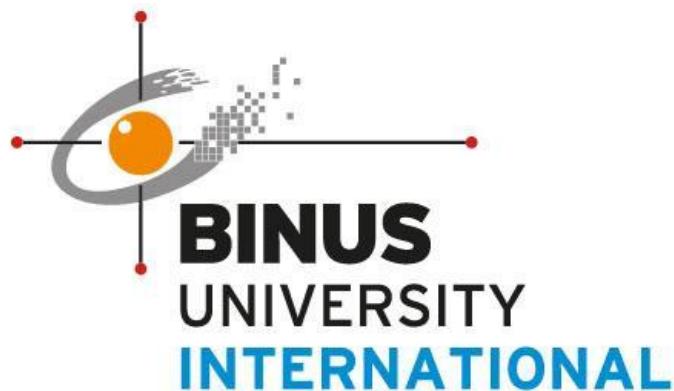


Final Project Report
“School Management System”
Object Oriented Programming

Lecturer:

Jude Joseph Lamug Martinez



Tirza Gabriella

2602109870

CLASS L2AC

A handwritten signature in black ink that appears to read "Tirza".

Computer Science Major
Binus International University
2022

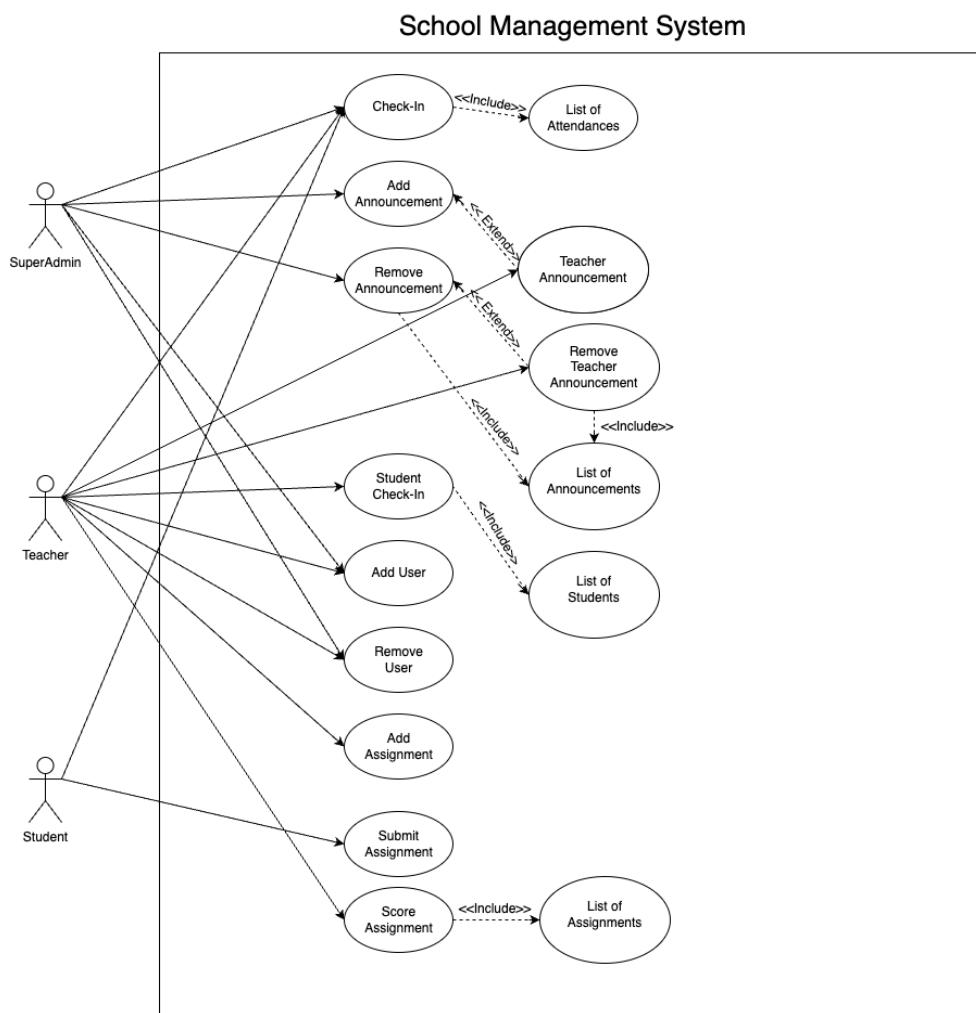
Table of Contents

I. Brief Description	3
II. Use Case Diagram	3
III. Activity Diagram	4
IV. Class Diagram	5
V. DataBase	6
VI. Data Structure Used	8
VII. How It Works	10
VIII. Screenshot of Application	10
IX. Lesson Learned	18
a. Error Handling	18
b. Reflection	18

