกข้อมูลลงในหน้า Log In

2. ลำดับเหตุการณ์ (Flow Events)

2.1 เหตุการณ์ปกติ (Main Flows)

- กรอก Username และ Password

2.2 เหตุการณ์ไม่ปกติ (Alternate Flows)

- ระบบไม่สามารถทำการต่อได้ ถ้าผู้ใช้ไม่กรอก

3. ความสัมพันธ์ (Associations)

3.1 Actors

- Actors ที่เริ่มยูสเคสนี้
- Member
- Actors ที่สัมพันธ์กับยูสเคส
- ไม่มี

3.2 ความสัมพันธ์กับยูสเคสอื่น

- ไม่มี

4. เงื่อนไขก่อนทำงานของยูสเคส (Pre-Conditions)

- ไม่มี

5. เงื่อนไขหลังทำงานของยูสเคส (Post-Conditions)

- ไม่มี

6. ความต้องการอื่นๆ (Special Requirement)

- ไม่มี

LIBRARY SYSTEM

Use-Case Specification

<< Borrow >>

1. การทำงานโดยย่อของยูสเคส (Brief Description)

- เริ่มต้นโดยผู้ใช้งานต้องการที่จะยืมหนังสือเล่มใดเล่มหนึ่ง

2. ลำดับเหตุการณ์ (Flow Events)

2.1 เหตุการณ์ปกติ (Main Flows)

- ยืมหนังสือได้

2.2 เหตุการณ์ไม่ปกติ (Alternate Flows)

- หนังสือที่ต้องการไม่มี

3. ความสัมพันธ์ (Associations)

3.1 Actors

- Actors ที่เริ่มยูสเคสนี้
- Member
- Actors ที่สัมพันธ์กับยูสเคส
- ไม่มี

3.2 ความสัมพันธ์กับยูสเคสอื่น

- ไม่มี

4. เงื่อนไขก่อนทำงานของยูสเคส (Pre-Conditions)

- ต้องผ่านการ Log In ก่อน

5. เงื่อนไขหลังทำงานของยูสเคส (Post-Conditions)

- ไม่มี

6. ความต้องการอื่นๆ (Special Requirement)

- ไม่มี

LIBRARY SYSTEM

Use-Case Specification

<< Save to Profile >>

1. การทำงานโดยย่อของยูสเคส (Brief Description)

- เริ่มต้นโดยผู้ใช้งาน Borrow หรือ Return

2. ลำดับเหตุการณ์ (Flow Events)

2.1 เหตุการณ์ปกติ (Main Flows)

- บันทึกข้อมูลการยืม หรือ คืน ลงใน Profile ของผู้ใช้

2.2 เหตุการณ์ไม่ปกติ (Alternate Flows)

- ไม่มี

3. ความสัมพันธ์ (Associations)

3.1 Actors

- Actors ที่เริ่มยูสเคสนี้
- Member
- Actors ที่สัมพันธ์กับยูสเคส
- Library

3.2 ความสัมพันธ์กับยูสเคสอื่น

- Borrow และ Return

4. เงื่อนไขก่อนทำงานของยูสเคส (Pre-Conditions)

- ต้องผ่านการ Borrow และ Return ให้เสร็จก่อน

5. เงื่อนไขหลังทำงานของยูสเคส (Post-Conditions)

- ไม่มี

6. ความต้องการอื่นๆ (Special Requirement)

- ไม่มี

LIBRARY SYSTEM

Use-Case Specification

<< Check Profile Member >>

1. การทำงานโดยย่อของยูสเคส (Brief Description)

- เริ่มต้นโดยผู้ใช้งาน ต้องการที่จะ ค้นหาหนังสือ

2. ลำดับเหตุการณ์ (Flow Events)

2.1 เหตุการณ์ปกติ (Main Flows)

- พบหนังสือที่ผู้ใช้ค้น

2.2 เหตุการณ์ไม่ปกติ (Alternate Flows)

- ไม่พบหนังสือที่ผู้ใช้ค้น

3. ความสัมพันธ์ (Associations)

3.1 Actors

- Actors ที่เริ่มยูสเคสนี้
- Library
- Actors ที่สัมพันธ์กับยูสเคส
- Member

LIBRARY SYSTEM

3.2 ความสัมพันธ์กับยูสเคสอื่น

- ไม่มี

4. เงื่อนไขก่อนทำงานของยูสเคส (Pre-Conditions)

- ต้องผ่านการ Borrow และ Log In

5. เงื่อนไขหลังทำงานของยูสเคส (Post-Conditions)

- จะทำการเช็คระยะเวลา และ ค่าปรับ

6. ความต้องการอื่นๆ (Special Requirement)

- ไม่มี

LIBRARY SYSTEM

Use-Case Specification

<< Return >>

1. การทำงานโดยย่อของยูสเคส (Brief Description)

- เริ่มต้นโดยผู้ใช้งาน ต้องการที่จะ คำนวณหนังสือ

2. ลำดับเหตุการณ์ (Flow Events)

2.1 เหตุการณ์ปกติ (Main Flows)

- Login และเลือก คำนวณหนังสือ

2.2 เหตุการณ์ไม่ปกติ (Alternate Flows)

- ไม่มี

3. ความสัมพันธ์ (Associations)

3.1 Actors

- Actors ที่เริ่มยูสเคสนี้
- Member
- Actors ที่สัมพันธ์กับยูสเคส
- Library

LIBRARY SYSTEM

3.2 ความสัมพันธ์กับยูสเคสอื่น

- ไม่มี

4. เงื่อนไขก่อนทำงานของยูสเคส (Pre-Conditions)

- ต้องผ่านการ Borrow และ Log In

5. เงื่อนไขหลังทำงานของยูสเคส (Post-Conditions)

- จะทำการเช็คระยะเวลา และ ค่าปรับ

6. ความต้องการอื่นๆ (Special Requirement)

- ไม่มี

LIBRARY SYSTEM

Use-Case Specification

<< Check Period >>

1. การทำงานโดยย่อของยูสเคส (Brief Description)

- เมื่อคืนหนังสือแล้ว

2. ลำดับเหตุการณ์ (Flow Events)

2.1 เหตุการณ์ปกติ (Main Flows)

- ผู้ใช้คืนหนังสือตรงเวลา

2.2 เหตุการณ์ไม่ปกติ (Alternate Flows)

- ผู้ใช้ยืมหนังสือเลยระยะเวลาที่กำหนด

3. ความสัมพันธ์ (Associations)

3.1 Actors

- Actors ที่เริ่มยูสเคสนี้
- Library
- Actors ที่สัมพันธ์กับยูสเคส

3.2 ความสัมพันธ์กับยูสเคสอื่น

- ไม่มี

4. เงื่อนไขก่อนทำงานของยูสเคส (Pre-Conditions)

- ต้องผ่านการ Borrow และ Log In

5. เงื่อนไขหลังทำงานของยูสเคส (Post-Conditions)

