A · P · U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Module Code:	CT038-2-OODJ (Object Oriented Programming With Java)	
Title:	Project Management System	
Intake Code:	APD2F2402CS(AI)/ APU2F2402CS(AI)	
Team Members:	Chan Kok Ming	TP074467
	Ching Jia Zhong	TP074569
	Liew Jing Xuan	TP074293
	Chan Bo Wen	TP068075
Hand out date:	26 March 2024	
Submission date:	7 June 2024	

Table of Contents

1.0 Design Solution

1.1 Use Case Diagram

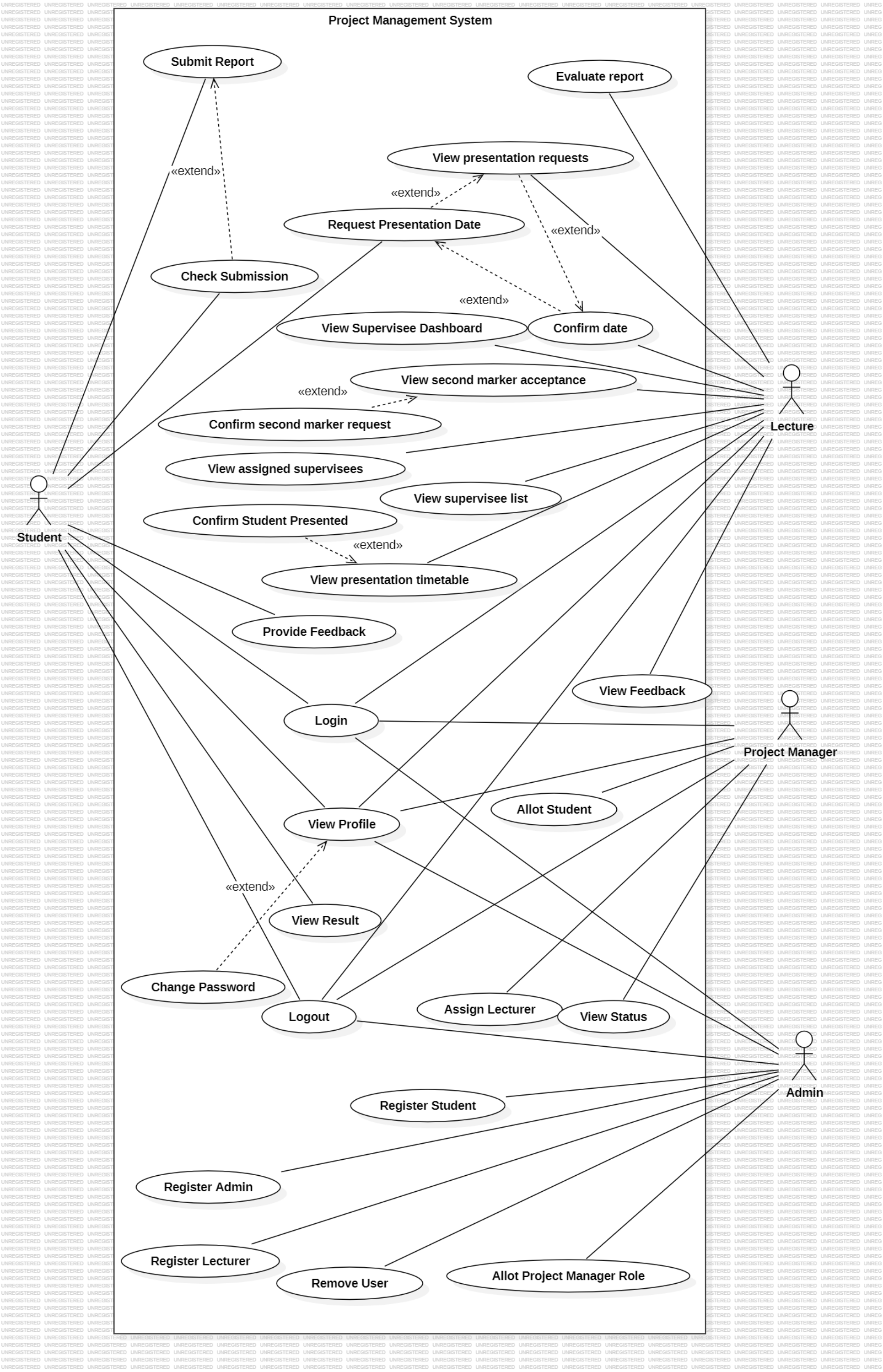


Figure 1.1.1: Use case diagram

1.2 Use Case Descriptions

Use Case	Login
Description	Allows user to log into the system.
Actor	Admin, Lecturer, Project Manager, Student
Preconditions	User must have a registered ID and password
Postconditions	User is logged into the system