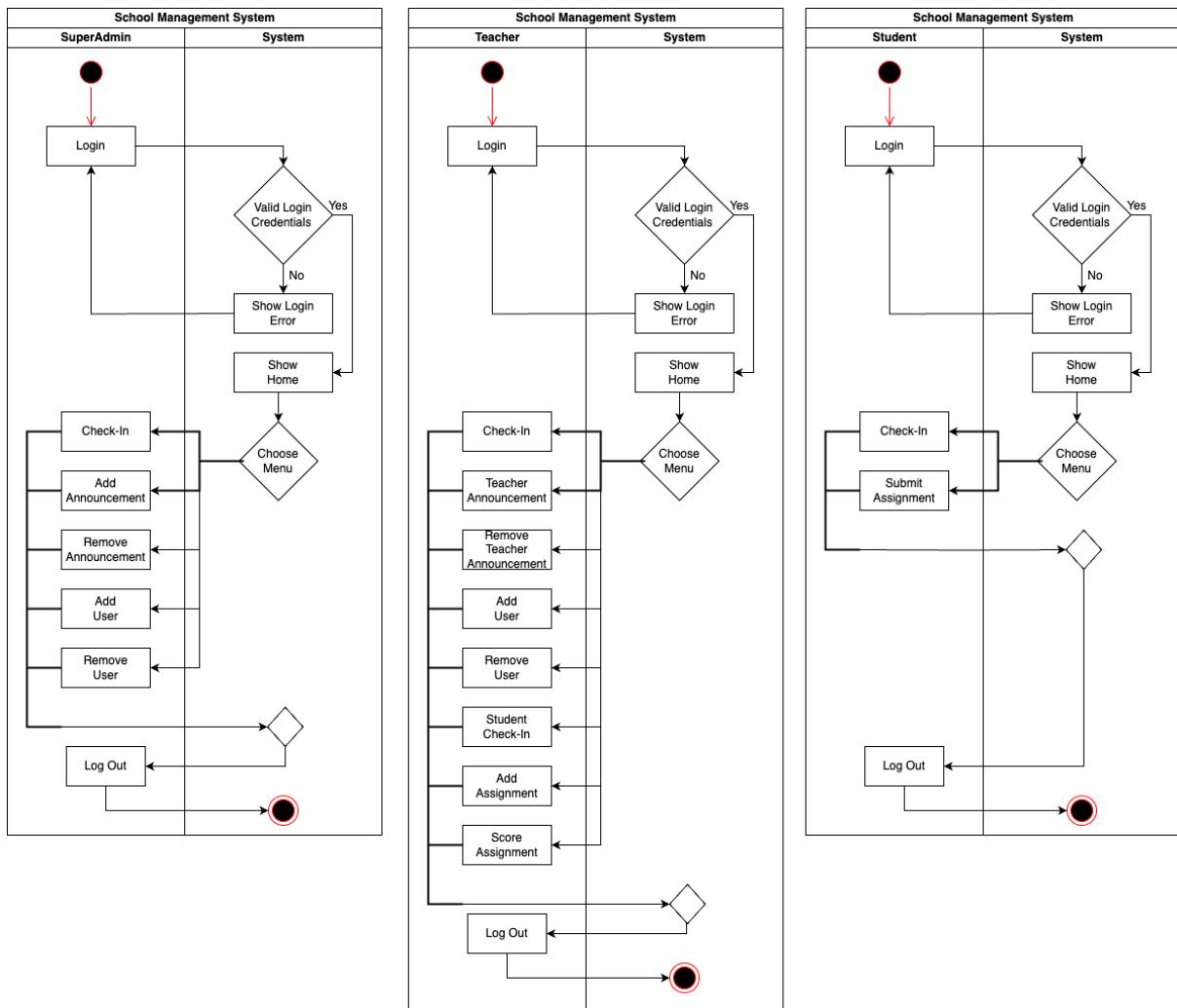
I. Brief Description

For my final project assignment, I have opted to develop a program called "School Management System". Prior to selecting this project, I faced considerable uncertainty regarding the most suitable topic that aligns with my skills. I aimed for a project that strikes a balance between complexity and simplicity while meeting the assignment criteria. After careful consideration, I settled on "School Management System". Essentially, it is my own interpretation of many management systems out there. I chose this because it allows for the implementation of fundamental concepts such as classes, inheritance, interface, GUI, Database, functions, instance variables, and objects. Moreover, it aligns perfectly with my knowledge of Object-Oriented in Java.

