

รายงานโครงการ

ชื่อระบบ: Help Desk

Platform: Website Application

Feature หลัก

1. แจ้งซ่อม
2. แสดงสถานะการซ่อมบำรุง
3. ประกาศ

รายชื่อสมาชิกและเลขประจำตัว

1. 59130500058 นายพีรวัศ อิบรรยืน
2. 59130500069 นายภูริภัทร อารยศิริกุล
3. 59130500086 นางสาววิรดา วีรภัทรสกุลชัย

Type of stakeholders	How are they related with your system
Internal Operational	
ผู้ให้บริการหอพัก	เป็นบุคคลที่เกี่ยวข้องกับระบบโดยตรง โดยเป็นผู้ใช้งานระบบนี้
ผู้พักอาศัยหอพัก	เป็นบุคคลที่ให้ข้อมูลการแจ้งซ่อมบำรุง และติดตามการซ่อมบำรุง

Feature Name	รายละเอียด	ผู้รับผิดชอบ	ความสำเร็จ (%ความก้าวหน้า)
Feature 1 : แจ้งซ่อม			
1. แจ้งปัญหาการซ่อมบำรุง	<p>ผู้พักอาศัยหอพักแจ้งปัญหาแก่ผู้ให้บริการหอพัก เพื่อให้ติดต่อช่างซ่อมบำรุงให้ดำเนินการต่อไปโดยจะมีหัวข้อในการแจ้งซ่อมมาให้เลือก</p> <ul style="list-style-type: none"> - constructor ของ class ReportProblem - method แจ้งปัญหา 	วิรดา, พีรวัศ, ภูริภัทร	100 %

Document-SEC02-PROJ1

	- JSP และ Servlet		
2. แสดงรายการการซ่อมบำรุง	1. แจ้งรายการการซ่อมที่ได้รับแจ้ง ให้ผู้แจ้งปัญหา 2. แสดงสถานการณ์ซ่อมทั้งหมดแก่ ผู้ให้บริการ	ภูริภัทร	100 %
Feature 2 : แสดงสถานะรายการซ่อมบำรุง			
1. แสดงสถานะการซ่อมบำรุง	1. แจ้งสถานะการซ่อม ให้ผู้แจ้งปัญหา 2. แสดงสถานะซ่อมทั้งหมดแก่ ผู้ให้บริการ	ภูริภัทร	100 %
2. เปลี่ยนสถานะการซ่อมบำรุง	ผู้ให้บริการหอพักเปลี่ยนสถานะของการซ่อมบำรุง - method สำหรับเปลี่ยนค่าใน database - method สำหรับเปลี่ยนค่าใน object - JSP และ Servlet	วิรดา, พิรวัส, ภูริภัทร	100 %
3. สรุปการซ่อมบำรุง	แสดงจำนวนการซ่อมในแต่ละประเภทเป็นตัวเลขให้แก่ ผู้ดูแลหอพักซึ่งจะแสดงข้อมูล 4 ส่วนคือ <ol style="list-style-type: none">จำนวนซ่อมทั้งหมดของหอในปีนั้น ๆห้องที่มีการซ่อมมากที่สุดในแต่ละปัญหาในปีนั้น ๆจำนวนซ่อมทั้งหมดของห้องปีก่อนหน้าห้องที่มีการซ่อมมากที่สุดในแต่ละปัญหาในปีก่อนหน้า	ภูริภัทร	100 %
Feature 3 : ประกาศ			
1. แจ้งประกาศ	ผู้ให้บริการแจ้งให้ผู้เข้าพักทราบบางสิ่ง เช่น จะมีการซ่อมบำรุงครั้งใหญ่ๆ - class ListAnnounce - class Announce - JSP และ Servlet	วิรดา, พิรวัส, ภูริภัทร	100 %
2. แสดงคู่มือการใช้งานระบบ/การใช้งานอุปกรณ์	เพื่อบอกวิธีการใช้งานระบบ หรืออุปกรณ์ต่าง ๆ	ภูริภัทร	100 %
อื่น ๆ			
1. หน้า login	หน้าเข้าสู่ระบบเพื่อเข้าสู่การทำรายการเพื่อคัดกรอง	ภูริภัทร	100 %

	User และแยกประเภทการแสดงผลให้ตรงกับประเภทผู้ใช้และหอที่อยู่		
--	---	--	--

Function Name	ระยะเวลา															
	Sprint 1				Sprint 2				Sprint 3							
	ก.พ.				มี.ค.				เม.ย.				พ.ค.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
แจ้งปัญหาการซ่อมบำรุง																
แสดงรายการการซ่อมบำรุง																
เปลี่ยนสถานะการณ์ซ่อมบำรุง																
แสดงสถานะการซ่อมบำรุง																
แจ้งประกาศ																
แสดงคุณมีการใช้งานระบบ/ การใช้งานอุปกรณ์																
สรุปการซ่อมบำรุง																
หน้า login																

Feature ที่ 1 : การแจ้งซ่อม ผู้รับผิดชอบ พิริวัศ, ภูริภัทร, วิรดา

ผลการดำเนินงาน โดย ภูริภัทร

- สร้าง class ConnectionBuilder สำหรับติดต่อกับ JDBC
- สร้าง class ListProblem สำหรับเก็บปัญหาที่เคยมีการแจ้งไว้ใน database โดยแบ่งการทำงานเป็น 2 แบบสำหรับ admin และ student
- สร้าง class Problem สำหรับเก็บแต่ละปัญหาจาก database
- สร้าง servlet และ jsp สำหรับแสดงรายการปัญหา
- สร้าง servlet และ jsp สำหรับแจ้งปัญหา

Document-SEC02-PROJ1

Document-SEC02-PROJ1



The screenshot shows a Java IDE interface with the following details:

- Title Bar:** Problem.java
- Menu Bar:** Source, History, File, Edit, View, Tools, Window, Help
- Toolbar:** Contains icons for New, Open, Save, Cut, Copy, Paste, Find, Select All, and others.
- Code Editor:** Displays the `Problem.java` file content. The code is a class definition with a constructor and exception handling blocks.

```
5 public class Problem {
6     private int causeId;
7     private int problemP;
8     private int problemC;
9     private int problemN;
10    private String name;
11    private int problemId;
12    private int statusId;
13    private String status;
14    private int roomNo;
15
16    public Problem(int causeId, int roomNo) {
17        this.causedId = causeId;
18        this.roomNo = roomNo;
19        try {
20            Connection connect = ConnectionBuilder.getConnection();
21            PreparedStatement ps = connect.prepareStatement(
22                    "SELECT RecordProblem.problemPast, RecordProblem.problemCurrent, RecordProblem.problemNow, Problem.problemId, Problem.problemName
23                    + "FROM RecordProblem "
24                    + "INNER JOIN Cause ON RecordProblem.Cause_causeId = Cause.causedId "
25                    + "INNER JOIN Problem ON Cause.Problem_problemId = Problem.problemId "
26                    + "INNER JOIN status ON Cause.status_statusId = status.statusId "
27                    + "WHERE RecordProblem.Cause_causeId = ? ");
28            ps.setInt(1,causeId);
29            ResultSet result = ps.executeQuery();
30            while(result.next()) {
31                problemP = result.getInt("problemPast");
32                problemC = result.getInt("problemCurrent");
33                problemN = result.getInt("problemNow");
34                problemId = result.getInt("problemId");
35                name = result.getString("problemName");
36                status = result.getString("statusName");
37                statusId = result.getInt("statusId");
38            }
39        } catch(SQLEXception | ClassNotFoundException e) {
40            e.printStackTrace();
41        }
42        catch(Exception e) {
43
44        }
45    }
46}
```



The screenshot shows the Eclipse IDE interface with the 'ListProblemServlet.java' file open in the editor. The code implements a HttpServlet to handle requests for a list of problems. It imports various Java and JSP packages, defines a protected void processRequest method, and uses session and attribute objects to manage user information and the list of problems.

```
1 package Servlet;
2
3 import Model.ListProblem;
4 import Model.User;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import javax.servlet.ServletException;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import javax.servlet.http.HttpSession;
12
13 public class ListProblemServlet extends HttpServlet {
14     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
15         throws ServletException, IOException {
16         response.setContentType("text/html;charset=UTF-8");
17
18         HttpSession s = request.getSession(true);
19         User user = (User)s.getAttribute("user");
20         ListProblem list = Model.ListProblem.getShowListProblem(user.getUserId());
21         String target = "/ShowListProblem.jsp";
22
23         request.setAttribute("message", list);
24
25         getServletContext().getRequestDispatcher(target).forward(request, response);
26     }
27
28     HttpServlet methods. Click on the + sign on the left to edit the code.
29
30 }
```



List Problem

#	problem name
1	กรปะด្ឋានទុក
2	WiFi បានទុក
3	Remote នៃការប្រើប្រាស់ទុក

กลับสู่เมนูหลัก

Document-SEC02-PROJ1

The screenshot shows two windows. The top window is a Java IDE editor titled "ReportProblemServlet.java" containing Java code for a servlet. The bottom window is a web browser titled "Report Problem" showing a list of problems with "Report" buttons.

ReportProblemServlet.java

```
1 package Servlet;
2
3 import Model.ReportProblem;
4 import java.io.IOException;
5 import java.io.PrintWriter;
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10
11 public class ReportProblemServlet extends HttpServlet {
12     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
13         throws ServletException, IOException {
14         response.setContentType("text/html;charset=UTF-8");
15
16         ReportProblem reportP = Model.ReportProblem.getReportProblem(2);
17         String target = "/ReportProblem.jsp";
18         request.setAttribute("message", reportP);
19         getServletContext().getRequestDispatcher(target).forward(request, response);
20     }
21
22     HttpServlet methods. Click on the + sign on the left to edit the code.
23 }
24
25
```

Report Problem

A sign in as STUDENT

#	problem name	report
1	กรปะดูชารุด	<button>Report</button>
2	หน้าต่างแตก	<button>Report</button>
3	ผุ้งวอดขาด	<button>Report</button>
4	ฝ้า/เพดานชำรุด	<button>Report</button>
5	WiFi ชำรุด	<button>Report</button>
6	เต้าเสียบชำรุด	<button>Report</button>
7	สวิทช์ไฟชำรุด	<button>Report</button>
8	หลอดไฟชำรุด	<button>Report</button>
9	แอร์ชำรุด	<button>Report</button>
10	Remote แอร์ชำรุด	<button>Report</button>
11	TV ชำรุด	<button>Report</button>
12	Remote TV ชำรุด	<button>Report</button>

ผลการดำเนินงาน โดย พิรవัล

- สร้าง class ProblemName สำหรับเก็บรายชื่อปัญหาจาก database
- เขียน constructor ของ class ReportProblem

Document-SEC02-PROJ1

The screenshot shows two Java code editors side-by-side in a development environment.