- ถ้าเลยระยะเวลาที่กำหนดจะ คิดค่าปรับแต่ถ้าไม่ จะบันทึกลงในข้อมูลของผู้ใช้

6. ความต้องการอื่นๆ (Special Requirement)

- ไม่มี

LIBRARY SYSTEM

Use-Case Specification

<< Calculation Mulet >>

1. การทำงานโดยย่อของยูสเคส (Brief Description)

- เมื่อคืนหนังสือเลยระยะเวลาที่กำหนด

2. ลำดับเหตุการณ์ (Flow Events)

2.1 เหตุการณ์ปกติ (Main Flows)

- คำนวณค่าปรับ

2.2 เหตุการณ์ไม่ปกติ (Alternate Flows)

- ไม่มี

3. ความสัมพันธ์ (Associations)

3.1 Actors

- Actors ที่เริ่มยูสเคสนี้
- Library
- Actors ที่สัมพันธ์กับยูสเคส
- Member

3.2 ความสัมพันธ์กับยูสเคสอื่น

- ไม่มี

4. เงื่อนไขก่อนทำงานของยูสเคส (Pre-Conditions)

- ต้องผ่านการ Borrow และ Log In

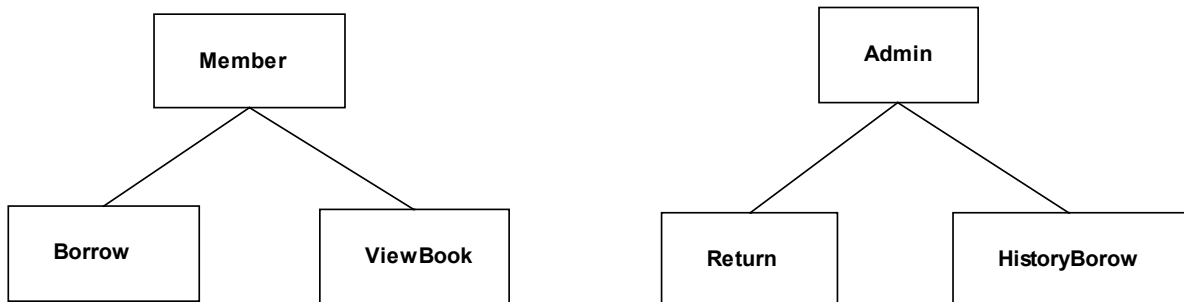
5. เงื่อนไขหลังทำงานของยูสเคส (Post-Conditions)

- เมื่อคืนเสร็จแล้วจะบันทึกลงในข้อมูลของผู้ใช้

6. ความต้องการอื่นๆ (Special Requirement)