Table 1.2.1: Use-case description of login

Use Case	Register Student
Description	Allows the admin to add a new student to the system
Actor	Admin
Preconditions	• Admin must be logged in. • Student ID must follow the format "TPXXXXXX"
Postconditions	New student data is added to Studentdata.txt

Table 1.2.2: Use-case description of register student

Use Case	Register Lecturer
Description	Allows the admin to add a new lecturer to the system
Actor	Admin
Preconditions	• Admin must be logged in. • Lecturer ID must follow the format "LRXXXXXX"
Postconditions	New student data is added to Studentdata.txt

Table 1.2.3: Use-case description of register lecturer

Use Case	Register Admin
Description	Allows the admin to add a new admin to the system
Actor	Admin
Preconditions	• Admin must be logged in • Lecturer ID must follow the format "LRXXXXXX"
Postconditions	New admin data is added to Admindata.txt

Table 1.2.4: Use-case description of register admin

Use Case	Remove User
Description	Allows the admin to remove a user from the system
Actor	Admin
Preconditions	• Admin must be logged in • Admin must select a user to remove
Postconditions	User data is removed from the respective data file.

Table 1.2.5: Use-case description of remove user

Use Case	Allot Project Manager Role
Description	Allows the admin to assign a lecturer as a project manager
Actor	Admin
Preconditions	• Admin must be logged in • Project Manager ID must follow the format "PMXXXXXX"

Postconditions	Lecturer data is moved from Lecturerdata.txt to PMdata.txt with the new ID.

Table 1.2.6: Use-case description of allot project manager role

Use Case	View Profile
Description	Displays the logged-in user's data on the profile screen.
Actor	Admin, Lecturer, Project Manager, Student
Preconditions	User must be logged in
Postconditions	User's data is displayed on the profile screen.
Extends	Change Password

Table 1.2.7: Use-case description of view profile

Use Case	Change Password
Description	Allows user to change their password
Actor	Admin, Lecturer, Project Manager, Student
Preconditions	User must be logged in
Postconditions	User's password is updated in the respective data file.

Table 1.2.8: Use-case description of change password

Use Case	Evaluate Report
Description	Evaluate report submitted by student
Actor	Lecturer
Preconditions	Students need to submit report to lecturer

Postconditions	Lecturers give the grade and feedback to students

Table 1.2.9: Use-case description of evaluate report

Use Case	Confirm Student Presented
Description	Update student presentation status to “Presented”
Actor	Lecturer
Preconditions	Student have delivered presentation to the lecturer
Postconditions	The delivered presentation will then be removed from the presentation timetable of the lecturer.

Table 1.2.10: Use-case description of Confirm Student Presented

Use Case	View Presentation requests
Description	View presentation request by student
Actor	Lecturer
Preconditions	Students need to request the presentation time to lecturer
Postconditions	Lecturers view the presentation request by student

Table 1.2.11: Use-case description of view presentation request

Use Case	Confirm date
Description	Confirm the date of presentation request
Actor	Lecturer
Preconditions	Students need to request the presentation time to lecturer
Postconditions	Lecturers confirm the date of presentation request

Table 1.2.12: Use-case description of confirm date

Use Case	View Supervisee Dashboard
Description	View the supervisee report status and presentation status
Actor	Lecturer
Preconditions	Project Manager allocate project to student and allocate lecturer as supervisor
Postconditions	To view the supervisee report status and presentation status

Table 1.2.13: Use-case description of view supervisee dashboard

Use Case	View Second Marker Acceptance
Description	Ensure lecturer is a second marker or not
Actor	Lecturer
Preconditions	Project Manager assign the role to lecturer as second marker
Postconditions	To view the lecturer is a second marker or not