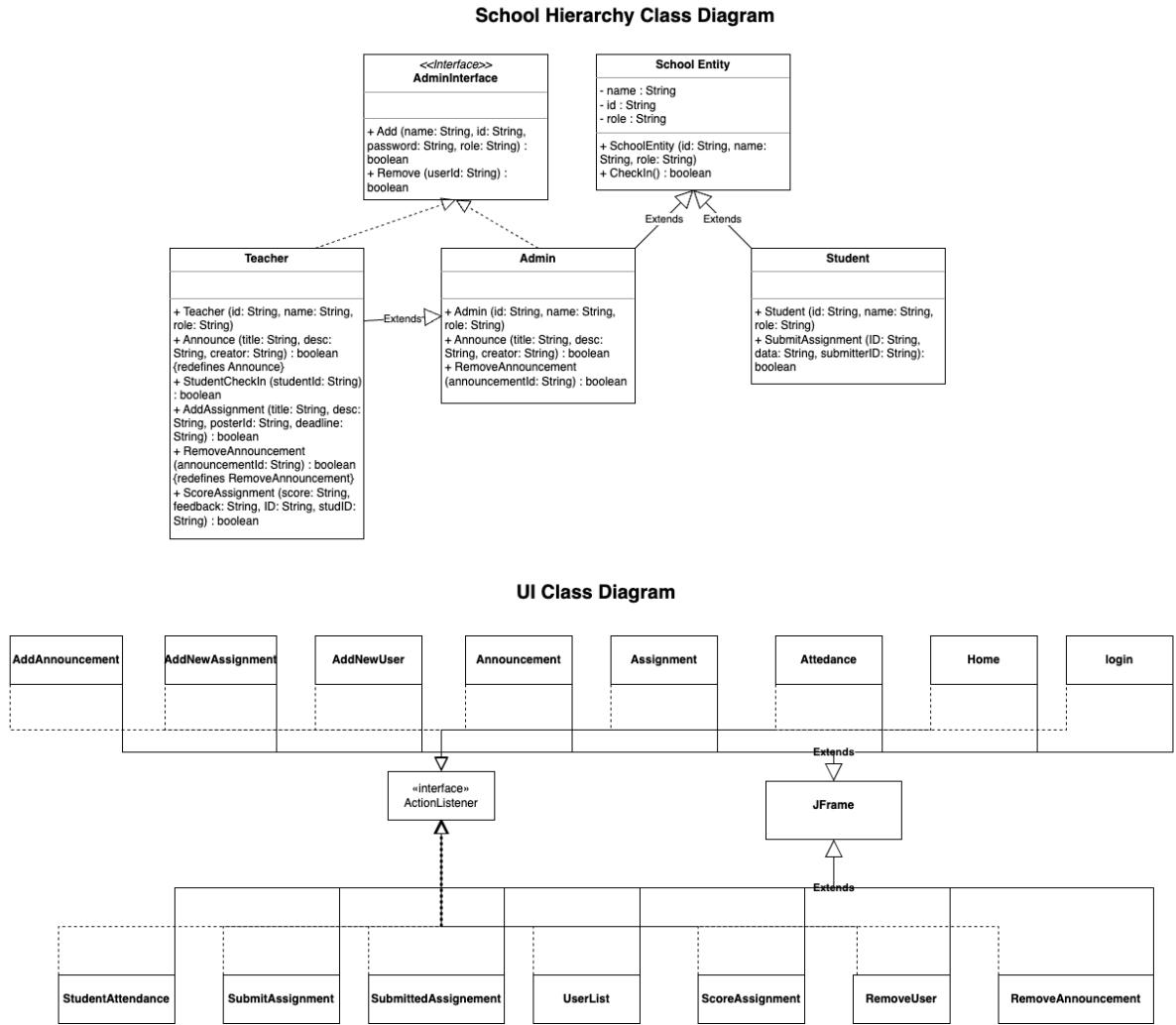
II. Use Case Diagram



III. Activity Diagram



IV. Class Diagram



School Hierarchy Class Diagram

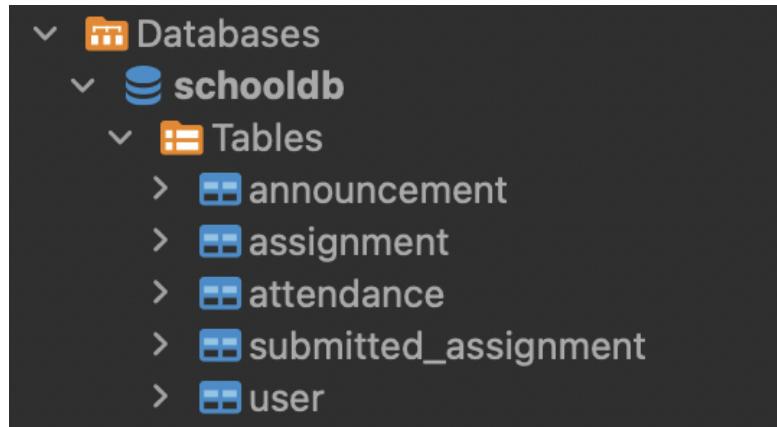
In this project, the Admin and Student class/object inherits the SchoolEntity class as they all use the same CheckIn function. The Admin and Teacher class implements AdminInterface interface, as they both implement Add and Remove methods. The Teacher class inherits Admin class as its functions are similar to Admin (with the addition of functions such as AddAssignment and RemoveAssignment). In Teacher class there are two overridden methods (polymorphism) namely Announcement and RemoveAnnouncement. Teacher (subclass) provides its own implementation of both methods that is already defined in Admin (superclass).

UI Class Diagram

All UI classes implement the ActionListener interface and inherit the JFrame class.

V. DataBase

In this project I'm using MySQL as the DataBase.



First I have the announcement table. Announcement table consists of announcement_title (varchar), announcement_description (varchar), user_id (varchar), , announcement_id (int).