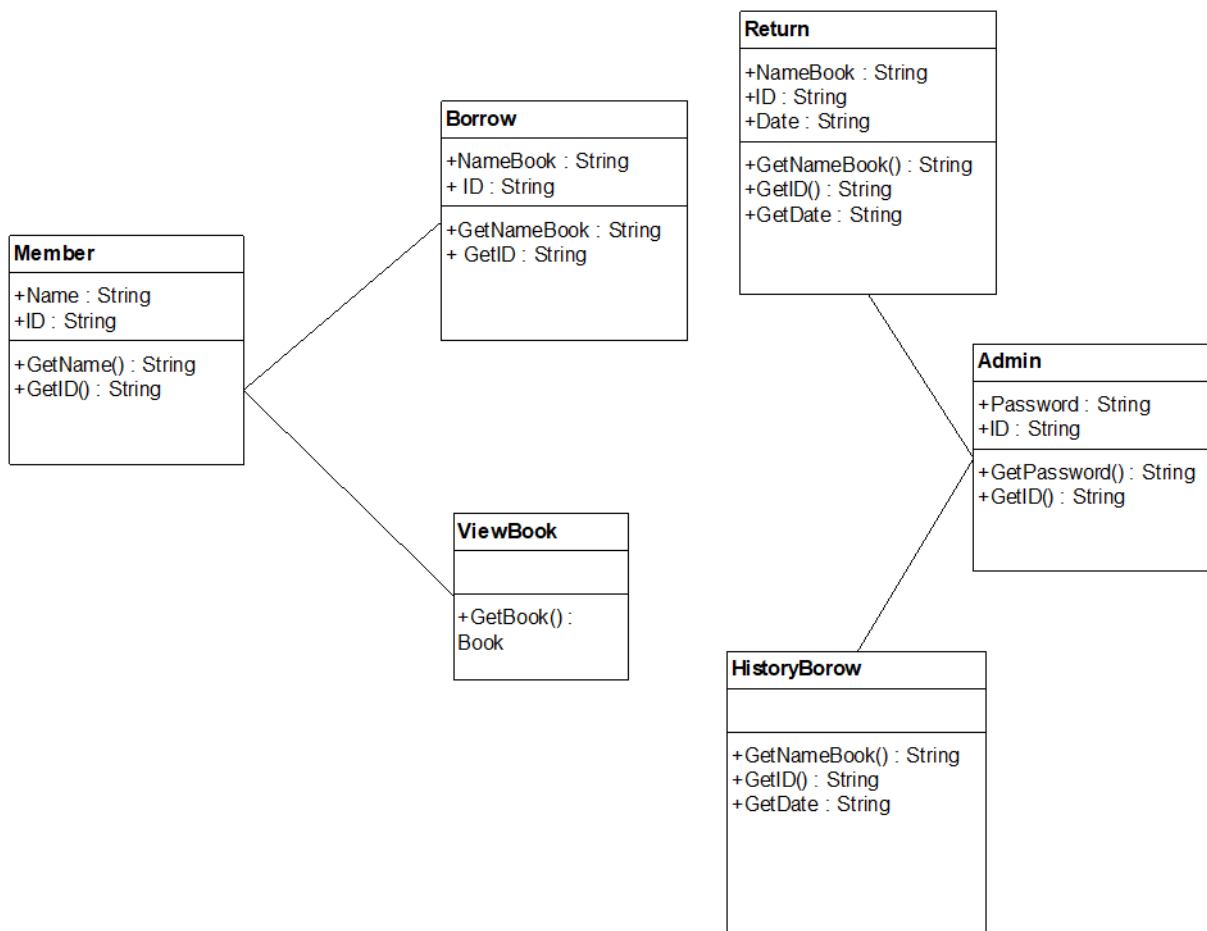
- ไม่มี

LIBRARY SYSTEM



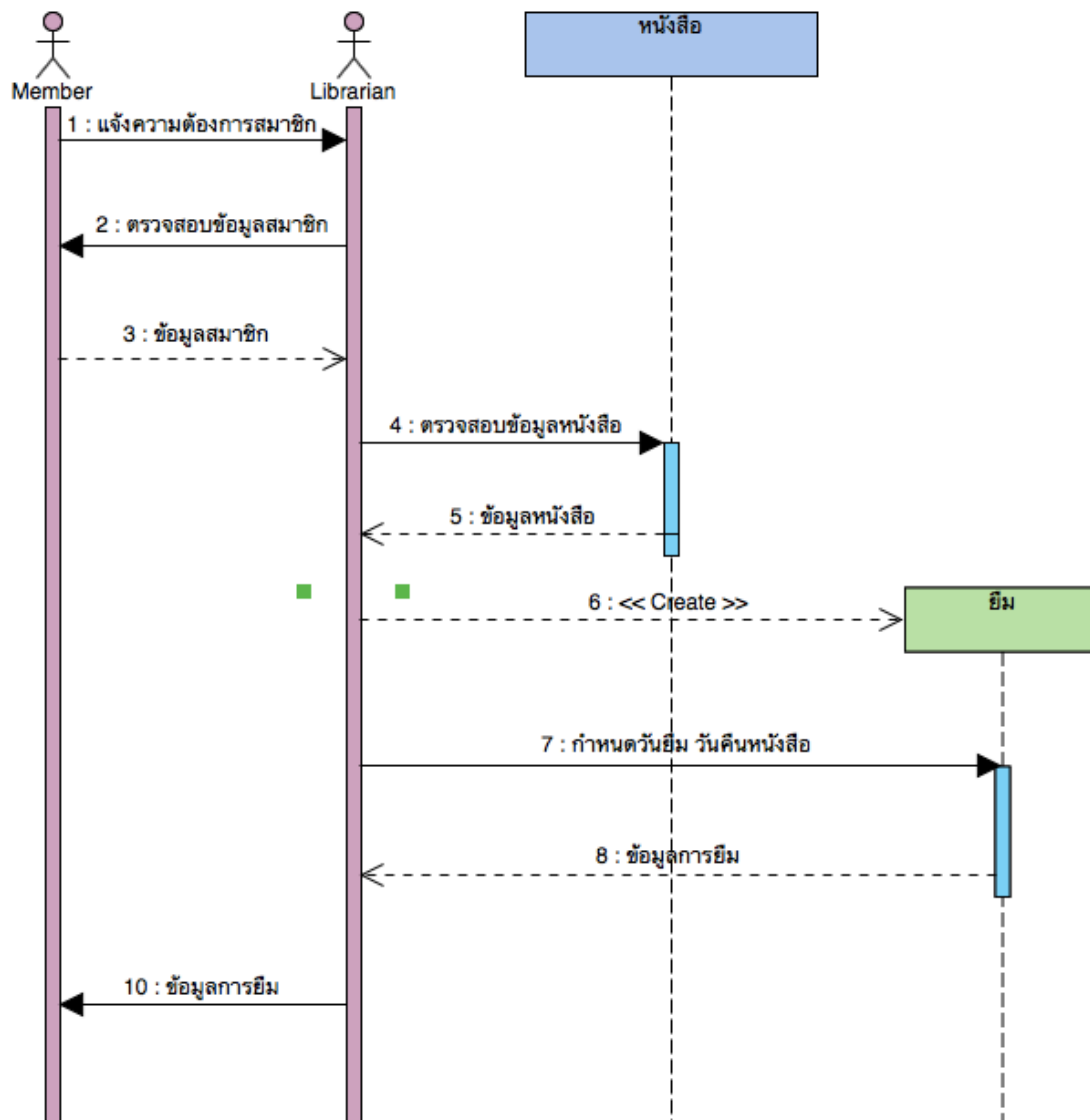
Analysis Class Diagram 7211 Library

LIBRARY SYSTEM



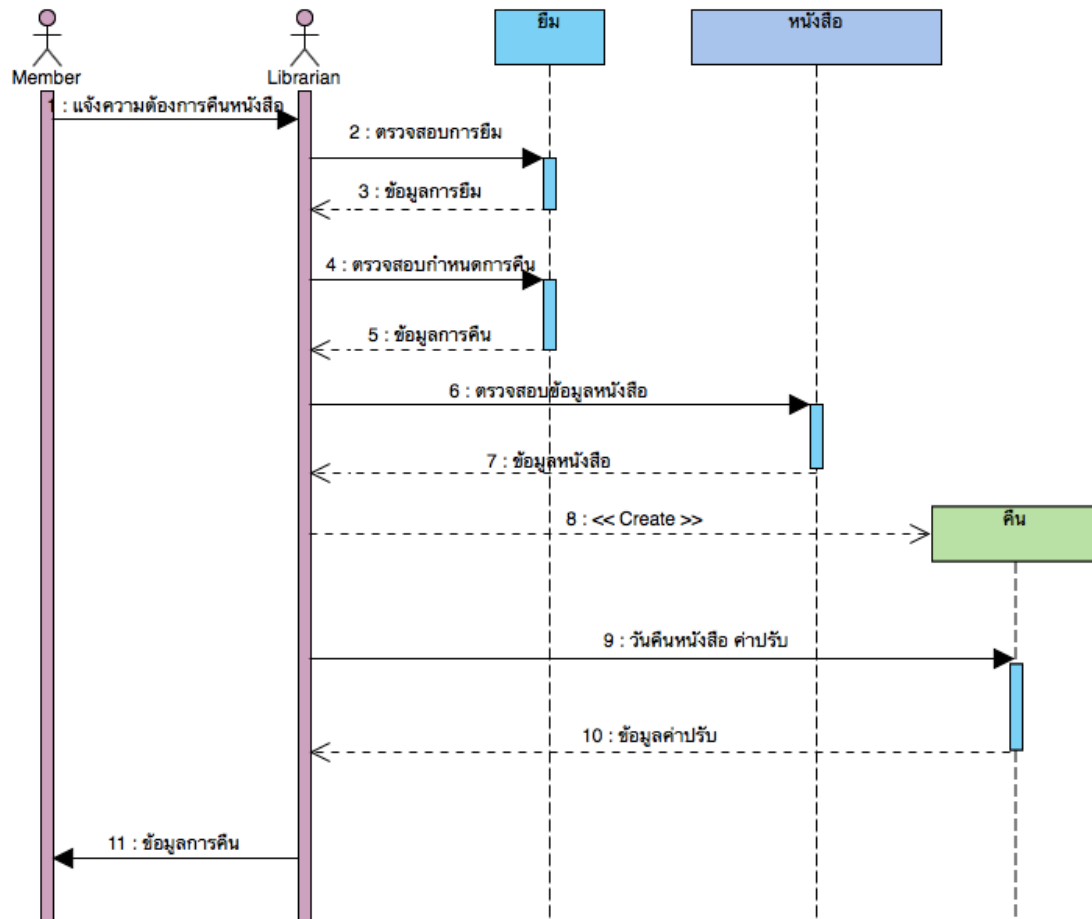
Class Diagram ระบบ Library

LIBRARY SYSTEM



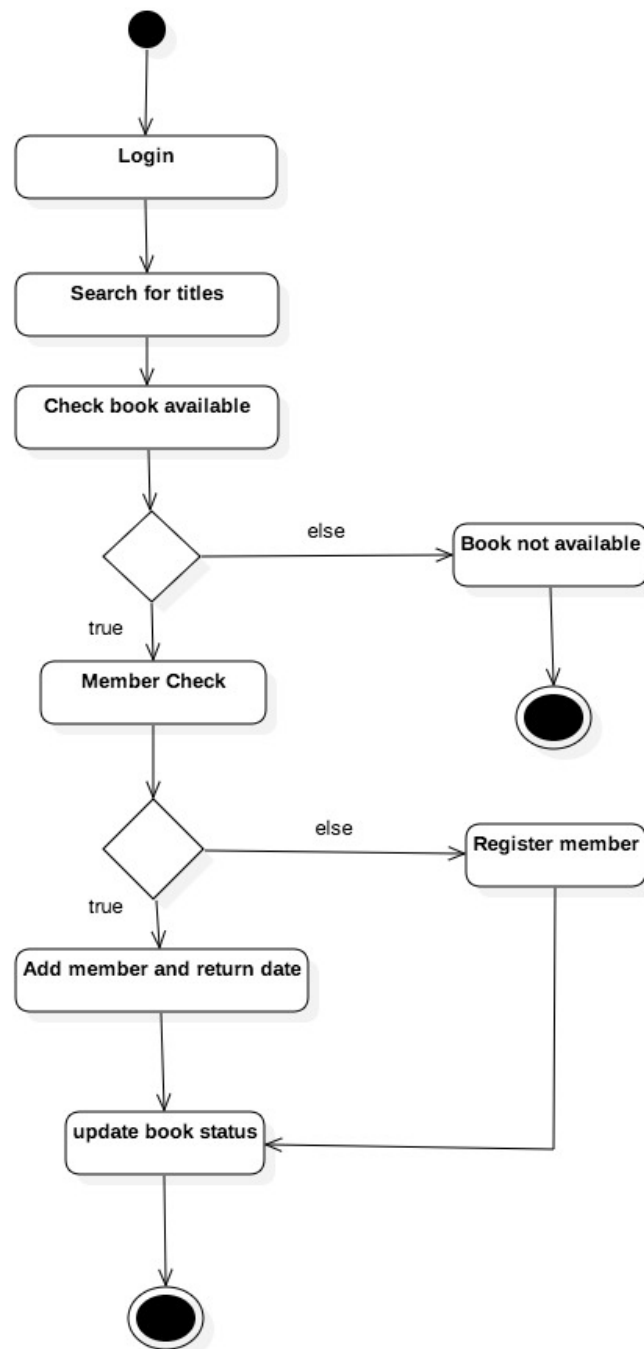
Sequence Diagram การยืมหนังสือ

LIBRARY SYSTEM



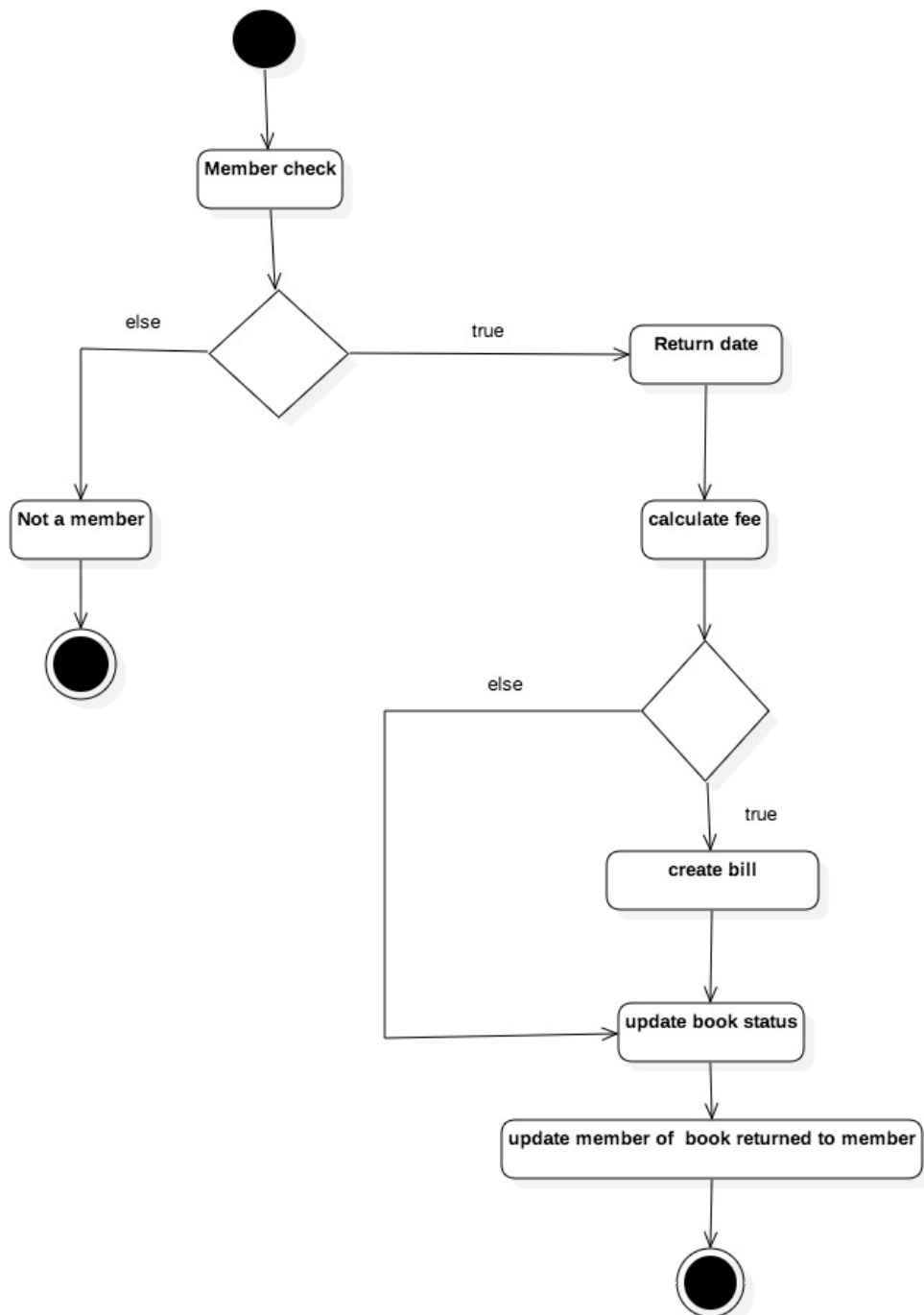
Sequence Diagram การคินหนังสือ

LIBRARY SYSTEM



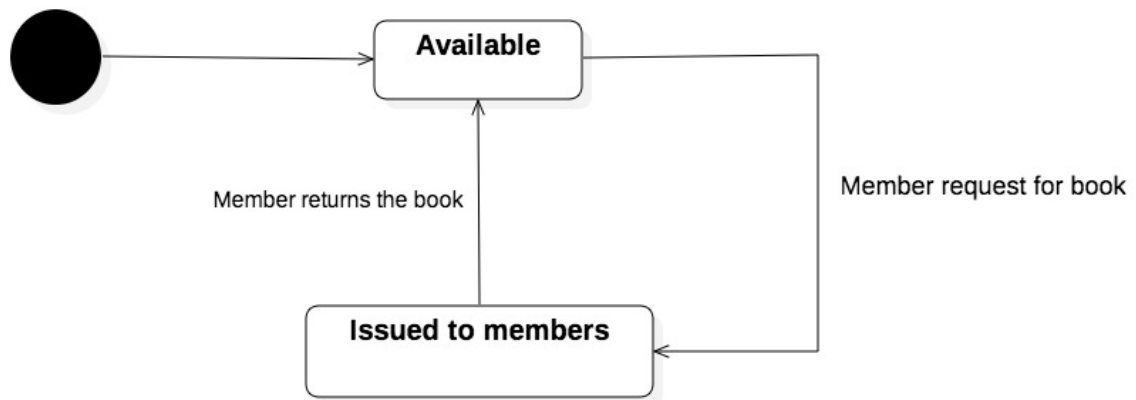
Activity การยืมหนังสือ

LIBRARY SYSTEM



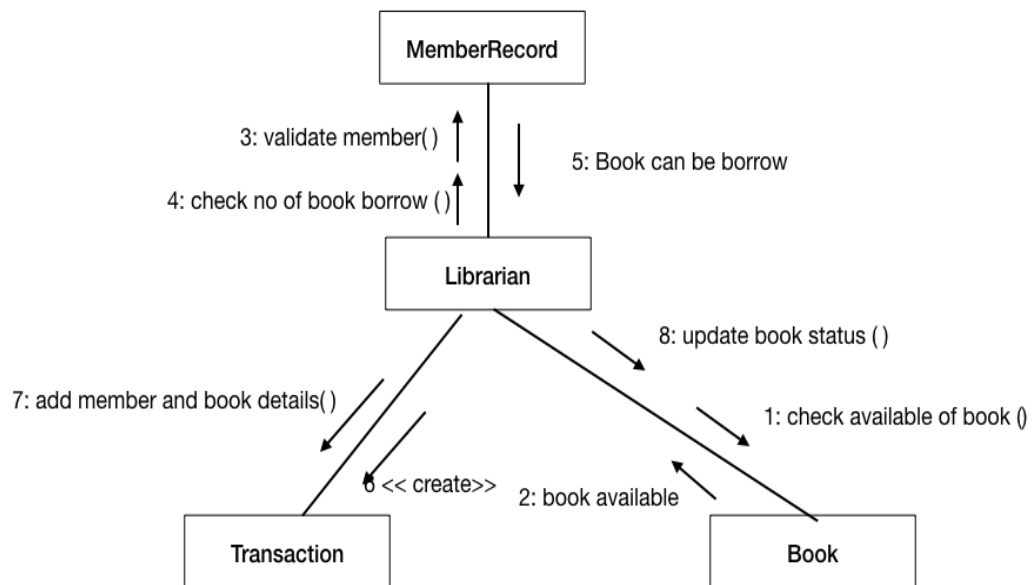
Activity การคืนหนังสือ

LIBRARY SYSTEM