Table 1.2.14: Use-case description of view second marker

Use Case	View Assigned Supervisees
Description	View the supervisee that assign by Project Manager
Actor	Lecturer
Preconditions	Project Manager need to assign supervisee to lecturer
Postconditions	View the supervisee that responsible by lecturer

Table 1.2.15: Use-case description of view assigned supervisees

Use Case	View Supervisee List
Description	View the presentation request list
Actor	Lecturer
Preconditions	Lecturers need to accept or rejected to the presentation request
Postconditions	View the supervisee list after the confirmation of the presentation request

Table 1.2.16: Use-case description of view supervisee list

Use Case	View presentation timetable
Description	View the presentation timetable that had been accepted
Actor	Lecturer
Preconditions	Lecturers need to accept the presentation request
Postconditions	View the accepted presentation time

Table 1.2.17: Use-case description of view presentation timetable

Use Case	View Feedback
Description	View the feedback
Actor	Lecturer
Preconditions	Student had sent the feedback to lecturer
Postconditions	View the feedback from student

Table 1.2.18: Use-case description of view feedback

Use Case	Submit Report
Description	Submit report write by student
Actor	Student
Preconditions	Project Manager allocate project to student
Postconditions	Students submit their report

Table 1.2.19: Use-case description of submit report

Use Case	Check Submission
Description	Check the submission by student
Actor	Student
Preconditions	Project Manager allocate project to student
Postconditions	Students check the report submission status

Table 1.2.20: Use-case description of check submission

Use Case	Request Presentation Date
Description	Request presentation date with lecturer
Actor	Student
Preconditions	Project Manager allocate project to student
Postconditions	Students request the presentation with lecturer

Table 1.2.21: Use-case description of request presentation date

Use Case	Provide Feedback
Description	Provide feedback to the lecturer for improvement
Actor	Student
Preconditions	Students want to provide feedback to lecturer
Postconditions	Students provide feedback to lecturer anonymously

Table 1.2.22: Use-case description of provide feedback

Use Case	Confirm second marker request
Description	Accept or reject the request to be second marker assigned by project manager
Actor	Lecturer
Preconditions	Project managers assign second marker request to lecturer
Postconditions	Lecturers accept to be second marker or reject to not be second marker

Table 1.2.23: Use-case description of confirm second marker request

Use Case	Allocate Student
Description	Allocate project to student
Actor	Project Manager
Preconditions	Admin enrols new student
Postconditions	Students accept the project and make submission

Table 1.2.24: Use-case description of allocate student

Use Case	Assign Lecturer
Description	Assign supervisor and second marker role to lecturer
Actor	Project Manager
Preconditions	Lecturer have not assigned to supervisor or second marker
Postconditions	If assign role is supervisor, lecturer need to mark the project done by student. If assign role is second marker, need to accept by lecturer.

Table 1.2.25: Use-case description of assign lecturer

Use Case	View Status
Description	View the scores of reports submitted by students and the report
Actor	Project Manager
Preconditions	Project have already allocated the project to student
Postconditions	View the result from student

Table 1.2.26: Use-case description of view status

Use Case	Logout
Description	Allow user logout the system
Actor	Admin, Project Manager, Lecturer, Student
Preconditions	Users have already login the system
Postconditions	Users click “confirm” to logout the system