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra	Expression	Comment
announcement_title	1	varchar(100)	[v]						
announcement_description	2	varchar(1000)	[]						
user_id	3	varchar(100)	[v]						
announcement_id	4	int	[v]		PRI		auto_increment		

The assignment table consists of assignment_title (varchar), assignment_desc (varchar), teacher_id (varchar), assignment_deadline (datetime), assignment_id (int).

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra	Expression	Comment
assignment_title	1	varchar(100)	[v]						
assignment_desc	2	varchar(100)	[v]						
teacher_id	3	varchar(100)	[v]						
assignment_deadline	4	datetime	[v]						
assignment_id	5	int	[v]		PRI		auto_increment		

The attendance table consists of user_id (varchar), checkin_datetime (varchar).

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra	Expression	Comment
user_id	1	varchar(100)	[v]						
checkin_datetime	2	varchar(100)	[v]						

The submitted_assignment table consists of assignment_id(int), assignment_data(varchar), assignment_status(varchar), assignment_score (varchar), user_id (varchar), assignment_feedback (varchar).

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra	Expression	Comment
assignment_id	1	int	[v]		[]				
assignment_data	2	varchar(100)	[v]		[]				
assignment_status	3	varchar(100)	[]		[]				
assignment_score	4	varchar(100)	[]		[]				
user_id	5	varchar(100)	[v]		[]				
assignment_feedback	6	varchar(500)	[]		[]				

The user table consists of user_name (varchar), user_id (varchar), user_role (varchar), user_password (varchar).

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra	Expression	Comment
user_name	1	varchar(100)	[v]		[]				
user_id	2	varchar(100)	[v]		[]	PRI			
user_role	3	varchar(100)	[v]		[]				
user_password	4	varchar(100)	[v]		[]				

VI. Data Structure Used

The data structures used in this project are as follows :

1. **Hashmap** : Hashmap is used at the Submit Assignment menu, where it is used to store the assignments available and to be later scanned, to see if the inputted assignment id exists when Students are submitting an assignment.

The screenshot shows a portion of Java code in an IDE. A red box highlights the following code block:

```
// clear assignments map
if(assignmentsMap != null){
    assignmentsMap.clear();
} else {
    assignmentsMap = new HashMap<>();
}

// insert the assignment data to table
while(rs.next()){
    Object[] data = {
        rs.getString( columnLabel: "assignment_id"),
        rs.getString( columnLabel: "assignment_title"),
        rs.getString( columnLabel: "assignment_desc"),
        rs.getString( columnLabel: "teacher"),
        rs.getString( columnLabel: "assignment_deadline"),
    };

    tableModel.addRow(data);
}

assignmentsMap.put(rs.getString( columnLabel: "assignment_id"),rs.getString( columnLabel: "assignment_title"));
```

A red box also highlights the line `assignmentsMap.put(rs.getString(columnLabel: "assignment_id"),rs.getString(columnLabel: "assignment_title"));`. The text "Clear or Initiate Hashmap" is displayed above the first highlighted block, and "Add new element to Hashmap" is displayed below the second highlighted block.

The screenshot shows a portion of Java code in an IDE. A red box highlights the following code block:

```
if(assignmentsMap.containsKey(ID)){
    Student std = new Student(currentEntity.id, currentEntity.name, currentEntity.role);
    if(std.SubmitAssignment(ID, data, currentEntity.id)) {
        // return to submitted assignments page
        new SubmittedAssignments(this.currentEntity, this.assignmentsMap).setVisible(true);
        this.setVisible(false);
    }
} else {
    JOptionPane.showMessageDialog( parentComponent: null, message: "Assignment doesn't exist!");
}
```

2. **Two Dimensional Array** : Tables in this Java project uses DefaultTableModel, a concrete implementation of the TableModel interface that uses a two-dimensional array of objects as its underlying data structure. The array stores each data in the table's cells, where both rows and columns are represented by indexes. The first array dimension represents the Rows, and the second dimension represents the Columns. Each cell can contain an object of any type. Aside from those, DefaultTableModel also has a separate array that stores the column names and row identifiers (if any)

```
// fetch attendance data
tableModel = new DefaultTableModel(tableHeader, rowCount: 0);
getAttendanceData();
```

```
// reset table
tableModel.setRowCount(0);

// fill table with data
while(rs.next()){
    Object[] data = {
        rs.getString(columnLabel: "user_name"),
        rs.getString(columnLabel: "user_id"),
        rs.getString(columnLabel: "user_role"),
        rs.getString(columnLabel: "checkin_datetime")
    };

    tableModel.addRow(data);
}

// update attendance table data and repaint
attendanceTable = new JTable(tableModel);
attendanceTable.repaint();
```

VII. How It Works

The program's navigation is mainly done by selecting menu buttons. Each user with a different role has different menus provided to them. Some of the menus, albeit similar, have different usages depending on the role (For example, teacher can only add students, admin can add users of any role). Aside from navigating through pages, this program has menus that enables the user to add new data, remove data, and to see all the data regarding a certain topic.