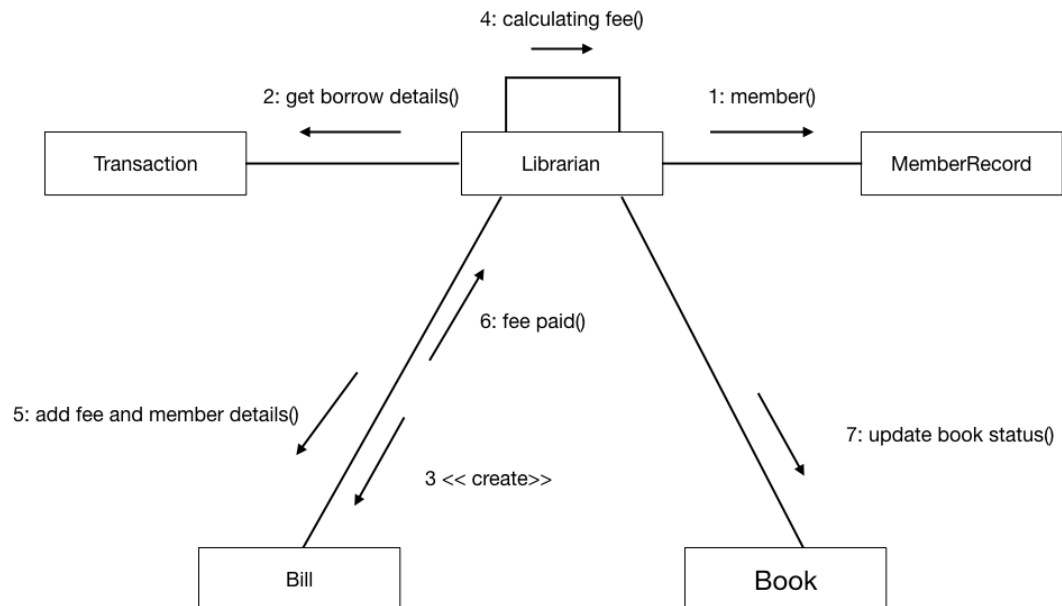
State Diagram for Book

LIBRARY SYSTEM



Collaboration Diagram for borrow Book

LIBRARY SYSTEM



Collaboration Diagram for returning Book

LIBRARY SYSTEM

Code

```
package frame;

public class AdminMenu extends JPanel {

    Image bg;

    Timer t1;

    public AdminMenu() {

        Frame.frame.setTitle("Login Admin System");

        bg = new ImageIcon("bggame.png").getImage();

        setLayout(null);

        setSize(Frame.frame.getWidth(), Frame.frame.getHeight());

        t1 = new Timer(5, new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                Frame.frame.repaint();

            }

        });

    }

};
```

LIBRARY SYSTEM

```
t1.start();

JButton adminbtn=new JButton(" Return ");

adminbtn.setBounds(425, 350, 150, 50);

adminbtn.addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent arg0) {

        Frame.frame.remove(Frame.frame.am);

        Frame.frame.rt = new Return();

        Frame.frame.add(Frame.frame.rt);

        Frame.frame.rt.requestFocusInWindow();

        Frame.frame.repaint();

    }

});

JButton Memberbtn=new JButton("History Borrow");

Memberbtn.setBounds(425, 425, 150, 50);

Memberbtn.addActionListener(new ActionListener() {
```