ProblemName.java

```
1 package Model;
2
3 import java.sql.*;
4 import java.util.ArrayList;
5
6 public class ProblemName {
7     private ArrayList<String> problemName;
8
9     public ProblemName() {
10         problemName = new ArrayList<String>();
11         try {
12             Connection connect = ConnectionBuilder.getConnection();
13             PreparedStatement ps = connect.prepareStatement("SELECT problemName FROM Problem");
14             ResultSet result = ps.executeQuery();
15             while(result.next()) {
16                 problemName.add(result.getString("problemName"));
17             }
18             ps.close();
19             connect.close();
20         } catch(SQLException | ClassNotFoundException e) {
21             e.printStackTrace();
22         } catch(Exception e) {
23             e.printStackTrace();
24         }
25     }
26
27     public ArrayList<String> getProblemName() {...3 lines...}
28     public void setProblemName(ArrayList<String> problemName) {...3 lines...}
29
30 }
31
32 }
```

ReportProblem.java

```
1 package Model;
2
3 import java.sql.*;
4 import java.util.ArrayList;
5
6 public class ReportProblem extends ProblemName {
7     private long userId;
8     private int roomID;
9     private ArrayList<Problem> list;
10
11     public ReportProblem(long userId) {
12         super();
13         this.userId = userId;
14         try {
15             Connection connect = ConnectionBuilder.getConnection();
16             PreparedStatement ps = connect.prepareStatement(
17                     "SELECT User.userid, Room.roomid "
18                     + "FROM User "
19                     + "INNER JOIN Renter ON User.userid = Renter.User_userid "
20                     + "INNER JOIN Renter_has_Room ON Renter.renterId = Renter_has_Room.Renter_renterId "
21                     + "INNER JOIN Room ON Renter_has_Room.Room_roomid = Room.roomid "
22                     + "WHERE userid = ?");
23             ps.setLong(1, userId);
24             ResultSet result = ps.executeQuery();
25             while(result.next()) {
26                 roomID = result.getInt("roomID");
27             }
28             ps.close();
29             connect.close();
30         } catch(SQLException | ClassNotFoundException e) {
31             e.printStackTrace();
32         } catch(Exception e) {
33             System.out.println(e.getMessage());
34         }
35
36         list = new ListProblem(userId).getList();
37     }
38
39     public void report(int problemId) {
40         boolean have = false;
41         for (int i = 0; i < list.size(); i++) {
42             if (list.get(i).getProblemId() == problemId) {
43                 have = true;
44             }
45         }
46     }
47 }
```

ผลการดำเนินงาน โดย วิรดา

- สร้าง class ReportProblem สำหรับแจ้งปัญหาเพื่อไปเก็บใน database
 - เขียน method เพื่อแจ้งปัญหา

Document-SEC02-PROJ1

Feature ที่ 2 : แสดงสถานะรายการซ้อมบำรุง ผู้รับผิดชอบ พิรவัล, ภูริภัทร, วิรดา

ผลการดำเนินงาน โดย ภูริภัทร

- สร้าง servlet และ jsp สำหรับแสดงสถานะปัจจุบัน
 - สร้าง class Summary สำหรับสรุปการซ้อมบำรุง
 - สร้าง servlet และ jsp สำหรับสรุปการซ้อมบำรุง
 - สร้าง servlet และ jsp สำหรับการเปลี่ยนสถานะ จำนวน 2 หน้า

Document-SEC02-PROJ1



```
1 package Servlet;
2
3 import Model.ListProblem;
4 import Model.User;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import javax.servlet.ServletException;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import javax.servlet.http.HttpSession;
12
13 public class ListStatusServlet extends HttpServlet {
14     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
15             throws ServletException, IOException {
16         response.setContentType("text/html;charset=UTF-8");
17
18         HttpSession s = request.getSession(true);
19         User user = (User)s.getAttribute("user");
20         ListProblem lists = Model.ListProblem.getShowListProblem(user.getUserId());
21         String target = "/showListStatus.jsp";
22
23         request.setAttribute("message", lists);
24
25         getServletContext().getRequestDispatcher(target).forward(request, response);
26     }
27
28     HttpServlet methods. Click on the + sign on the left to edit the code.
29
30 }
```

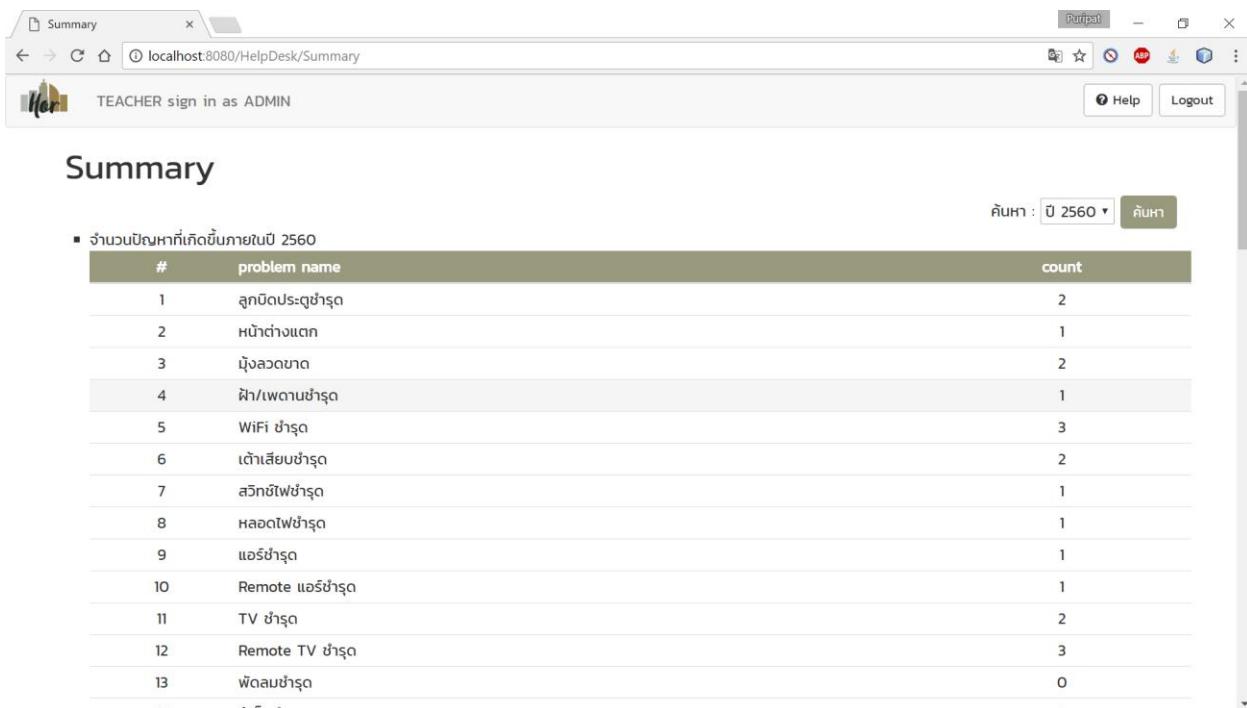


```
1 package Model;
2
3 import java.util.ArrayList;
4
5 public class Summary extends ProblemName{
6     private long userId;
7     private int[] past;
8     private int[] maxPast;
9     private int[] roomMaxPast;
10    private int[] now;
11    private int[] maxNow;
12    private int[] roomMaxNow;
13    private ArrayList<Problem> list;
14
15    public Summary(Long userId) {
16        this.userId = userId;
17        past = new int[27];
18        for (int i = 0; i < past.length; i++) {
19            past[i] = 0;
20        }
21        maxPast = past;
22        roomMaxPast = past;
23        now = past;
24        maxNow = past;
25        roomMaxNow = past;
26        list = new ListProblem(userId).getList();
27        for (int i = 0; i < list.size(); i++) {
28            past[(list.get(i).getProblemId())-1] += list.get(i).getProblemP();
29            if(past[(list.get(i).getProblemId())-1] < list.get(i).getProblemP()) {
30                maxPast[(list.get(i).getProblemId())-1] = list.get(i).getProblemP();
31                roomMaxPast[(list.get(i).getProblemId())-1] = list.get(i).getRoomP();
32            }
33            now[(list.get(i).getProblemId())-1] += list.get(i).getProblemN();
34            if(now[(list.get(i).getProblemId())-1] < list.get(i).getProblemN()) {
35                maxNow[(list.get(i).getProblemId())-1] = list.get(i).getProblemN();
36                roomMaxNow[(list.get(i).getProblemId())-1] = list.get(i).getRoomN();
37            }
38        }
39    }
40}
```

Document-SEC02-PROJ1



```
1  import javax.servlet.http.HttpServlet;
2  import javax.servlet.http.HttpSession;
3
4  public class SummaryServlet extends HttpServlet {
5      @Override
6      protected void doGet(HttpServletRequest request, HttpServletResponse response)
7          throws ServletException, IOException {
8          response.setContentType("text/html;charset=UTF-8");
9
10         HttpSession s = request.getSession(true);
11         User user = (User)s.getAttribute("user");
12         Summary summary = Model.Summary.getSummary(user.getUserId());
13         String search = "current";
14         String target = "/Summary.jsp";
15
16         request.setAttribute("summary", summary);
17         request.setAttribute("search", search);
18
19         getServletContext().getRequestDispatcher(target).forward(request, response);
20     }
21
22     @Override
23     protected void doPost(HttpServletRequest request, HttpServletResponse response)
24         throws ServletException, IOException {
25         response.setContentType("text/html;charset=UTF-8");
26
27         String search = request.getParameter("search");
28         HttpSession s = request.getSession(true);
29         User user = (User)s.getAttribute("user");
30         Summary summary = Model.Summary.getSummary(user.getUserId());
31         String target = "/Summary.jsp";
32
33         request.setAttribute("summary", summary);
34         request.setAttribute("search", search);
35
36         getServletContext().getRequestDispatcher(target).forward(request, response);
37     }
38 }
```



Document-SEC02-PROJ1

Summary

localhost:8080/HelpDesk/Summary

ผู้อ่านที่เกิดปัญหามากที่สุดและจำนวนครั้งของปี 2560

#	problem name	room	count
1	ลูกบิดประตูชำรุด	1	2
2	หน้าต่างแตก	1	1
3	บุ้งหลวบหาด	2	2
4	ฝ้า/เพดานชำรุด	1	1
5	WiFi ชำรุด	1	3
6	เต้าเสียงชำรุด	1	1
		2	
7	สวิงชี้ไฟชำรุด	2	1
8	หลอดไฟชำรุด	1	1
9	แอร์ชำรุด	1	1
10	Remote แอร์ชำรุด	1	1
11	TV ชำรุด	1	2
12	Remote TV ชำรุด	1	3
13	พัดลมชำรุด	ไม่มี	-

UpdateStatus1Servlet.java

Source History

```
1 package Servlet;
2
3 import Model.ListProblem;
4 import Model.User;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import javax.servlet.ServletException;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import javax.servlet.http.HttpSession;
12
13 public class UpdateStatus1Servlet extends HttpServlet {
14
15     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
16         throws ServletException, IOException {
17         response.setContentType("text/html;charset=UTF-8");
18
19         HttpSession s = request.getSession(true);
20         User user = (User)s.getAttribute("user");
21         ListProblem lists = Model.ListProblem.getShowListProblem(user.getUserId());
22         String target = "/UpdateStatus1.jsp";
23
24         request.setAttribute("message", lists);
25
26         getServletContext().getRequestDispatcher(target).forward(request, response);
27     }
28
29     HttpServlet methods. Click on the + sign on the left to edit the code.
30 }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68 }
```