The main class hierarchy of this app consists of the classes SchoolEntity, Admin, Teacher, and Student. These classes are used to represent the entity that is logged in to the app. Each of these different user class types also provide different functions that will later be implemented in the UI to add or remove data.

VIII. Screenshot of Application

First, we need to run the program from the “splash” class, and it will pop up an image of the project name, the creator of the program and details about the creator like the picture shown below (Image 1.1).



Image 1.1 Splash

Next, a login page will appear. The user can insert their name and password that the Teacher or SuperAdmin already created before. (Image 1.2)

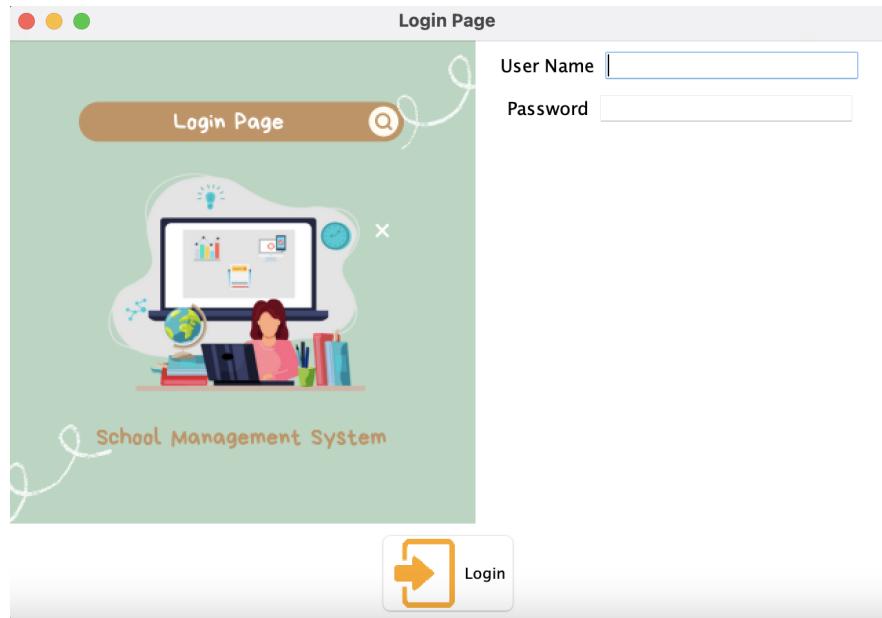


Image 1.2 Login

If the login is successful, a pop up message “Successfully logged in!!! Welcome (your username)” will appear. But if the user and password does not exist. Then a pop up message “Invalid Login” will appear. (Image 1.3, Image 1.4)

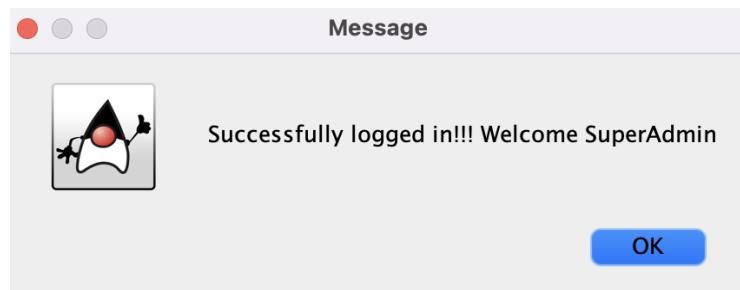


Image 1.3 Pop-up Message



Image 1.4 Pop-Up Message Invalid

If we login as the SuperAdmin, there will be a menu feature that we can see in the UI. And below the program there's a button which we can choose which features the SuperAdmin is going to use. (Image 1.5)