LIBRARY SYSTEM

```
@Override
```

```
public void actionPerformed(ActionEvent arg0) {
```

```
    Frame.frame.remove(Frame.frame.am);
```

```
    Frame.frame.hb = new HistoryBorrow();
```

```
    Frame.frame.add(Frame.frame.hb);
```

```
    Frame.frame.hb.requestFocusInWindow();
```

```
    Frame.frame.repaint();
```

```
}
```

```
});
```

```
JButton RE = new JButton("HOME");
```

```
RE.setBounds(425, 500, 150, 50);
```

```
RE.addActionListener(new ActionListener () {
```

```
    public void actionPerformed(ActionEvent arg0) {
```

```
        Frame.frame.remove(Frame.frame.am);
```

```
        Frame.frame.lb = new Library();
```

```
        Frame.frame.add(Frame.frame.lb);
```

LIBRARY SYSTEM

```
        Frame.frame.lb.requestFocusInWindow();

        Frame.frame.repaint();

    }

});

    add(adminbtn);

add(RE);

    add(Memberbtn);

}

@Override

    protected void paintComponent(Graphics g) {

        Graphics2D g2d = (Graphics2D) g;

        g.drawImage(bg, 0, 0, 1000, 625,this);

    }

}
```

LIBRARY SYSTEM

```
public class Booklist extends JPanel {  
  
    Image bg;  
  
    Timer t1;  
  
    public Booklist() {  
  
        Frame.frame.setTitle("Booklist System");  
  
        bg = new ImageIcon("2.png").getImage();  
  
        setLayout(null);  
  
        setSize(Frame.frame.getWidth(), Frame.frame.getHeight());  
  
  
        t1 = new Timer(5,new ActionListener() {  
  
            @Override  
  
            public void actionPerformed(ActionEvent e) {  
  
                Frame.frame.repaint();  

```

LIBRARY SYSTEM

```
        }

    });

    t1.start();

    JButton adminbtn=new JButton(" BORROW ");

    adminbtn.setBounds(350,500, 200, 30);

    adminbtn.addActionListener(new ActionListener() {

        @Override

        public void actionPerformed(ActionEvent arg0) {

            Frame.frame.remove(Frame.frame.mb);

            Frame.frame.br = new Borrow();

            Frame.frame.add(Frame.frame.br);

            Frame.frame.br.requestFocusInWindow();

            Frame.frame.repaint();

        }

    });

    JPanel panel = new JPanel();
```


LIBRARY SYSTEM

```
panel.setLayout(new BorderLayout(panel, BorderLayout.Y_AXIS));
```

```
String[] columnNames = { "ลำดับ", "ชื่อหนังสือ" };
```

```
Object[][] data = {  
  
    { "1", "Smith" },  
  
    { "2", "Doe" },  
  
    { "3", "Black" },  
  
    { "4", "White", },  
  
    { "5", "Brown", },  
  
    { "6", "Harry Potter" } };
```

```
JTable table = new JTable(data, columnNames);
```

```
JButton hm=new JButton(" Home ");
```

```
hm.setBounds(425, 500, 150, 50);
```

```
hm.addActionListener(new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent arg0) {
```

```
        Frame.frame.remove(Frame.frame.bl);
```

LIBRARY SYSTEM

```
        Frame.frame.mb = new Member();

        Frame.frame.add(Frame.frame.mb);

        Frame.frame.mb.requestFocusInWindow();

        Frame.frame.repaint();

    }

});

panel.add(table.getTableHeader(), BorderLayout.PAGE_START);

panel.add(table);

add(hm);

add(panel);

}

@Override

protected void paintComponent(Graphics g) {

    Graphics2D g2d = (Graphics2D) g;

    g.drawImage(bg, 0, 0, 1000, 625, this);
```

LIBRARY SYSTEM

```
    }

    public class Borrow extends JPanel {

        Image bg;

        Timer t1;

        String NB ,I;

        public Borrow() {

            Frame.frame.setTitle("Borrow System");

            bg = new ImageIcon("bggame.png").getImage();

            setLayout(null);

            setSize(Frame.frame.getWidth(), Frame.frame.getHeight());

            t1 = new Timer(5,new ActionListener() {

                @Override

                public void actionPerformed(ActionEvent e) {

                    Frame.frame.repaint();

                }

            });

        }

    }

};
```

LIBRARY SYSTEM

```
t1.start();

JLabel ID=new JLabel(" YOU ID : ");

ID.setBounds(350, 350, 100, 25);

JLabel PW=new JLabel("NAME BOOK : ");

PW.setBounds(350, 400, 100, 25);

JTextField addID=new JTextField(20);

addID.setBounds(425, 350, 150, 25);

JTextField addNB=new JTextField(20);

addNB.setBounds(425, 400, 150, 25);

JButton loginbtn=new JButton(" ยืนยัน");

loginbtn.setBounds(425, 450, 150, 50);

loginbtn.addActionListener(new ActionListener() {

@Override

    public void actionPerformed(ActionEvent arg0) {

        I = addID.getText();

        NB = addNB.getText();
```

LIBRARY SYSTEM

```
        if(I != "" ){

new Dia("การยืมสำเร็จ");

Frame.frame.remove(Frame.frame.br);

Frame.frame.mb = new Member();

Frame.frame.add(Frame.frame.mb);

Frame.frame.mb.requestFocusInWindow();

Frame.frame.repaint();

        }else{

new Dia("โปรดกรอกข้อมูล");

        }

    }

});

add(ID);

add(addID);

add(PW);

add(addNB);
```

LIBRARY SYSTEM

```
        add(loginbtn);

    }

    @Override

    protected void paintComponent(Graphics g) {

        Graphics2D g2d = (Graphics2D) g;

        g.drawImage(bg, 0, 0, 1000, 625,this);

    }

}
```

LIBRARY SYSTEM

```
public class Dia {  
  
    Dia(String s){  
  
        JDialog d=new JDialog();  
  
        JPanel p=new JPanel(null);  
  
        JLabel l=new JLabel();  
  
        d.setTitle(s);  
  
        l.setText(s+".");  
  
        if(s.length()>20) {  
  
            l.setBounds(15, 5, 200, 25);  
  
        }  
  
        else if(s.length()>10) {  
  
            l.setBounds(50, 5, 200, 25);  
  
        }  
  
        else if(s.length()>5){  
  
            l.setBounds(70, 5, 200, 25);  
  
        }  
    }  
}
```

LIBRARY SYSTEM

```
}

else {

    l.setBounds(75, 5, 200, 25);

}

JButton b=new JButton("Close");

b.setBounds(60, 30, 70,25);

b.addActionListener(new ActionListener(){

    @Override

    public void actionPerformed(ActionEvent arg0) {

        // TODO Auto-generated method stub

        d.setVisible(false);

    }

});

p.add(l);

p.add(b);

d.add(p);
```