Document-SEC02-PROJ1

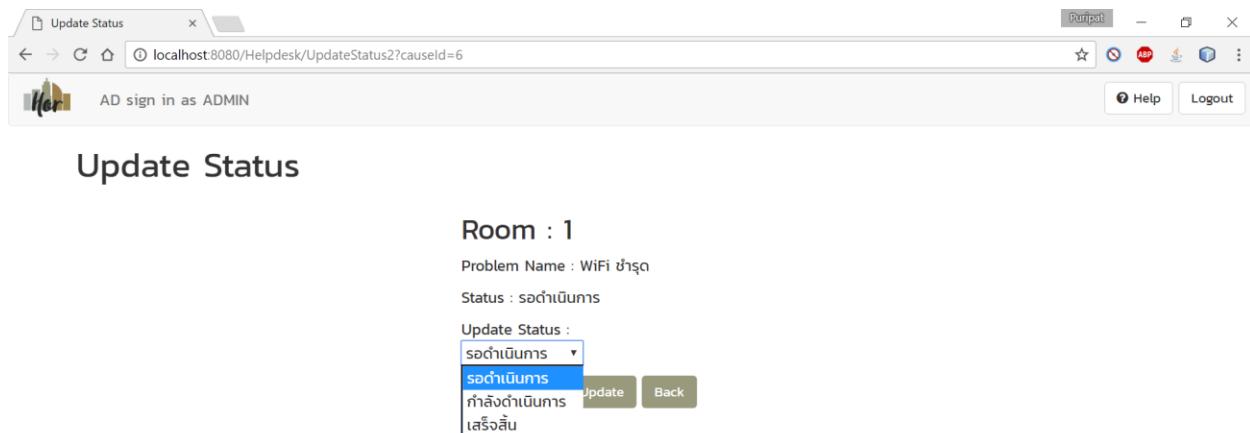
The screenshot shows a web application interface titled "Update Status". At the top, there's a header with a logo, the text "AD sign in as ADMIN", and navigation links for "Help" and "Logout". Below the header is a table titled "Update Status" containing a list of 11 problems. Each row in the table includes a problem ID, name, room number, current status (represented by a colored button), and an "Update" button.

#	problem name	room	status	
1	กรประตูชำรุด	1	เสร็จสิบ	<button>Update</button>
2	หน้าต่างแตก	1	รอดำเนินการ	<button>Update</button>
3	บังคับดูด	1	กำลังดำเนินการ	<button>Update</button>
4	ผ้า/เพดานชำรุด	1	รอดำเนินการ	<button>Update</button>
5	WiFi ชำรุด	1	รอดำเนินการ	<button>Update</button>
6	Remote แอร์ชำรุด	1	กำลังดำเนินการ	<button>Update</button>
7	อ่างล้างหน้าชำรุด	1	เสร็จสิบ	<button>Update</button>
8	ท่อปั่นร้อน	1	รอดำเนินการ	<button>Update</button>
9	กรประตูชำรุด	2	รอดำเนินการ	<button>Update</button>
10	แอร์ชำรุด	2	กำลังดำเนินการ	<button>Update</button>
11	Remote TV ชำรุด	2	รอดำเนินการ	<button>Update</button>

The screenshot shows a Java code editor with the file "UpdateStatus2Servlet.java" open. The code contains three methods: doGet, doPost, and getServletInfo. The doGet method retrieves a cause ID from the request, finds the corresponding problem, and forwards it to a target JSP page. The doPost method updates the status of a problem using its status ID and forwards it to the same target JSP page. The getServletInfo method returns a short description.

```
13
14     @Override
15     protected void doGet(HttpServletRequest request, HttpServletResponse response)
16         throws ServletException, IOException {
17
18         int causeId = Integer.parseInt(request.getParameter("causeId"));
19         HttpSession s = request.getSession(true);
20         String cid = causeId+"";
21         s.setAttribute("causeId", cid);
22         Problem p = Model.Problem.getProblem(causeId);
23         String target = "/UpdateStatus2.jsp";
24
25         request.setAttribute("message", p);
26
27         getServletContext().getRequestDispatcher(target).forward(request, response);
28     }
29
30
31     @Override
32     protected void doPost(HttpServletRequest request, HttpServletResponse response)
33         throws ServletException, IOException {
34         response.setContentType("text/html;charset=UTF-8");
35
36         int statusId = Integer.parseInt(request.getParameter("statusId"));
37         HttpSession s = request.getSession(true);
38         Problem p = Model.Problem.getProblem(Integer.parseInt((String)s.getAttribute("causeId")));
39         p.setStatusId(statusId);
40         s.setAttribute("causeId", "");
41         String target = "/UpdateStatus2.jsp";
42
43         request.setAttribute("message", p);
44
45         getServletContext().getRequestDispatcher(target).forward(request, response);
46     }
47
48     @Override
49     public String getServletInfo() {
50         return "Short description";
51     }
52 }
```

Document-SEC02-PROJ1



ผลการดำเนินงาน โดย พีรวัล

- แก้ไข method setStatus ของ class Problem เพื่อใช้เปลี่ยนค่า attribute ของ object นั้นๆ

```
112 public void setStatus(int statusId) {  
113     this.status_ = status;  
114     this.statusId = statusId;  
115     try {  
116         Connection connect = ConnectionBuilder.getConnection();  
117         PreparedStatement ps;  
118         ps = connect.prepareStatement(  
119             "SELECT statusName "  
120             + "FROM status "  
121             + "WHERE statusId = ?");  
122         ps.setInt(1,statusId);  
123         ResultSet result = ps.executeQuery();  
124         while(result.next()){  
125             status_ = result.getString("statusName");  
126         }  
127         ps.close();  
128         connect.close();  
129     } catch (SQLException | ClassNotFoundException e) {  
130         e.printStackTrace();  
131     } catch (Exception e) {  
132         e.printStackTrace();  
133     }  
134 }
```

ผลการดำเนินงาน โดย วิรดา

- แก้ไข method setStatusId ของ class Problem เพื่อใช้เปลี่ยนค่าใน database

```
88 public void setStatusId(int statusId) {  
89     this.statusId = statusId;  
90     try {  
91         Connection connect = ConnectionBuilder.getConnection();  
92         PreparedStatement ps;  
93         ps = connect.prepareStatement(  
94             "UPDATE Cause "  
95             + "SET status_statusId = ? "  
96             + "WHERE causeId = ?");  
97         ps.setInt(1, statusId);  
98         ps.setInt(2, causeId);  
99         int record = ps.executeUpdate();  
100        ps.close();  
101        connect.close();  
102        setStatus(statusId);  
103    } catch (SQLException | ClassNotFoundException e) {  
104        e.printStackTrace();  
105    } catch (Exception e) {  
106        e.printStackTrace();  
107    }  
108 }
```

Feature ที่ 3 : ประกาศ ผู้รับผิดชอบ พิรวัส, ภูริภัทร, วิรดา

ผลการดำเนินงาน โดย พิรวัส

- สร้าง class Announce สำหรับเก็บประกาศจาก database

```

1  public class Announce {
2      private int AnnounceId;
3      private String AnnounceTitle;
4      private String AnnounceDes;
5      private String AnnounceTime;
6
7      public Announce() {
8      }
9
10     public Announce(int AnnounceId) {
11         this.AnnounceId=AnnounceId;
12         try {
13             Connection connect = ConnectionBuilder.getConnection();
14             PreparedStatement ps = connect.prepareStatement(
15                 "SELECT * "
16                 + "FROM Announce "
17                 + "WHERE AnnounceId = ? ");
18             ps.setInt(1,AnnounceId);
19             ResultSet result = ps.executeQuery();
20             while(result.next()){
21                 AnnounceTitle = result.getString("AnnounceTitle");
22                 AnnounceDes = result.getString("AnnounceDes");
23                 AnnounceTime = result.getString("AnnounceTime");
24             }
25         } catch(SQLException | ClassNotFoundException e) {
26             e.printStackTrace();
27         }
28         catch(Exception e) {
29             System.out.println(e.getMessage());
30         }
31     }
32
33     public int ListAnnounce() {...3 lines}
34     public void setAnnounceId(int AnnounceId) {...3 lines}
35     public String getAnnounceTitle() {...3 lines}
36     public void setAnnounceTitle(String AnnounceTitle) {...3 lines}
37     public String getAnnounceDes() {...3 lines}
38     public void setAnnounceDes(String AnnounceDes) {...3 lines}
39     public String getAnnounceTime() {...3 lines}
40     public void setAnnounceTime(String AnnounceTime) {...3 lines}
41
42 }

```

ผลการดำเนินงาน โดย วิรดา

- สร้าง class ListAnnounce สำหรับเก็บประกาศที่เคยมีการแจ้งไว้ใน database โดยแบ่งการทำงานเป็น 2 แบบสำหรับ admin และ student

```

1  public class ListAnnounce {
2      private long userid;
3      private String status;
4      private ArrayList<Announce> list;
5
6      public ListAnnounce() {
7          list = new ArrayList<Announce>();
8      }
9
10     public ListAnnounce(long userId) {
11         this();
12         this.userid = userId;
13         try {
14             Connection connect = ConnectionBuilder.getConnection();
15             PreparedStatement ps = connect.prepareStatement("SELECT userType FROM User WHERE userId = ?");
16             ps.setLong(1,userId);
17             ResultSet result = ps.executeQuery();
18             while(result.next()){
19                 status = result.getString("userType");
20             }
21             ps.close();
22             if(status.equalsIgnoreCase("student")){
23                 ps = connect.prepareStatement(
24                     "SELECT User.userId, Room.roomNo, Dormitory.dormId, Announce.AnnounceId, Announce.announceTime "
25                     + "FROM User "
26                     + "INNER JOIN Renter ON User.userId = Renter.User.userId "
27                     + "INNER JOIN Renter_has_Room ON Renter.renterId = Renter_has_Room.Renter_renterId "
28                     + "INNER JOIN Room ON Renter_has_Room.Room.roomId = Room.roomId "
29                     + "INNER JOIN Dormitory ON Room.Dormitory_dormId = Dormitory.dormId "
30                     + "INNER JOIN Announce ON Dormitory.dormId = Announce.Dormitory_dormId "
31                     + "WHERE User.userId = ? "
32                     + "ORDER BY Announce.announceTime DESC ");
33             }
34         }
35     }
36
37 }

```

Document-SEC02-PROJ1

```
39 else {
40     ps = connect.prepareStatement(
41         "SELECT User.userId, Dormitory.dormId, Announce.announceId, Announce.announceTime "
42         + "FROM User "
43         + "INNER JOIN Dormitory ON User.userId = Dormitory.User_userId "
44         + "INNER JOIN Announce ON Dormitory.dormId = Announce.Dormitory_dormId "
45         + "WHERE userId = ? "
46         + "ORDER BY Announce.announceTime DESC");
47
48     ps.setLong(1,userId);
49     result = ps.executeQuery();
50     while(result.next()) {
51         list.add(new Announce(result.getInt("AnnounceId")));
52     }
53     ps.close();
54     connect.close();
55 }
56 catch(SQLException | ClassNotFoundException e) {
57     e.printStackTrace();
58 }
59 catch(Exception e) {
60     System.out.println(e.getMessage());
61 }
62
63 public long getUserId() {...3 lines}
64 public void setUserId(long userId) {...3 lines}
65 public String getStatus() {...3 lines}
66 public void setStatus(String status) {...3 lines}
67 public ArrayList<Announce> getList() {...3 lines}
68 public void setList(ArrayList<Announce> list) {...3 lines}
69 public static ListAnnounce getListAnnounce(long userId){...4 lines}
70 }
```