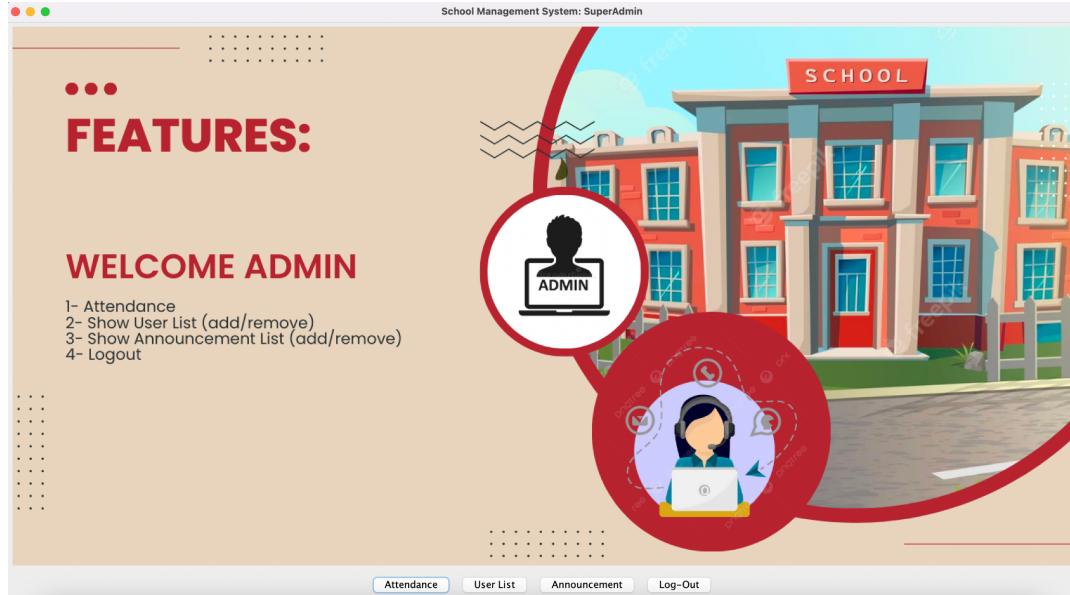


Image 1.5 SuperAdmin Homepage

The first feature is attendance. Here, SuperAdmin can check-in by themselves. SuperAdmin can also see other's (Teacher and Student) attendance. (Image 1.6)

Attendance Menu : SuperAdmin's				
Name	ID	Role	Check In Time	
SuperAdmin	230506ADM01	admin	2023.05.16.14.53.48	
Tirza Gabriella	230516TCH01	teacher	2023.05.16.14.53.44	
Tirza Gabriella	230516TCH01	teacher	2023.05.16.14.54.55	
Tirza Gabriella	230516TCH01	teacher	2023.05.16.15.11.56	
Tirza Gabriella	230516TCH01	teacher	2023.05.16.15.23.23	
Tirza Gabriella	230516TCH01	teacher	2023.05.16.16.20.32	
Meng	230516STD01	student	2023.05.16.16.30.07	
Meng	230516STD01	student	2023.05.16.16.35.14	
bong	230516STD02	student	2023.05.16.16.35.27	
bong	230516STD02	student	2023.05.16.16.36.57	
Meng	230516STD01	student	2023.05.16.16.45.03	
bong	230516STD00	student	2023.05.16.17.01.52	
bong	230516STD00	student	2023.05.16.17.03.38	
Tirza Gabriella	230516TCH01	teacher	2023.05.16.19.54.25	
SuperAdmin	230506ADM01	admin	2023.05.25.18.25.58	
Tirza Gabriella	230516TCH01	teacher	2023.05.25.18.26.42	
SuperAdmin	230506ADM01	admin	2023.05.25.19.05.41	
SuperAdmin	230506ADM01	admin	2023.05.25.19.09.34	
SuperAdmin	230516TCH01	teacher	2023.05.25.19.11.39	
Tirza Gabriella	230516TCH01	teacher	2023.05.25.19.11.58	
Meng	230516STD01	student	2023.05.25.19.12.27	
SuperAdmin	230506ADM01	admin	2023.05.25.19.18.54	
SuperAdmin	230506ADM01	admin	2023.05.25.19.18.57	
Tirza Gabriella	230516TCH01	teacher	2023.05.25.19.20.39	
Tirza Gabriella	230516TCH01	teacher	2023.05.25.19.21.43	
Meng	230516STD01	student	2023.05.25.19.21.50	
SuperAdmin	230506ADM01	admin	2023.05.25.19.26.31	
Tirza Gabriella	230516TCH01	teacher	2023.05.25.19.28.17	
Meng	230516STD01	student	2023.05.25.19.28.36	
bong	230516STD02	student	2023.05.25.19.30.31	
SuperAdmin	230506ADM01	admin	2023.05.25.19.31.41	
Tirza Gabriella	230516TCH01	teacher	2023.05.25.19.39.16	
Meng	230516STD01	student	2023.05.25.19.39.34	
bong	230516STD02	student	2023.05.25.19.41.17	
SuperAdmin	230506ADM01	admin	2023.05.25.21.05.40	
SuperAdmin	230506ADM01	admin	2023.05.26.18.06.46	
Meng	230516STD01	student	2023.05.26.18.57.01	
bong	230516STD02	student	2023.05.26.18.58.02	
SuperAdmin	230506ADM01	admin	2023.05.26.20.58.35	
SuperAdmin	230506ADM01	admin	2023.05.28.14.31.57	
Tirza Gabriella	230516TCH01	teacher	2023.05.28.14.34.54	
bong	230516STD02	student	2023.05.28.14.35.30	
SuperAdmin	230506ADM01	admin	2023.05.29.14.55.10	
SuperAdmin	230506ADM01	admin	2023.05.30.18.51.55	

Image 1.6 SuperAdmin Attendance

The second feature is Show User List. SuperAdmin's main task is to sign up Students and Teachers. SuperAdmin can also add users (user name, userID, user role, and user password) or remove users by user Id. (Image 1.7, Image 1.8, Image 1.9)

Name	ID	Role
Meng	230516STD01	student
bong	230516STD02	student
Tirza Gabriella	230516TCH01	teacher

Image 1.7 SuperAdmin User List

This screenshot shows the 'User List' section of the SuperAdmin interface. At the top, there are three colored dots (red, yellow, green). Below them, the title '[SuperAdmin] User List' is displayed. The table has three columns: 'Name', 'ID', and 'Role'. The data entries are:

Name	ID	Role
Meng	230516STD01	student
bong	230516STD02	student
Tirza Gabriella	230516TCH01	teacher

Image 1.8 SuperAdmin Add User

This screenshot shows the 'Add User' section of the SuperAdmin interface. At the top, there are three colored dots (red, yellow, green). Below them, the title 'Add User' is displayed. The form contains four input fields:

- User_Name:** A text input field.
- User_ID:** A text input field with a placeholder: 'User_ID ([yymmdd][ADM/TCH/STD][2-digit-num]). Example: 230507TCH01'.
- User_Role:** A dropdown menu with 'student' selected.
- User_Password:** A text input field.

At the bottom right of the form area, there are 'Add' and 'Cancel' buttons.

Image 1.9 SuperAdmin Remove User

The third feature is the Show Announcement List. SuperAdmin can see, add, or remove announcements. They can also remove announcements that teachers have already made. (Image 1.10)

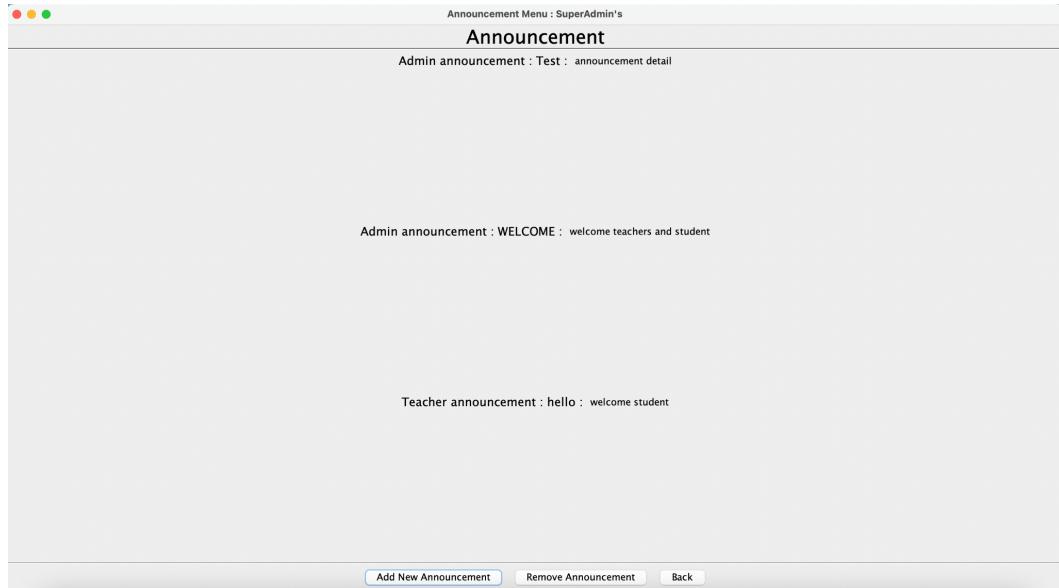


Image 1.10 SuperAdmin Announcement List

If we log in as a Teacher, there is also a list of menu features that we can see in the UI. Below the program there's a button where the Teacher can choose which features are going to be used by them. (Image 1.11)



Image 1.11 Teacher Homepage

Teacher's Attendance feature is similar to SuperAdmin attendance. The difference is that Teachers can't see SuperAdmin's attendance. They can only see students' and other teachers' attendance. In addition, Teachers can do students' check-in by student ID. (Image 1.12, Image 1.13)

Name	ID	Role	Check In Time
Tirza Gabriella	230516TC001	teacher	2023.05.16.14.53.44
Tirza Gabriella	230516TC001	teacher	2023.05.16.14.54.55
Tirza Gabriella	230516TC001	teacher	2023.05.16.15.11.56
Tirza Gabriella	230516TC001	teacher	2023.05.16.15.23.23
Meng	230516STD01	student	2023.05.16.16.20.32
Meng	230516STD01	student	2023.05.16.16.34.07
bong	230516STD02	student	2023.05.16.16.35.14
bong	230516STD02	student	2023.05.16.16.35.27
Meng	230516STD01	student	2023.05.16.16.45.03
bong	230516STD02	student	2023.05.16.17.01.52
bong	230516STD02	student	2023.05.16.17.32.38
Tirza Gabriella	230516TC001	teacher	2023.05.25.18.25.25
Tirza Gabriella	230516TC001	teacher	2023.05.25.18.26.42
Tirza Gabriella	230516TC001	teacher	2023.05.25.19.11.58
Meng	230516STD01	student	2023.05.25.19.12.27
Tirza Gabriella	230516TC001	teacher	2023.05.25.19.20.39
Tirza Gabriella	230516TC001	teacher	2023.05.25.19.20.43
Meng	230516STD01	student	2023.05.25.19.20.50
Tirza Gabriella	230516TC001	teacher	2023.05.25.19.28.17
Meng	230516STD01	student	2023.05.25.19.28.36
bong	230516STD02	student	2023.05.25.19.30.31
Tirza Gabriella	230516TC001	teacher	2023.05.25.19.39.16
Meng	230516STD01	student	2023.05.25.19.39.34
bong	230516STD02	student	2023.05.26.18.00.17
Meng	230516STD01	student	2023.05.26.18.57.01
bong	230516STD02	student	2023.05.26.19.28.02
Tirza Gabriella	230516TC002	teacher	2023.05.28.14.34.54
bong	230516STD02	student	2023.05.28.14.35.30