LIBRARY SYSTEM

```
d.setSize(200, 100);  
  
d.setLocationRelativeTo(null);  
  
d.setResizable(false);  
  
d.setVisible(true);  
  
}  
  
}
```

LIBRARY SYSTEM

```
public class Frame extends JFrame{

    static Frame frame;

    Library lb;

    Login lg;

    AdminMenu am;

    Return rt;

    HistoryBorrow hb;

    Loginmember lm;

    Member mb;

    Borrow br;

    Booklist bl;

    public Frame() {

        frame=this;

        setTitle("Library System");

        setSize(1000, 625);
```

LIBRARY SYSTEM

```
        setResizable(false);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        lb = new Library();

        add(lb);

        setVisible(true);

    }

    public static void main(String[] args){

        new Frame();

    }

}
```

LIBRARY SYSTEM

```
public class HistoryBorrow extends JPanel {

    Image bg;

    Timer t1;

    public HistoryBorrow() {

        Frame.frame.setTitle("HistoryBorrow");

        bg = new ImageIcon("1.png").getImage();

        setLayout(null);

        setSize(Frame.frame.getWidth(), Frame.frame.getHeight());

        t1 = new Timer(5, new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                Frame.frame.repaint();

            }

        });

    }

}
```

LIBRARY SYSTEM

```
t1.start();

JPanel panel = new JPanel();

panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

String[] columnNames = { "ชื่อผู้ใช้", "ชื่อหนังสือ", "วันที่" };

Object[][] data = {

    { "Thanthap", "HarryPotter", "12/5/2561" },

    { "Adam", "HarryPotter3", "15/5/2561" },

    { "Jame", "HarryPotter4", "19/5/2561"},

    { "Marry", "HarryPotter7.1", "20/5/2561" },

    { "Lucas", "HarryPotter6", "25/5/2561" },

    { "Steven", "HarryPotter5", "27/5/2561" } };

JTable table = new JTable(data, columnNames);

JButton hm=new JButton(" Home ");

hm.setBounds(425, 500, 150, 50);

hm.addActionListener(new ActionListener() {

    @Override
```

LIBRARY SYSTEM

```
        public void actionPerformed(ActionEvent arg0) {

            Frame.frame.remove(Frame.frame.hb);

            Frame.frame.am = new AdminMenu();

            Frame.frame.add(Frame.frame.am);

            Frame.frame.am.requestFocusInWindow();

            Frame.frame.repaint();

        }

    });

    panel.add(table.getTableHeader(), BorderLayout.PAGE_START);

    panel.add(table);

    add(panel);

    add(hm);

}

@Override

protected void paintComponent(Graphics g) {

    Graphics2D g2d = (Graphics2D) g;
```

LIBRARY SYSTEM

```
        g.drawImage(bg, 0, 0, 1000, 625,this);

    public class Library extends JPanel {

        Image bg;

        Timer t1;

        public Library() {

            bg = new ImageIcon("bggame.png").getImage();

            setLayout(null);

            setSize(Frame.frame.getWidth(), Frame.frame.getHeight());

            t1 = new Timer(5,new ActionListener() {

                @Override

                public void actionPerformed(ActionEvent e) {

                    Frame.frame.repaint();

                }

            });

            t1.start();
```

LIBRARY SYSTEM

```

JButton adminbtn=new JButton(" ADMIN ");

adminbtn.setBounds(425, 350, 150, 50);

adminbtn.addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent arg0) {

        Frame.frame.remove(Frame.frame.lb);

        Frame.frame.lg = new Login();

        Frame.frame.add(Frame.frame.lg);

        Frame.frame.lg.requestFocusInWindow();

        Frame.frame.repaint();

    }

});

JButton Memberbtn=new JButton("Member");

Memberbtn.setBounds(425,425, 150, 50);

Memberbtn.addActionListener(new ActionListener() {

    @Override
```


LIBRARY SYSTEM

```
        public void actionPerformed(ActionEvent arg0) {

            Frame.frame.remove(Frame.frame.lb);

            Frame.frame.lm = new Loginmember();

            Frame.frame.add(Frame.frame.lm);

            Frame.frame.lm.requestFocusInWindow();

            Frame.frame.repaint();

        }

    });

    add(adminbtn);

    add(Memberbtn);

}

@Override

protected void paintComponent(Graphics g) {

    Graphics2D g2d = (Graphics2D) g;

    g.drawImage(bg, 0, 0, 1000, 625, this);

}
```

LIBRARY SYSTEM

```
}
```

```
public class Login extends JPanel {
```

```
    Image bg;
```

```
    Timer t1;
```

```
    public Login() {
```

```
        Frame.frame.setTitle("Login Admin System");
```

```
        bg = new ImageIcon("bggame.png").getImage();
```

```
        setLayout(null);
```

```
        setSize(Frame.frame.getWidth(), Frame.frame.getHeight());
```

```
        System.out.println("admin");
```

```
        t1 = new Timer(5, new ActionListener() {
```

```
            @Override
```

```
            public void actionPerformed(ActionEvent e) {
```

```
                Frame.frame.repaint();
```

```
            }
```

```
        });
```

LIBRARY SYSTEM

```
t1.start();
```

```
JLabel ID=new JLabel(" ID : ");
```

```
ID.setBounds(350, 350, 100, 25);
```

```
JLabel PW=new JLabel("Password : ");
```

```
PW.setBounds(350, 400, 100, 25);
```

```
JTextField addID=new JTextField(20);
```

```
addID.setBounds(425, 350, 150, 25);
```

```
JPasswordField addPW=new JPasswordField(20);
```

```
addPW.setBounds(425, 400, 150, 25);
```

```
JButton loginbtn=new JButton(" LOGIN ");
```

```
loginbtn.setBounds(425, 450, 150, 50);
```

```
loginbtn.addActionListener(new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent arg0) {
```

```
        File inputFile = new File("Admin.txt");
```

```
        BufferedReader reader=null;
```

LIBRARY SYSTEM

```
try {  
  
    reader = new BufferedReader(new FileReader(inputFile));  
  
    String password= reader.readLine();  
  
    reader.close();  
  
    if(addPW.getText().equals(password)) {  
  
        Frame.frame.remove(Frame.frame.lg);  
  
        Frame.frame.am = new AdminMenu();  
  
        Frame.frame.add(Frame.frame.am);  
  
        Frame.frame.am.requestFocusInWindow();  
  
        Frame.frame.repaint();  
  
    }else {  
  
        new Dia("โปรดกรอกข้อมูลใหม่อีกครั้ง");  
  
    }  
  
} catch (FileNotFoundException e) {
```

LIBRARY SYSTEM

```
        return ;

    } catch (IOException e) {

        return ;

    }

}

});

add(ID);

add(addID);

add(PW);

add(addPW);

add(loginbtn);

}

@Override

protected void paintComponent(Graphics g) {

    Graphics2D g2d = (Graphics2D) g;

    g.drawImage(bg, 0, 0, 1000, 625,this); }}
```