ผลการดำเนินงาน โดย ภูริภัทร

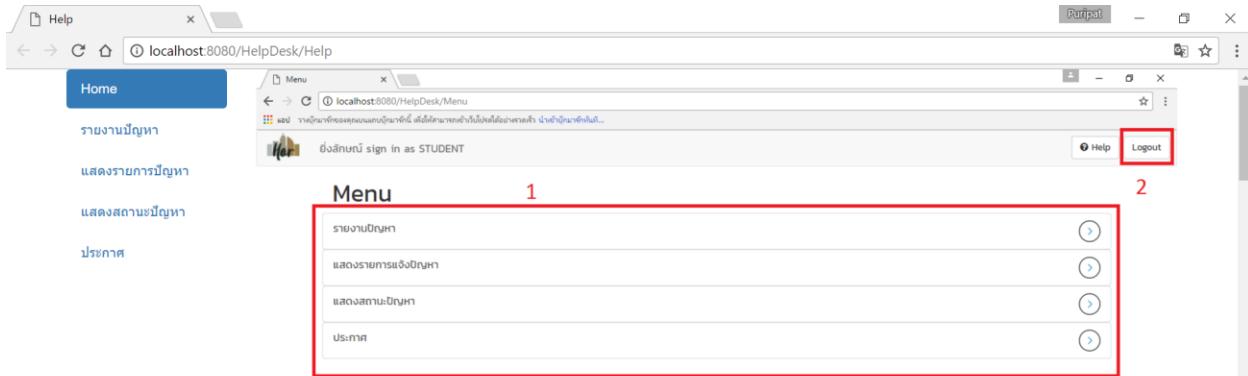
- สร้าง Servlet และ JSP สำหรับคุณมือการใช้งานแต่จะแบ่งการแสดงผลออกเป็น 2 แบบให้กับ student และ admin
 - สร้าง Servlet และ JSP สำหรับประกาศ 2 หน้า
 - สร้าง servlet และ jsp สำหรับเพิ่มประกาศ
 - สร้าง servlet และ jsp สำหรับแก้ไขประกาศ



The screenshot shows the Eclipse IDE interface with the Java editor open. The title bar says "HelpServlet.java". The code editor displays the following Java code:

```
1 package Servlet;
2
3 import Model.User;
4 import java.io.IOException;
5 import java.io.PrintWriter;
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10 import javax.servlet.http.HttpSession;
11
12 public class HelpServlet extends HttpServlet {
13
14     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
15             throws ServletException, IOException {
16         response.setContentType("text/html;charset=UTF-8");
17
18         HttpSession s = request.getSession(true);
19         String target = "/Help.jsp";
20
21         getServletContext().getRequestDispatcher(target).forward(request, response);
22     }
23
24     HttpServlet methods. Click on the + sign on the left to edit the code.
25
26 }
```

Document-SEC02-PROJ1



1. ปุ่มเพื่อเลือกการทำงานที่ต้องการ
2. ปุ่มเพื่อ logout ออกจากระบบ

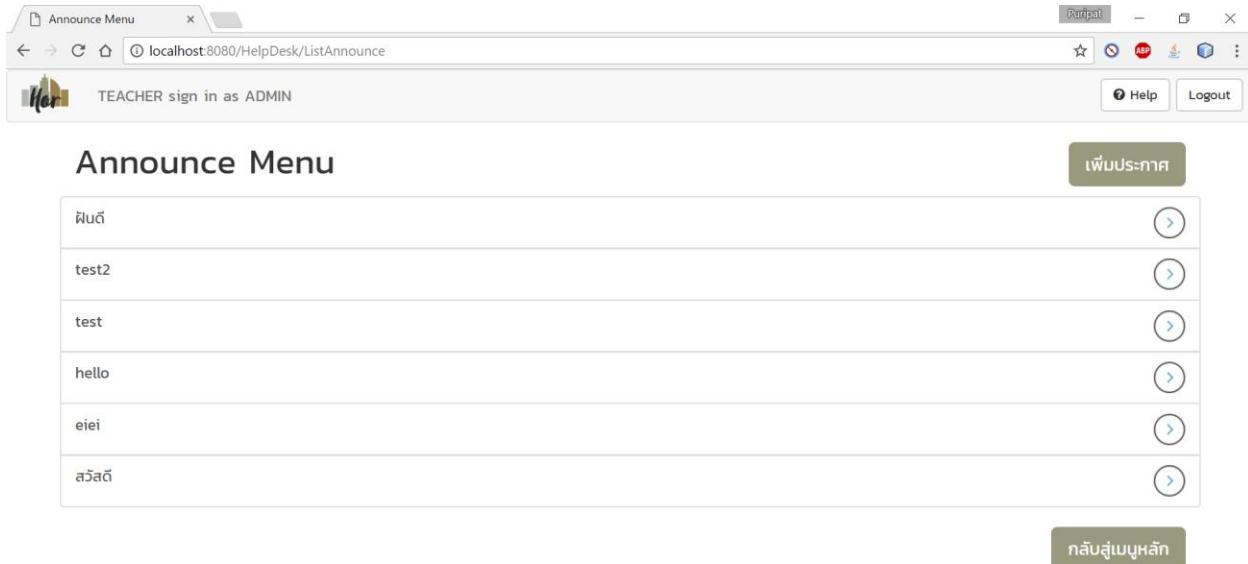


```
ListAnnounceServlet.java
```

```
Source History
```

```
1 package Servlet;
2
3 import Model.ListAnnounce;
4 import Model.User;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import javax.servlet.ServletException;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import javax.servlet.http.HttpSession;
12
13 public class ListAnnounceServlet extends HttpServlet {
14     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
15     throws ServletException, IOException {
16         response.setContentType("text/html;charset=UTF-8");
17
18         HttpSession s = request.getSession(true);
19         String announceId = "0";
20         s.setAttribute("announceId", announceId);
21         User user = (User)s.getAttribute("user");
22         ListAnnounce listA = Model.ListAnnounce.getListAnnounce(user.getUserId());
23         String target = "/AnnounceMenu.jsp";
24
25         request.setAttribute("listAnnounce", listA);
26
27         getServletContext().getRequestDispatcher(target).forward(request, response);
28     }
29 }
```

Document-SEC02-PROJ1



```
16 public class AnnounceFormServlet extends HttpServlet {
17
18     @Override
19     protected void doGet(HttpServletRequest request, HttpServletResponse response)
20             throws ServletException, IOException {
21         response.setContentType("text/html;charset=UTF-8");
22
23         String target = "/AnnounceForm.jsp";
24
25         getServletContext().getRequestDispatcher(target).forward(request, response);
26     }
27
28     @Override
29     protected void doPost(HttpServletRequest request, HttpServletResponse response)
30             throws ServletException, IOException {
31         request.setCharacterEncoding("UTF-8");
32         response.setContentType("text/html;charset=UTF-8");
33
34         String title = request.getParameter("title");
35         String description = request.getParameter("description");
36         SimpleDateFormat sdf = new SimpleDateFormat("YYYY-MM-dd");
37         String date = sdf.format(new Date());
38         int dormId = 0;
39         HttpSession s = request.getSession(true);
40         User user = (User)s.getAttribute("user");
41
42         try {
43             Connection connect = ConnectionBuilder.getConnection();
44             PreparedStatement ps = connect.prepareStatement(
45                     "SELECT User.userid, dormitory.dormId "
46                     + "FROM User "
47                     + "INNER JOIN Dormitory ON User.userId = Dormitory.User_userId "
48                     + "WHERE User.userId = ? ");
49             ps.setLong(1,user.getUserId());
50             ResultSet result = ps.executeQuery();
51             while(result.next()){
52                 dormId = result.getInt("dormId");
53             }
54             ps.close();
55         }
```

Document-SEC02-PROJ1

The screenshot shows a Java development environment and a web browser. The IDE window displays the code for `AnnounceFormServlet.java`, which handles the insertion of announcements into a database and generates a unique announcement ID. The browser window shows the "Announce Form" page, which is accessible at `localhost:8080/HelpDesk/AnnounceForm`. The page includes fields for "Title" and "Description", and buttons for "ADD" and "Back". The user is identified as "TEACHER sign in as ADMIN".

```
55     ps = connect.prepareStatement(
56         "INSERT INTO Announce "
57         + "VALUES (null, ?, ?, ?, ?)");
58     ps.setString(1,description);
59     ps.setString(2,date);
60     ps.setString(3,title);
61     ps.setInt(4,dormId);
62     int record = ps.executeUpdate();
63     ps.close();
64
65     ps = connect.prepareStatement("SELECT MAX(AnnounceId) FROM Announce");
66     result = ps.executeQuery();
67     while(result.next()){
68         String announcedId = ""+result.getInt("MAX(AnnounceId)");
69         s.setAttribute("announcedId", announcedId);
70     }
71     ps.close();
72     connect.close();
73 }
74 catch(SQLException | ClassNotFoundException e){
75     e.printStackTrace();
76 }
77 catch(Exception e){
78     System.out.println(e.getMessage());
79 }
80
81 String target = "/HelpDesk/Announce";
82 response.sendRedirect(target);
83 }
```

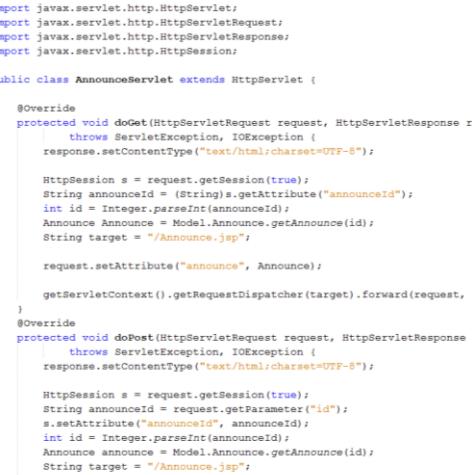
Announce Form

Title :

Description :

ADD **Back**

Document-SEC02-PROJ1



The screenshot shows a Java code editor with the file `AnnounceServlet.java` open. The code implements two methods: `doGet` and `doPost`. Both methods handle `HttpServletRequest` and `HttpServletResponse` objects. They retrieve an announcement by ID from a database and forward the request to a JSP page (`/Announce.jsp`) with the announcement as a parameter. The code uses annotations like `@Override` and `@Override`, and imports various Java and javax.servlet packages.

```
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class AnnounceServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

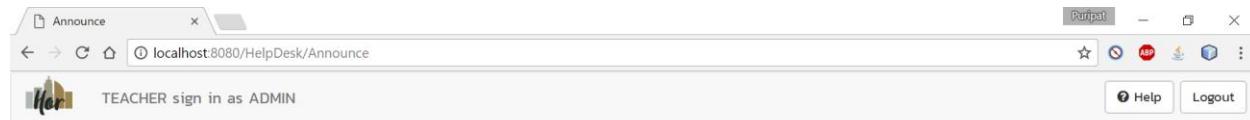
        HttpSession s = request.getSession(true);
        String announceId = (String)s.getAttribute("announceId");
        int id = Integer.parseInt(announceId);
        Announce Announce = Model.Announce.getAnnounce(id);
        String target = "/Announce.jsp";

        request.setAttribute("announce", Announce);
        getServletContext().getRequestDispatcher(target).forward(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        HttpSession s = request.getSession(true);
        String announceId = request.getParameter("id");
        s.setAttribute("announceId", announceId);
        int id = Integer.parseInt(announceId);
        Announce announce = Model.Announce.getAnnounce(id);
        String target = "/Announce.jsp";

        request.setAttribute("announce", announce);
        getServletContext().getRequestDispatcher(target).forward(request, response);
    }
}
```