Image 1.12 Teacher Attendance

[Tirza Gabriella] Student List		
Name	ID	User_ID
Meng	230516STD01	230516STD02
bong	230516STD02	

Image 1.13 Teacher's Student Attendance Menu

The Show User List feature of the Teacher is the same as SuperAdmin. Both can see the list, add, and remove. The difference being teachers can only add or remove a student. Teachers can't add or remove admin or other teachers.

The Announcement feature from the Teacher class is similar to SuperAdmin, but there is a major difference with the Announcement Removal feature, in which the Teacher can't remove announcements other than his/hers.

There's one additional feature for teachers, which is the Assignment method. Teachers can post assignments, delete assignments, score the assignments, change the status (pending / scored) , and give feedback. (Image 1.14, Image 1.15)

Assignments				
Id	Title	Desc	Teacher	Deadline
1	title example	this is an example desc for the assignment	Tirza Gabriella	2023-05-20 00:00:00
2	title 2	desc 2	Tirza Gabriella	2023-05-01 00:00:00
3	assignment 1	test	Tirza Gabriella	2024-01-20 00:00:00
4	test	assignment1	Tirza Gabriella	2023-09-20 00:00:00
5	test	test1	Tirza Gabriella	2023-05-25 00:00:00
6	assignment	test1	Tirza Gabriella	2023-05-25 00:00:00
7	test5	nyow	Tirza Gabriella	2023-05-28 00:00:00
8	n,nnn	mmmmn	Tirza Gabriella	2023-03-20 00:00:00
9			Tirza Gabriella	2023-03-20 00:00:00
10	test20	ansja	Tirza Gabriella	2020-09-10 00:00:00
11	Math	5 + 10 =	Tirza Gabriella	2023-06-20 00:00:00

Image 1.14 Teacher Assignment Menu

Add new assignment

Title	<input type="text"/>
Description	<input type="text"/>
Deadline (YYYY-MM-DD)	<input type="text"/>

Add **Cancel**

Image 1.15 Teacher Add Assignment Menu

If we log in as a Student, there is also a list of menu features that we can see in the UI. Below the screen there's a button where the Student can choose which feature to use. (Image 1.16)



Image 1.16 Student Homepage

In the student attendance feature, students can only see their own attendance and check-in themselves. But they can't see others' attendance.

In the announcement feature, students can see announcements made by the admin or the teacher, but they can't add or remove announcements.

In the assignment feature, students can see the assignment and submit their assignment works. (Image 1.17)

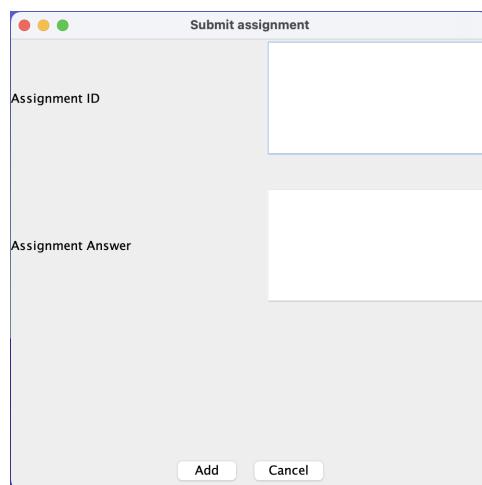


Image 1.17 Student Submit Assignment Menu

IX. Lesson Learned

a. Error Handling

Dealing with errors can be highly frustrating, especially when it involves debugging our own program. However, I have developed a personal approach to identify the issues in my code. Firstly, I make use of the print function to note down specific areas where I anticipate certain outcomes. If the expected results are successfully printed in the console, it confirms my assumptions. Conversely, if nothing is printed or an unexpected result occurs, it indicates an error that requires further investigation. Secondly, the part that I'm not used to is to use the DataBase because it's such a new thing for me and it's never taught in class. This means I need more time to research how to use MySQL and the helper tools related to it (DBeaver).

b. Reflection

From this project I acquired a lot of new knowledge especially about Object-Oriented-Programming (class, inheritance, interface, etc) and some modules (JFrame, ActionListener, JTable, etc). Programming can evoke a mix of challenges, enjoyment, and intrigue, yet the pressure of meeting deadlines adds a certain level of tension. Despite this, I find contentment in the outcome of my program, recognizing that there are numerous areas where further enhancements can still be implemented (for example, to add error-handling if user inputs passed dates as deadline dates, tidying up the UI, re-think some better flows, etc) to the existing program.