Table 1.2.27: Use-case description of logout

1.3 Class Diagram

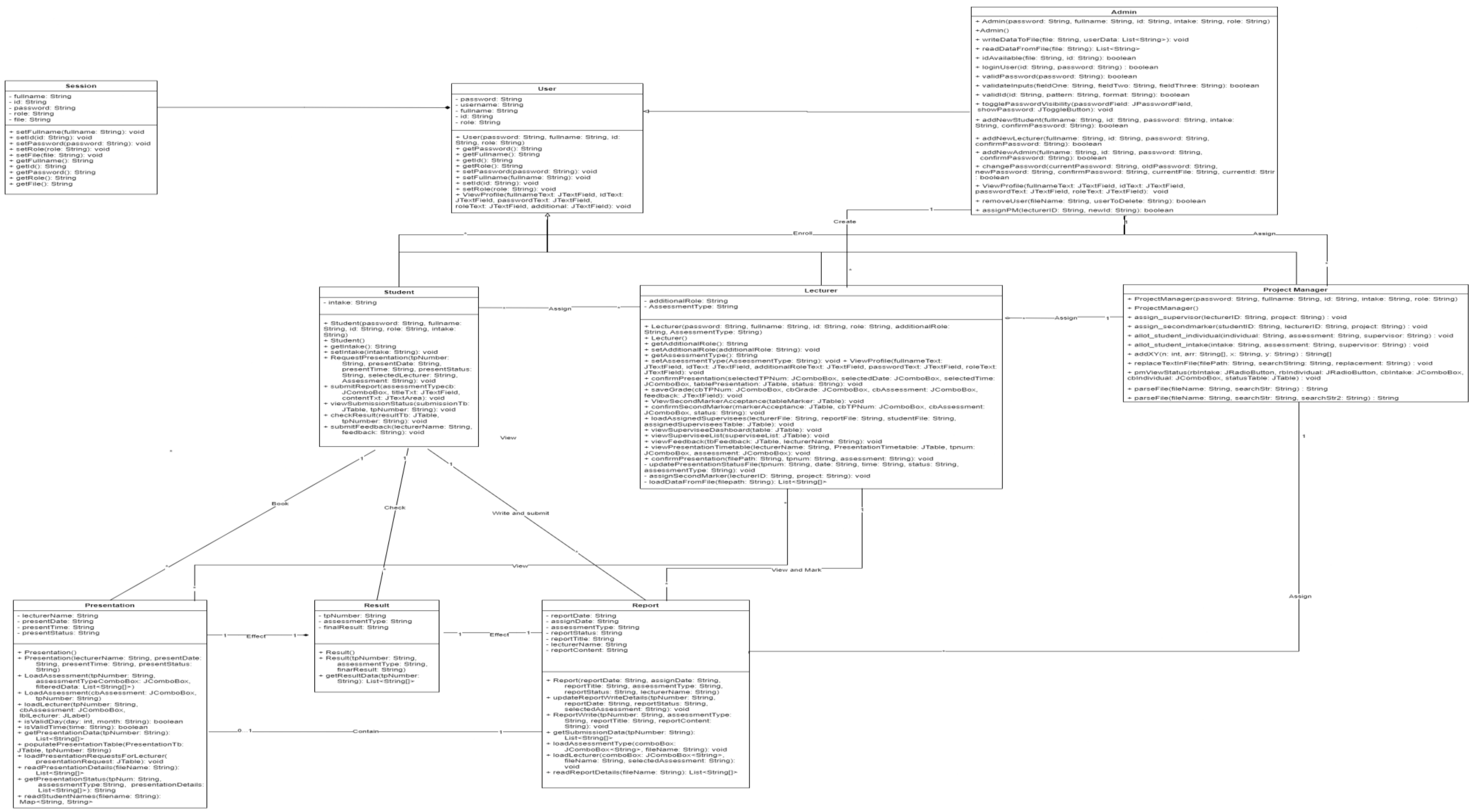

Session
- fullname: String
- id: String
- password: String
- role: String
- file: String
+ setFullname(fullname: String): void
+ setId(id: String): void
+ setPassword(password: String): void
+ setRole(role: String): void
+ setFile(file: String): void
+ getFullname(): String
+ getId(): String
+ getPassword(): String
+ getRole(): String
+ getFile(): String
User
- password: String
- username: String
- fullname: String
- id: String
- role: String
+ User(password: String, fullname: String, id: String, role: String)
+ getPassword(): String
+ getFullname(): String
+ getId(): String
+ getRole(): String
+ setPassword(password: String): void
+ setFullname(fullname: String): void
+ setId(id: String): void
+ setRole(role: String): void
+ ViewProfile(fullnameText: JTextField, idText: JTextField, passwordText: JTextField, roleText: JTextField, additional: JTextField): void
Admin
+ Admin(password: String, fullname: String, id: String, intake: String, role: String)
+Admin()
+ writeDataToFile(file: String, userData: List<String>): void
+ readDataFromFile(file: String): List<String>
+ idAvailable(file: String, id: String): boolean
+ loginUser(id: String, password: String) : boolean
+ validPassword(password: String): boolean
+ validateInputs(fieldOne: String, fieldTwo: String, fieldThree: String): boolean
+ validId(id: String, pattern: String, format: String): boolean
+ togglePasswordVisibility(passwordField: JPasswordField, showPassword: JToggleButton): void
+ addNewStudent(fullname: String, id: String, password: String, intake: String, confirmPassword: String): boolean
+ addNewLecturer(fullname: String, id: String, password: String, confirmPassword: String): boolean
+ addNewAdmin(fullname: String, id: String, password: String, confirmPassword: String): boolean
+ changePassword(currentPassword: String, oldPassword: String, newPassword: String, confirmPassword: String, currentFile: String, currentId: Strir : boolean
+ ViewProfile(fullnameText: JTextField, idText: JTextField, passwordText: JTextField, roleText: JTextField): void
+ removeUser(fileName: String, userToDelete: String): boolean
+ assignPM(lecturerID: String, newId: String): boolean
Create
Enroll
Assign
Student
- intake: String
+ Student(password: String, fullname: String, id: String, role: String, intake: String)
+ Student()
+ getIntake(): String
+ setIntake(intake: String): void
+ RequestPresentation(tpNumber: String, presentDate: String, presentTime: String, presentStatus: String, selectedLecturer: String, Assessment: String): void