Announce

អើយខោ : test2

๕๔

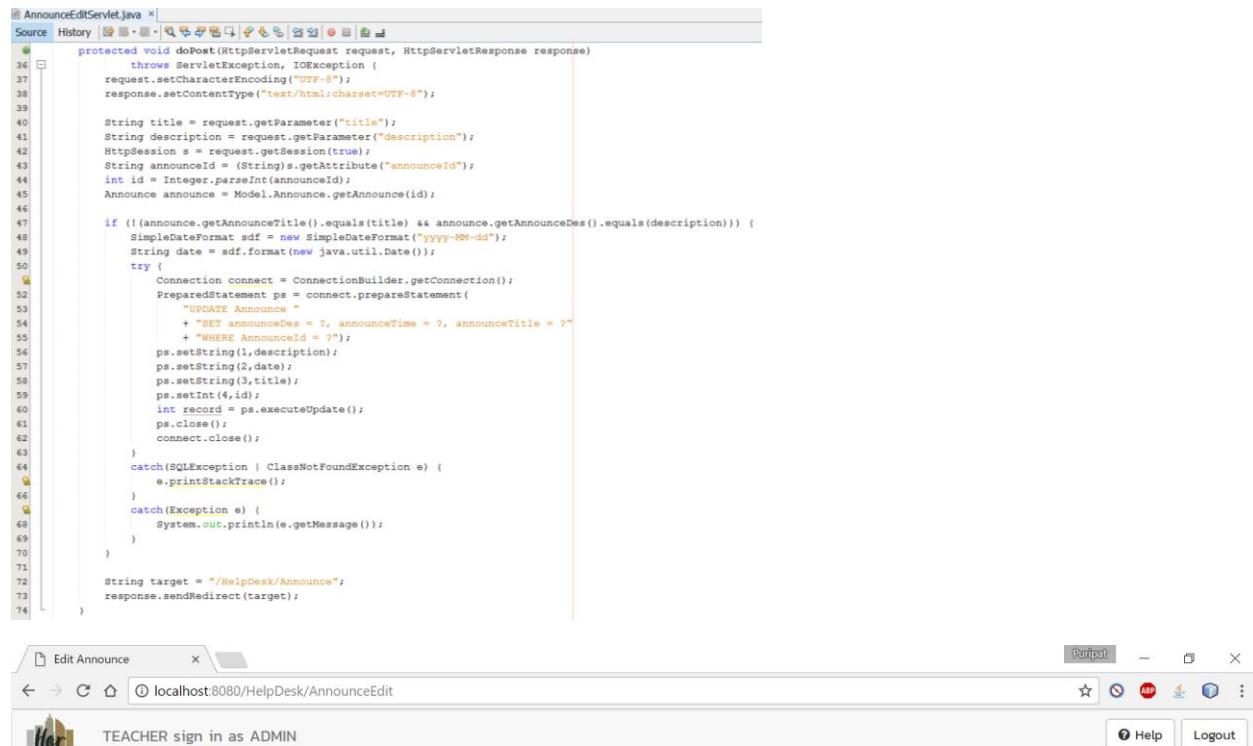
คำอธิบาย :

บัคภาษาไทย

Back

```
16 public class AnnounceEditServlet extends HttpServlet {
17
18     @Override
19     protected void doGet(HttpServletRequest request, HttpServletResponse response)
20             throws ServletException, IOException {
21         response.setContentType("text/html;charset=UTF-8");
22
23         HttpSession s = request.getSession(true);
24         String announceId = (String)s.getAttribute("announceId");
25         int id = Integer.parseInt(announceId);
26         Announce announce = Model.Announce.getAnnounce(id);
27         String target = "/AnnounceEdit.jsp";
28
29         request.setAttribute("announce", announce);
30
31         getServletContext().getRequestDispatcher(target).forward(request, response);
32     }
33 }
```

Document-SEC02-PROJ1

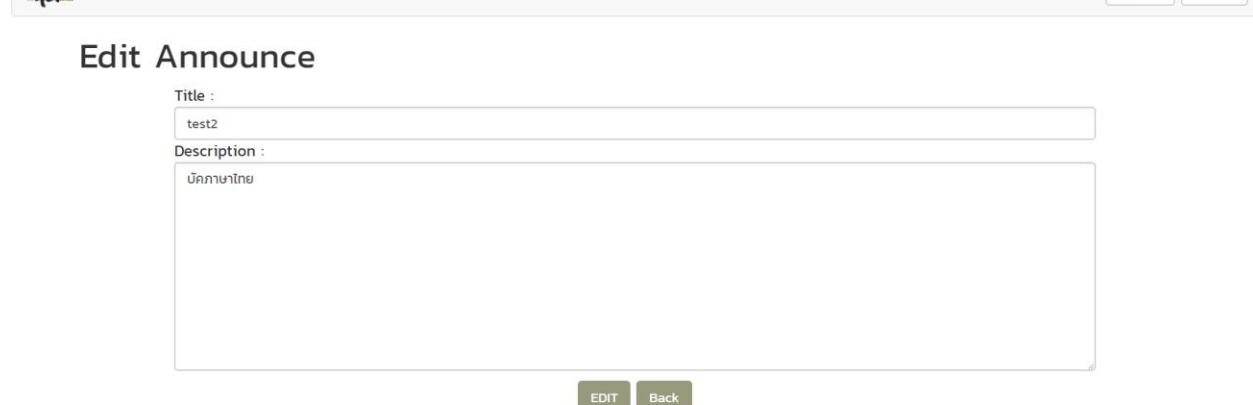


```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html;charset=UTF-8");

    String title = request.getParameter("title");
    String description = request.getParameter("description");
    HttpSession s = request.getSession(true);
    String announceId = (String)s.getAttribute("announceId");
    int id = Integer.parseInt(announceId);
    Announce announce = Model.Announce.getAnnounce(id);

    if (!announce.getAnnounceTitle().equals(title) && announce.getAnnounceDes().equals(description)) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        String date = sdf.format(new java.util.Date());
        try {
            Connection connect = ConnectionBuilder.getConnection();
            PreparedStatement ps = connect.prepareStatement(
                "UPDATE Announce "
                + "SET announceDes = ?, announceTime = ?, announceTitle = ?"
                + "WHERE announceId = ?");
            ps.setString(1,description);
            ps.setString(2,date);
            ps.setString(3,title);
            ps.setInt(4,id);
            int record = ps.executeUpdate();
            ps.close();
            connect.close();
        } catch(SQLException | ClassNotFoundException e) {
            e.printStackTrace();
        }
        catch(Exception e) {
            System.out.println(e.getMessage());
        }
    }

    String target = "/HelpDesk/Announce";
    response.sendRedirect(target);
}
```



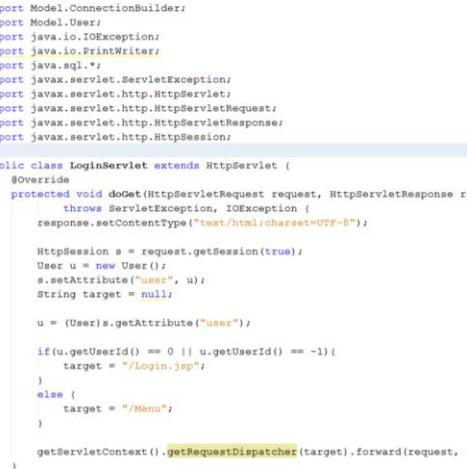
ខ្លួន ។ ដ៏រាបពិទមខុប ភូរិវត្ថុ

ផលការគាំងឱនងាន ໂដឺ ភូរិវត្ថុ

- សរាង Servlet សំអរែប login และ logout
- សំអរែប login servlet ដើម្បីផ្តល់ព័ត៌មានអ្នកប្រើប្រាស់ពេលរក្សាទុក

Document-SEC02-PROJ1

- สร้าง class User ใช้สำหรับเก็บเป็น Session เพื่อนำข้อมูลไปใช้ต่อในหน้าต่าง ๆ
 - สร้าง JSP และ Servlet ที่เป็นหน้ากลางสำหรับเรียกใช้งาน function ต่าง ๆ
 - ทำการเชื่อมหน้า JSP ต่าง ๆ ด้วย Session
 - แก้ไข Servlet ของหน้า JSP ที่ต้องใช้ method Java
 - ตกแต่ง CSS



The screenshot shows the Eclipse IDE interface with the Java code for `LoginServlet.java`. The code implements a servlet for user login, handling both GET and POST requests. It uses JSTL and JavaBeans to manage user session and forward requests to a target page.

```
1 package Servlet;
2
3 import Model.ConnectionBuilder;
4 import Model.User;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import java.sql.*;
8 import javax.servlet.ServletException;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 import javax.servlet.http.HttpSession;
13
14 public class LoginServlet extends HttpServlet {
15     @Override
16     protected void doGet(HttpServletRequest request, HttpServletResponse response)
17             throws ServletException, IOException {
18         response.setContentType("text/html;charset=UTF-8");
19
20         HttpSession s = request.getSession(true);
21         User u = new User();
22         s.setAttribute("user", u);
23         String target = null;
24
25         u = (User)s.getAttribute("user");
26
27         if(u.getUserId() == 0 || u.getUserId() == -1){
28             target = "/Login.jsp";
29         }
30         else {
31             target = "/Menu";
32         }
33
34         getServletContext().getRequestDispatcher(target).forward(request, response);
35     }
36     @Override
37     protected void doPost(HttpServletRequest request, HttpServletResponse response)
38             throws ServletException, IOException {
39         response.setContentType("text/html;charset=UTF-8");
40     }
41 }
```

The screenshot shows a Java code editor with the file `LoginServlet.java` open. The code implements a doPost method to handle user login. It retrieves email and password from the request, prepares a database query to find a matching user, and then sets an attribute on the session and forwards the request to a target page based on the result set. It also handles exceptions and provides a getServletInfo method.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");

    String email = request.getParameter("email");
    String password = request.getParameter("password");

    try {
        HttpSession s = request.getSession();
        Connection connect = ConnectionBuilder.getConnection();
        PreparedStatement ps = connect.prepareStatement("SELECT userId FROM User WHERE userEmail = ? AND password = ?");
        ps.setString(1, email);
        ps.setString(2, password);
        ResultSet result = ps.executeQuery();
        String target = null;

        if(result.isBeforeFirst()){
            User u = new User(-1);
            s.setAttribute("user", u);
            target = "/Login.jsp";
        }
        else {
            while(result.next()){
                User u = new User(result.getLong("userId"));
                s.setAttribute("user", u);
            }
            target = "/Menu";
        }

        getServletContext().getRequestDispatcher(target).forward(request, response);
    }
    catch(Exception e) {
        System.out.println(e.getMessage());
    }
}