LIBRARY SYSTEM

```
public class Loginmember extends JPanel {

    Image bg;

    Timer t1;

    public Loginmember() {

        Frame.frame.setTitle("Login member ");

        bg = new ImageIcon("bggame.png").getImage();

        setLayout(null);

        setSize(Frame.frame.getWidth(), Frame.frame.getHeight());

        t1 = new Timer(5,new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                Frame.frame.repaint();

            }

        });

        t1.start();
    }
}
```

LIBRARY SYSTEM

```
JLabel ID=new JLabel("  Name  : ");

ID.setBounds(350, 350, 100, 25);

JLabel SID=new JLabel("student's ID : ");

SID.setBounds(350, 400, 100, 25);

JTextField addID=new JTextField(20);

addID.setBounds(425, 350, 150, 25);

JTextField addSID=new JTextField(20);

addSID.setBounds(425, 400, 150, 25);

JButton loginbtn=new JButton(" LOGIN ");

loginbtn.setBounds(425, 450, 150, 50);

loginbtn.addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent arg0) {

        Frame.frame.remove(Frame.frame.lm);

        Frame.frame.mb = new Member();

        Frame.frame.add(Frame.frame.mb);
```

LIBRARY SYSTEM

```
        Frame.frame.mb.requestFocusInWindow();

        Frame.frame.repaint();

    }

});

add(ID);

add(addID);

add(SID);

add(addSID);

add(loginbtn);

}

@Override

protected void paintComponent(Graphics g) {

    Graphics2D g2d = (Graphics2D) g;

    g.drawImage(bg, 0, 0, 1000, 625,this);

}

}
```


LIBRARY SYSTEM

```
public class Member extends JPanel {  
  
    Image bg;  
  
    Timer t1;  
  
    public Member() {  
  
        Frame.frame.setTitle("Member System");  
  
        bg = new ImageIcon("bggame.png").getImage();  
  
        setLayout(null);  
  
        setSize(Frame.frame.getWidth(), Frame.frame.getHeight());  
  
        t1 = new Timer(5,new ActionListener() {  
  
            @Override  
  
            public void actionPerformed(ActionEvent e) {  
  
                Frame.frame.repaint();  
  
            }  
  
        });  
  
        t1.start();  
    }  
}
```

LIBRARY SYSTEM

```

JButton adminbtn=new JButton(" BORROW ");

adminbtn.setBounds(425, 350, 150, 50);

adminbtn.addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent arg0) {

        Frame.frame.remove(Frame.frame.mb);

        Frame.frame.br = new Borrow();

        Frame.frame.add(Frame.frame.br);

        Frame.frame.br.requestFocusInWindow();

        Frame.frame.repaint();

    }

});

JButton Memberbtn=new JButton("VIEW BOOK");

Memberbtn.setBounds(425, 425, 150, 50);

Memberbtn.addActionListener(new ActionListener() {

    @Override
```

LIBRARY SYSTEM

```
        public void actionPerformed(ActionEvent arg0) {

            Frame.frame.remove(Frame.frame.mb);

            Frame.frame.bl = new Booklist();

            Frame.frame.add(Frame.frame.bl);

            Frame.frame.bl.requestFocusInWindow();

            Frame.frame.repaint();

        }

    });

    JButton RE = new JButton("HOME");

    RE.setBounds(425, 500, 150, 50);

    RE.addActionListener(new ActionListener () {

        public void actionPerformed(ActionEvent arg0) {

            Frame.frame.remove(Frame.frame.mb);

            Frame.frame.lb = new Library();

            Frame.frame.add(Frame.frame.lb);

            Frame.frame.lb.requestFocusInWindow();
```

LIBRARY SYSTEM

```
        Frame.frame.repaint();

    }

});

    add(adminbtn);

add(RE);

    add(Memberbtn);

}

@Override

    protected void paintComponent(Graphics g) {

        Graphics2D g2d = (Graphics2D) g;

        g.drawImage(bg, 0, 0, 1000, 625,this);

    }

}
```

LIBRARY SYSTEM

```
public class Return extends JPanel {  
  
    Image bg;  
  
    Timer t1;  
  
    public Return() {  
  
        Frame.frame.setTitle("Return System");  
  
        bg = new ImageIcon("bggame.png").getImage();  
  
        setLayout(null);  
  
        setSize(Frame.frame.getWidth(), Frame.frame.getHeight());  
  
        t1 = new Timer(5,new ActionListener() {  
  
            @Override  
  
            public void actionPerformed(ActionEvent e) {  
  
                Frame.frame.repaint();  
  
            }  
  
        });  
  
        t1.start();  
    }  
}
```

LIBRARY SYSTEM

```
JLabel ID=new JLabel(" YOUR ID : ");
```

```
ID.setBounds(350, 350, 100, 25);
```

```
JLabel NB=new JLabel("NAME BOOK : ");
```

```
NB.setBounds(350, 400, 100, 25);
```

```
JLabel date=new JLabel(" DATE ");
```

```
date.setBounds(350, 450, 100, 25);
```

```
JTextField addID=new JTextField(20);
```

```
addID.setBounds(450, 350, 150, 25);
```

```
JTextField addNB=new JTextField(20);
```

```
addNB.setBounds(450, 400, 150, 25);
```

```
JTextField adddate=new JTextField(20);
```

```
adddate.setBounds(450, 450, 150, 25);
```

```
JButton RTbtn=new JButton(" ยืนยัน");
```

```
RTbtn.setBounds(450, 500, 100, 30);
```

```
RTbtn.addActionListener(new ActionListener() {
```

```
    @Override
```

LIBRARY SYSTEM

```
        public void actionPerformed(ActionEvent arg0) {  
  
            new Dia("การคืนสำเร็จ");  
  
            Frame.frame.remove(Frame.frame.rt);  
  
            Frame.frame.am = new AdminMenu();  
  
            Frame.frame.add(Frame.frame.am);  
  
            Frame.frame.am.requestFocusInWindow();  
  
            Frame.frame.repaint();  
  
        }  
  
    });  
  
    add(ID);  
  
        add(addID);  
  
        add(NB);  
  
    add(date);  
  
    add(adddate);  
  
        add(addNB);
```

LIBRARY SYSTEM

```
        add(RTbtn);

    }

    @Override

    protected void paintComponent(Graphics g) {

        Graphics2D g2d = (Graphics2D) g;

        g.drawImage(bg, 0, 0, 1000, 625,this);

    }

}
```


LIBRARY SYSTEM

รายชื่อผู้จัดทำ

นายกิตติศักดิ์	ศรีเดช	5904062630012
นายฐานทัพ	โชติพานิช	5904062630080
นายชนพล	โสমনะพันธุ์	5904062630209
นายชนานนท์	คำวัน	590406263024