@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
```

Document-SEC02-PROJ1

```
public class User {
    private long userId = 0;
    private String status;
    private String name;

    public User() {
    }

    public User(long userId) {
        this.userId = userId;
        if(userId > 0){
            try {
                Connection connect = ConnectionBuilder.getConnection();
                PreparedStatement ps = connect.prepareStatement("SELECT userFirstName, userType FROM User WHERE userid = ?");
                ps.setLong(1, userId);
                ResultSet result = ps.executeQuery();
                while(result.next()){
                    status = result.getString("userType");
                    name = result.getString("userFirstName");
                }
                ps.close();
                connect.close();
            } catch(SQLException | ClassNotFoundException e) {
                e.printStackTrace();
            }
            catch(Exception e) {
                System.out.println(e.getMessage());
            }
        }
    }

    public long getUserId() {...3 lines}
    public void setUserId(long userId) {...3 lines}
    public String getStatus() {...3 lines}
    public void setStatus(String status) {...3 lines}
    public String getName() {...3 lines}
    public void setName(String name) {...3 lines}
    @Override
    public String toString() {...3 lines}
}
```



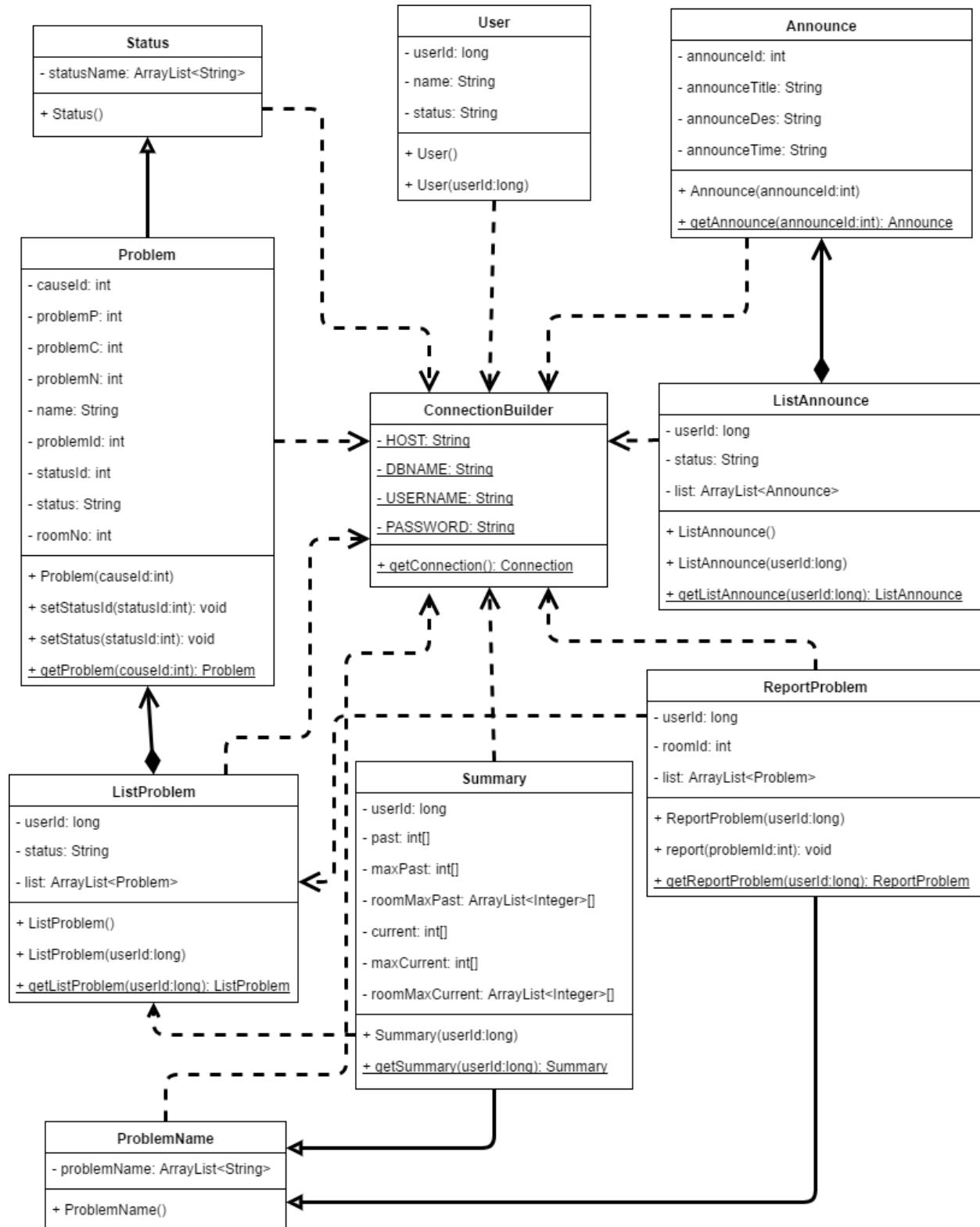
E-mail :	<input type="text"/> Email
Password :	<input type="password"/> Password
<input type="button" value="Login"/>	



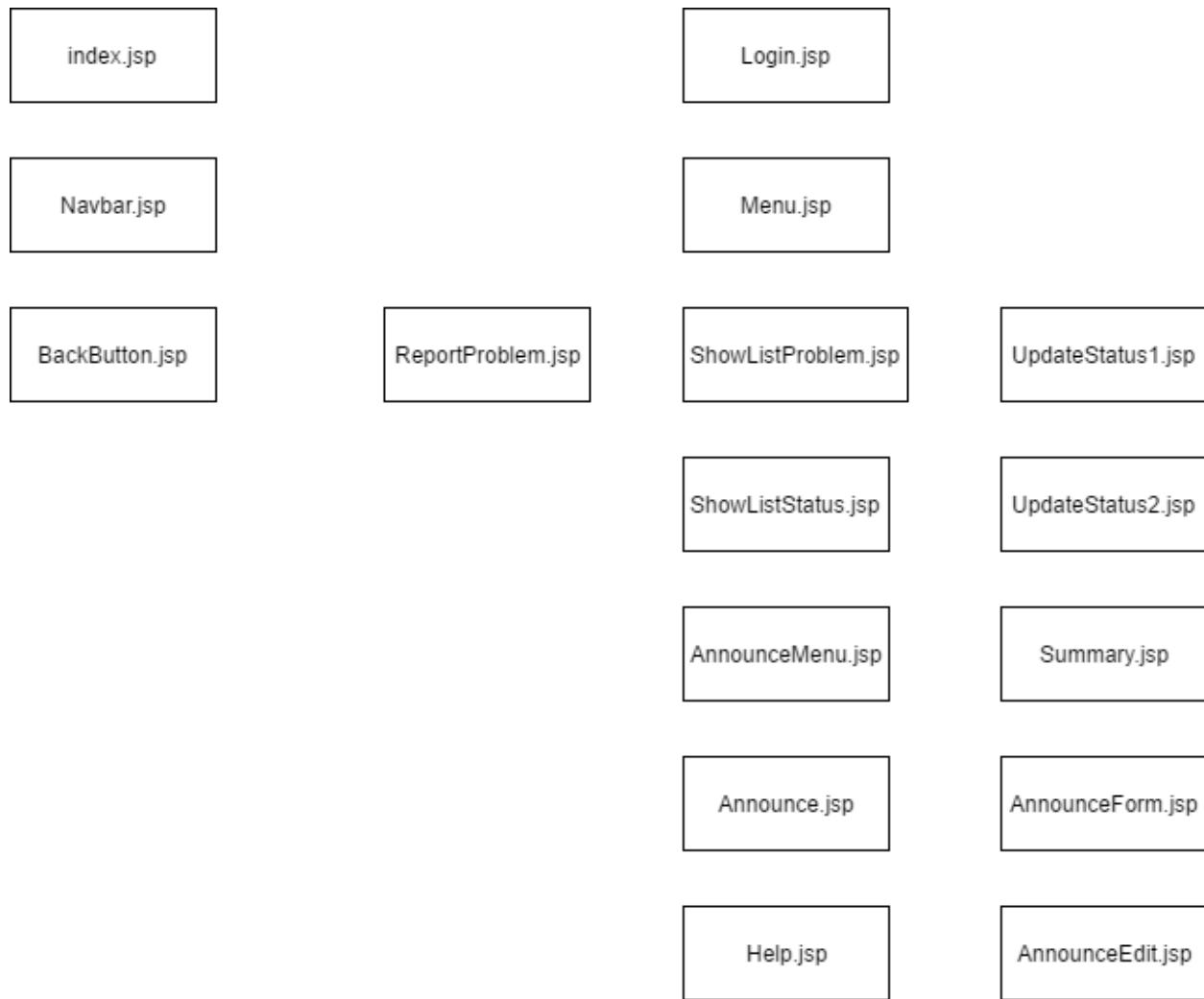
Menu

- รายงานปัจมุหा 
 - แสดงรายการแจ้งปัจมุหा 
 - แสดงสถานะปัจมุหा 
 - ประกาศ 

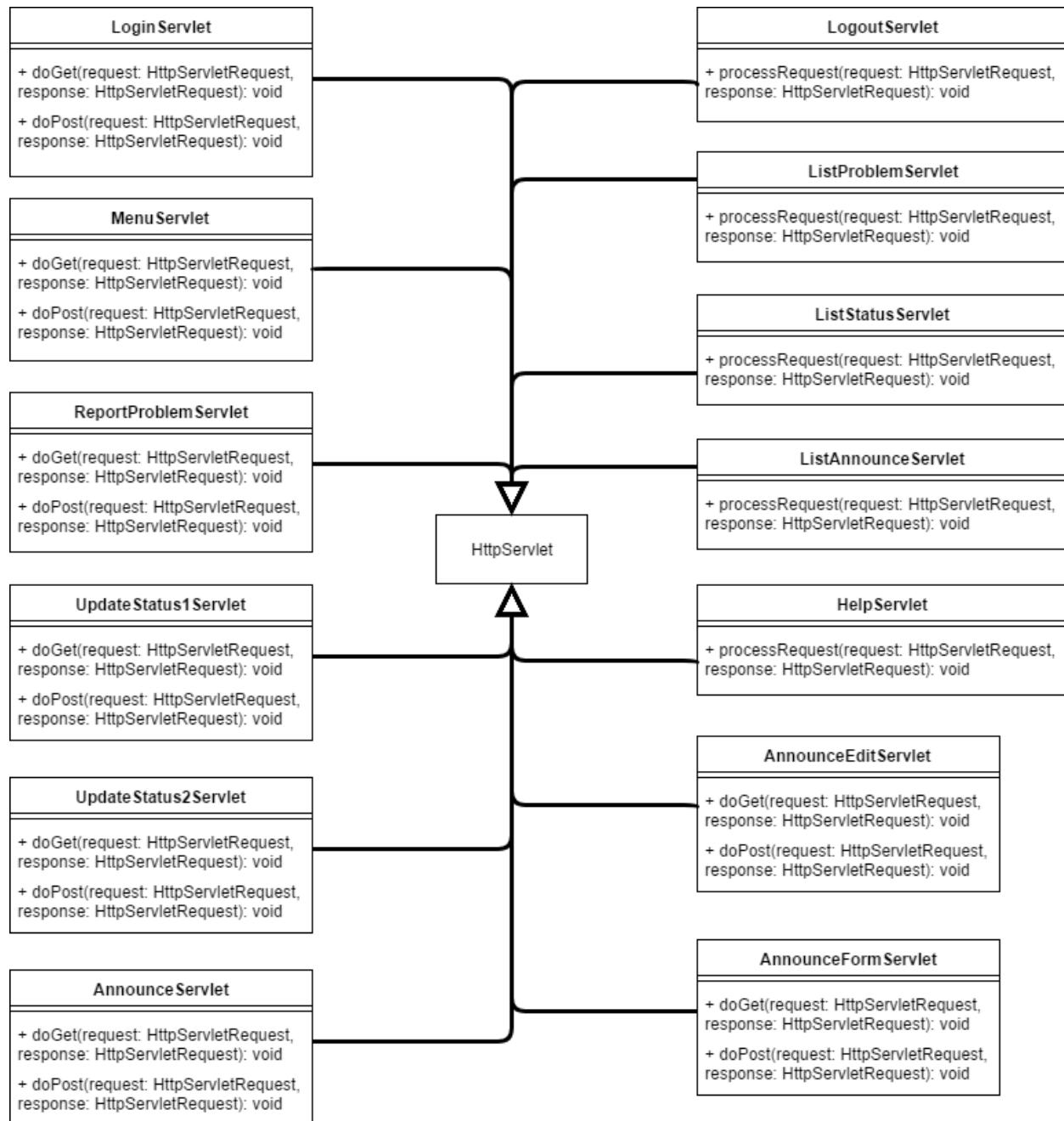
UML Diagram for Model



UML Diagram for View



UML Diagram for Controller



การใช้งาน Feature

1. Function Login

The screenshot shows a web browser window titled "Login". The URL bar displays "Not secure | 10.4.56.26:8080/HelpDesk/Login". The page features a stylized logo with the letters "Her" integrated into a bar chart-like background. Below the logo are two input fields: "Email" and "Password", each with a corresponding icon (user and lock). A "Login" button is located at the bottom right.

ผู้ ADMIN

The screenshot shows a web browser window titled "Menu". The URL bar displays "10.4.56.26:8080/HelpDesk/Menu". The page title is "Menu". It lists several administrative tasks: "แสดงรายการแจ้งปัญหา", "แสดงสถานะปัญหา", "เปลี่ยนแปลงสถานะปัญหา", "สรุปปัญหา", and "ประวัติ". Each item has a circular "View" icon to its right. The top right corner of the page shows "Puripat" and navigation icons.

ผู้ STUDENT

The screenshot shows a web browser window titled "Menu". The URL bar displays "10.4.56.26:8080/HelpDesk/Menu". The page title is "Menu". It lists four tasks: "รายงานปัญหา", "แสดงรายการแจ้งปัญหา", "แสดงสถานะปัญหา", and "ประวัติ". Each item has a circular "View" icon to its right. The top right corner of the page shows "Puripat" and navigation icons.

class ที่เกี่ยวข้อง

- User
- LoginServlet

Document-SEC02-PROJ1

- LogoutServlet
- MenuServlet
- HelpServlet
- ReportProblemServlet
- ListProblemServlet
- ListStatusServlet
- UpdateStatus1Servlet
- UpdateStatus2Servlet
- SummaryServlet
- ListAnnounceServlet

2. Function แจ้งซ่อม

The screenshot shows a web application titled "Report Problem". The URL in the address bar is "10.4.56.26:8080/HelpDesk/ReportProblem". The page displays a table of 15 reported problems:

#	problem name	report
1	ลูกปืนประดู่ช้ำดู	Report
2	หน้าต่างแตก	Report
3	บั้งลมด้านนอก	Report
4	ผ้า/พัดน้ำร้อน	Report
5	WiFi ชำรุด	Report
6	เตาเตียงชำรุด	Report
7	ส้วมอิฟชำรุด	Report
8	ห้องน้ำฝ้าชำรุด	Report
9	แมลงวันชำรุด	Report
10	Remote แอร์ชำรุด	Report
11	TV ชำรุด	Report
12	Remote TV ชำรุด	Report
13	พัดลมชำรุด	Report
14	ถังขยะชำรุด	Report
15	เตียงชำรุด	Report

Document-SEC02-PROJ1

The screenshot shows a web browser window titled "Report Problem" with the URL "10.4.56.26:8080/HelpDesk/ReportProblem". The page displays a list of 15 problem items, each with a "Report" button to its right. A modal dialog box is overlaid on the page, containing a large exclamation mark icon and the text "Are you sure?". Below the dialog are two buttons: "No" (gray) and "Yes" (red).

#	problem name	report
1	อุกบิดปะ:ցဲ့ရှုံ	စောင်ပြုမာ
2	မပါတော်မော်	ကျိုးပြုပြုမာ
3	ပွဲ့ဗောင်္ဂလာ	Report
4	မီာ/ဖတေနံရှုံ	Report
5	WiFi ဦးရှုံ	Report
6	ပေါ်စီးပွဲ့ရှုံ	Report
7	ဆိုဒ်ပါ့ဗဲ့ရှုံ	Report
8	မလောက်ဖော်ရှုံ	Report
9	အော်ဦးရှုံ	Report
10	Remote အော်ဦးရှုံ	Report
11	TV ဦးရှုံ	Report
12	Remote TV ဦးရှုံ	Report
13	ပိုလော်ရှုံ	Report
14	ဂျူးခိုးရှုံ	Report
15	တော်ဦးရှုံ	Report

class ที่เกี่ยวข้อง

- User
- ReportProblem
- MenuServlet
- HelpServlet
- LogoutServlet

3. Function แสดงรายการปัญหา

ฟัง ADMIN

The screenshot shows a web browser window titled "Show List Problem" with the URL "10.4.56.26:8080/HelpDesk>ShowListProblem". The page displays a list of 5 problem items, each with its room number listed to the right. A button labeled "กลับสู่เมนูหลัก" is located at the bottom right of the page.

#	problem name	room
1	อุกบิดပြီးցဲ့ရှုံ	1
2	မပါတော်မော်	1
3	อุกบิดပြီးցဲ့ရှုံ	2
4	Remote အော်ဦးရှုံ	2
5	ရော်ပေါ်	2

ฟัง STUDENT

Document-SEC02-PROJ1

The screenshot shows a web browser window titled "Show List Problem" with the URL "10.4.56.26:8080/HelpDesk>ShowListProblem". The page title is "List Problem". A table lists two problems:

#	problem name
1	ຊຸກບົດປະດູບ່າຮຸດ
2	ໜ້າຕ່າງແດກ

A large green button labeled "ກລັບສຸມບຸກ" with a mouse cursor icon is visible on the right side of the page.

class ທີ່ເກີຍວ່າຂອງ

- User
- ListProblem
- MenuServlet
- HelpServlet
- LogoutServlet

4. Function ແສດງສຕານະປໍ່ມູ່ຫາ

ຝຶ່ງ ADMIN

The screenshot shows a web browser window titled "List Status" with the URL "10.4.56.26:8080/HelpDesk>ShowListStatus". The page title is "List Status". A table lists six problems with their room numbers and status:

#	problem name	room	status
1	ຊຸກບົດປະດູບ່າຮຸດ	1	ສະດ້າບິນການ
2	ໜ້າຕ່າງແດກ	1	ກ່າວສົດ່າບິນການ
3	ຮລອດໄຟບ່າຮຸດ	1	ເສື່ອສັນ
4	ຊຸກບົດປະດູບ່າຮຸດ	2	ກ່າວສົດ່າບິນການ
5	Remote ແລ້ວເປົ້າຮຸດ	2	ສະດ້າບິນການ
6	ກ່ອປ່າສົ່ງ	2	ສະດ້າບິນການ

A large green button labeled "ກລັບສຸມບຸກລັກ" with a mouse cursor icon is visible on the right side of the page.

ຝຶ່ງ STUDENT

The screenshot shows a web browser window titled "List Status" with the URL "10.4.56.26:8080/HelpDesk>ShowListStatus". The page title is "List Status". A table lists three problems with their room numbers and status:

#	problem name	status
1	ຊຸກບົດປະດູບ່າຮຸດ	ສະດ້າບິນການ
2	ໜ້າຕ່າງແດກ	ກ່າວສົດ່າບິນການ
3	ຮລອດໄຟບ່າຮຸດ	ເສື່ອສັນ

A large green button labeled "ກລັບສຸມບຸກລັກ" with a mouse cursor icon is visible on the right side of the page.

class ที่เกี่ยวข้อง

- User
- ListProblem
- MenuServlet
- HelpServlet
- LogoutServlet

5. Function เปลี่ยนสถานะปัญหา

The screenshots illustrate the 'Update Status' feature of a web application. The application is running on a local host at 10.4.56.26:8080/HelpDesk/UpdateStatus1.

Screenshot 1: Update Status - List View

#	problem name	room	status	Action
1	ลูกบินประดู่ช่ารุด	1	สอนปีบ่มาร์	Update
2	หน้าต่างแตก	1	กำลังดำเนินการ	Update
3	หลอดไฟชำรุด	1	เสร็จสิ้น	Update
4	ลูกบินประดู่ช่ารุด	2	กำลังดำเนินการ	Update
5	Remote แบตหมด	2	สอนปีบ่มาร์	Update
6	ท่อปัสสาวะ	2	สอนปีบ่มาร์	Update

Screenshot 2: Update Status - Room 1 Detail

Room : 1

Problem Name : ลูกบินประดู่ช่ารุด
 Status : สอนปีบ่มาร์
 Update Status :

[Update](#) [Back](#)

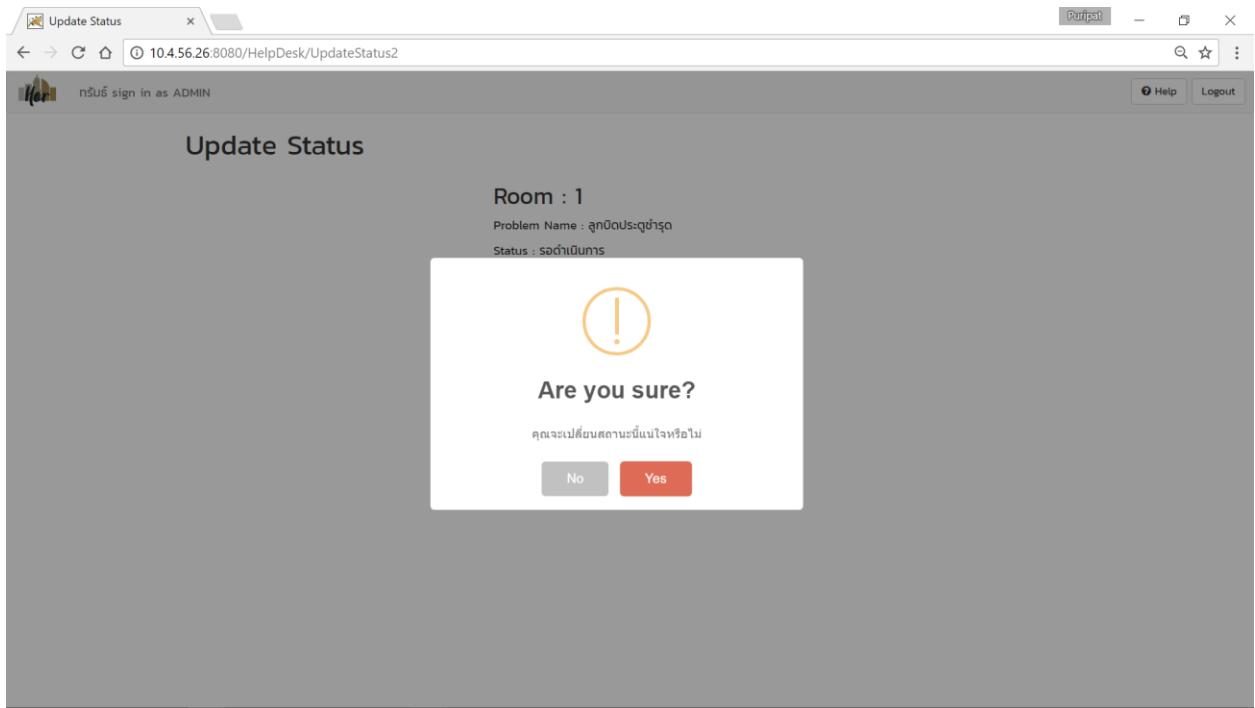
Screenshot 3: Update Status - Room 1 Detail (with dropdown arrow)

Room : 1

Problem Name : ลูกบินประดู่ช่ารุด
 Status : สอนปีบ่มาร์
 Update Status :

[Back](#)

Document-SEC02-PROJ1



class ទี่កើតឡើខ្លួន

- User
- ListProblem
- Problem
- UpdateStatus1Servlet
- UpdateStatus2Servlet
- MenuServlet
- HelpServlet
- LogoutServlet

6. Function ស្រួលប្រើប្រាស់

Document-SEC02-PROJ1

Screenshot of a web application showing a summary report for the year 2560. The report lists various problems and their counts.

The top screenshot shows the initial state where the year is set to 2560. The bottom screenshot shows a dropdown menu where the year is being changed from 2560 to 2559, with an arrow pointing to the selection.

Summary Report (Year 2560):

#	problem name	count
1	อุกบิดประดุจชุด	4
2	หน้าต่างแตก	1
3	บุ้งอวบน้ำด	0
4	ฝ้า/เพดานชำรุด	0
5	WiFi ชำรุด	0
6	เด้าสีขอบชำรุด	0
7	สวักอิฟชำรุด	0
8	หลอดไฟชำรุด	3
9	แอร์ชำรุด	0
10	Remote แอร์ชำรุด	1
11	TV ชำรุด	0
12	Remote TV ชำรุด	0
13	พัดลมชำรุด	0
14	ตู้เย็นชำรุด	0
15	เตียงชำรุด	0
16	ผูกต่ำด	0
17	ตู้เสื้อผ้าชำรุด	0
18	โต๊ะชำรุด	0

Document-SEC02-PROJ1

#	problem name	room	count
1	สุกเป็นประดุจารุด	1	2
2	หมาดังเด็ก	1	1
3	บึงลวดมาก	ไม่มี	-
4	ผ้า/เพตานชำรุด	ไม่มี	-
5	WiFi ชำรุด	ไม่มี	-
6	เต้าเสียบชำรุด	ไม่มี	-
7	สวีกปั๊ฟชำรุด	ไม่มี	-
8	หลอดไฟชำรุด	1	3
9	แอร์ชำรุด	ไม่มี	-
10	Remote แวร์ชำรุด	2	1
11	TV ชำรุด	ไม่มี	-
12	Remote TV ชำรุด	ไม่มี	-
13	พัดลมชำรุด	ไม่มี	-
14	hor-helpdesk	ไม่มี	-

class ที่เกี่ยวข้อง

- User
- Summary
- SummaryServlet
- MenuServlet
- HelpServlet
- LogoutServlet

7. Function ประมวลผล

ฟัง ADMIN

Announce Menu

ดูไฟ	เพิ่มประกาศ
ข้อมูลไฟ	(>)
ดูไฟ	(>)

กู้ลับสูญเสียหลัก

Document-SEC02-PROJ1

The image consists of three vertically stacked screenshots of a web-based application titled "Announce Form".

Screenshot 1: Shows the initial "Announce Form" page. It has two input fields: "Title:" containing "ทดสอบ" and "Description:" containing "ส่งรายงานผลติดต่อภายใน". Below the fields are "ADD" and "Back" buttons.

Screenshot 2: Shows the same page after entering data. A modal dialog box appears in the center, asking "Are you sure?". The message below it reads: "คุณจะเพิ่มประกาศชื่นใจหรือไม่" and "Title : ทดสอบ". There are "No" and "Yes" buttons at the bottom of the dialog.

Screenshot 3: Shows the result of the action. The page now displays the title "Announce" and the content "หัวข้อ : ตัดไฟ" and "คำอธิบาย : รับปรับปรุงระบบไฟฟ้า วันที่ 10 พ.ค. 60". A "แก้ไข" (Edit) button with an edit icon is visible on the right side of the content area. A "Back" button is located at the bottom of this screen.

Document-SEC02-PROJ1

Screenshot of the 'Edit Announce' page for an ADMIN user.

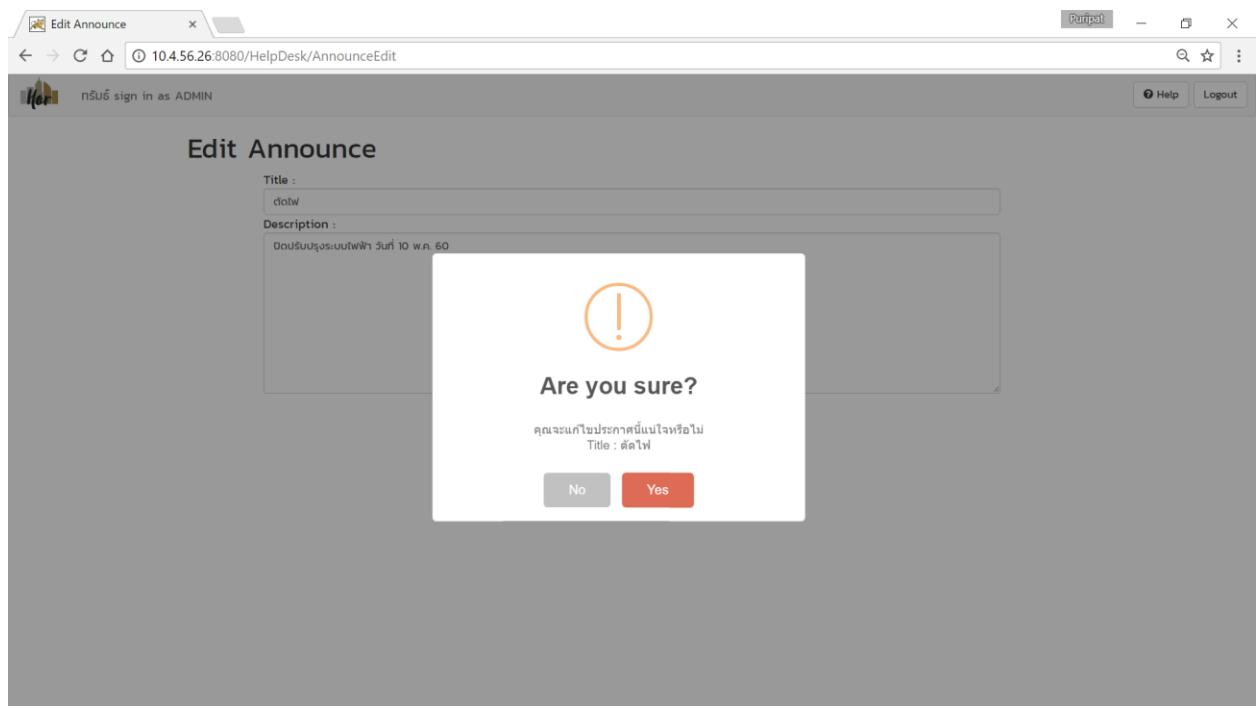
The URL is 10.4.56.26:8080/HelpDesk/AnnounceEdit.

The page title is 'Edit Announce'.

Form fields:

- Title: ดูดิว
- Description: ចុះបែនប្រចាំខែ ខែធ្នូ ឆ្នាំ ២០២៣

Buttons: EDIT, Back



Screenshot of the 'Announce Menu' page for a STUDENT user.

The URL is 10.4.56.26:8080/HelpDesk/ListAnnounce.

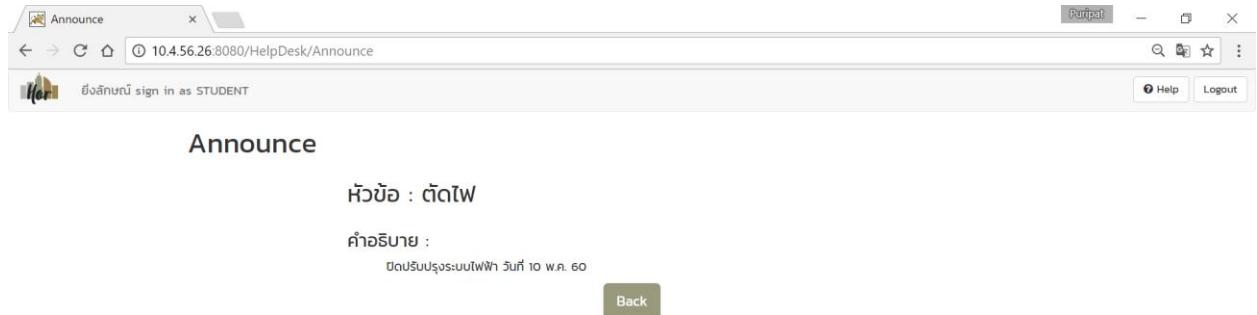
The page title is 'Announce Menu'.

Table of announcements:

ការបែងចែក	លទ្ធផល
ដឹកជញ្ជូន	
ចំណាំអាជីវកម្ម	
សំណើ	

Buttons: កត់ស្តីរបស់ខ្លួន

Document-SEC02-PROJ1



class ទីកើតឡើង

- User
- ListAnnounce
- Announce
- ListAnnounceServlet
- AnnounceServlet
- AnnounceFormServlet
- AnnounceEditServlet
- MenuServlet
- HelpServlet
- LogoutServlet

8. Function គ្មានការិយាល័យ

ដោយ ADMIN

Document-SEC02-PROJ1

The screenshot shows a web browser window with two tabs open. The left tab is titled 'Home' and lists navigation options: 1. ផែនទីដើមការរាយការណ៍ទំនាក់ទំនង, 2. ផែនទី logout ចូលការទួន្នុ. The right tab is titled 'Menu' and displays a list of menu items: 1. អគ្គនាយករាយការណ៍បិបុណ្ណា, 2. អគ្គសេខាបិបុណ្ណា, 3. អធិសេបនលកោណ៍បិបុណ្ណា, 4. អតិថិជន, 5. ប្រភាគ. A red box highlights the 'Logout' button in the top right corner of the right tab.

The screenshot shows a 'List Problem' page with a table. The table has columns: #, problem name, and room. The data is as follows:

#	problem name	room
1	ក្នុងបុណ្ណោះ	1
2	អំពីរាយ	1
3	ក្នុងបុណ្ណោះ	2

A red box highlights the '#', 'problem name', and 'room' columns in the table.

ផ្ទៃ STUDENT

The screenshot shows a 'Menu' page with a list of menu items: 1. រាយការណ៍បិបុណ្ណា, 2. អគ្គនាយករាយការណ៍បិបុណ្ណា, 3. អគ្គសេខាបិបុណ្ណា, 4. អគ្គសេខាបិបុណ្ណា, 5. ប្រភាគ. A red box highlights the 'Logout' button in the top right corner of the right tab.

The screenshot shows a 'Report Problem' page with a table. The table has columns: #, problem name, report, and status. The data is as follows:

#	problem name	report	status
1	ក្នុងបុណ្ណោះ	សិក្សានៅ	ក្នុងបុណ្ណោះ
2	អំពីរាយ	សិក្សានៅ	ក្នុងបុណ្ណោះ

A red box highlights the '#', 'problem name', 'report', and 'status' columns in the table.

class ទី 2 ត្រូវបានបញ្ជាក់

- HelpServlet