+ submitReport(assessmentTypecb: JComboBox, titleTxt: JTextField, contentTxt: JTextArea): void
+ viewSubmissionStatus(submissionTb: JTable, tpNumber: String): void
+ checkResult(resultTb: JTable, tpNumber: String): void
+ submitFeedback(lecturerName: String, feedback: String): void
Assign
Lecturer
- additionalRole: String
- AssessmentType: String
+ Lecturer(password: String, fullname: String, id: String, role: String, additionalRole: String, AssessmentType: String)
+ Lecturer()
+ getAdditionalRole(): String
+ setAdditionalRole(additionalRole: String): void
+ getAssessmentType(): String
+ setAssessmentType(AssessmentType: String): void + ViewProfile(fullnameText: JTextField, idText: JTextField, additionalRoleText: JTextField, passwordText: JTextField, roleText: JTextField): void
+ confirmPresentation(selectedTPNum: JComboBox, selectedDate: JComboBox, selectedTime: JComboBox, tablePresentation: JTable, status: String): void
+ saveGrade(cbTPNum: JComboBox, cbGrade: JComboBox, cbAssessment: JComboBox, feedback: JTextField): void
+ ViewSecondMarkerAcceptance(tableMarker: JTable): void
+ confirmSecondMarker(markerAcceptance: JTable, cbTPNum: JComboBox, cbAssessment: JComboBox, status: String): void
+ loadAssignedSupervisees(lecturerFile: String, reportFile: String, studentFile: String, assignedSuperviseesTable: JTable): void
+ viewSuperviseeDashboard(table: JTable): void
+ viewSuperviseeList(superviseeList: JTable): void
+ viewFeedback(tbFeedback: JTable, lecturerName: String): void
+ viewPresentationTimetable(lecturerName: String, PresentationTimetable: JTable, tpnum: JComboBox, assessment: JComboBox): void
+ confirmPresentation(filePath: String, tpnum: String, assessment: String): void
- updatePresentationStatusFile(tpnum: String, date: String, time: String, status: String, assessmentType: String): void
- assignSecondMarker(lecturerID: String, project: String): void
- loadDataFromFile(filepath: String): List<String[]>
Assign
Project Manager
+ ProjectManager(password: String, fullname: String, id: String, intake: String, role: String)
+ ProjectManager()
+ assign_supervisor(lecturerID: String, project: String) : void
+ assign_secondmarker(studentID: String, lecturerID: String, project: String) : void
+ allot_student_individual(individual: String, assessment: String, supervisor: String) : void
+ allot_student_intake(intake: String, assessment: String, supervisor: String) : void
+ addXY(n: int, arr: String[], x: String, y: String) : String[]
+ replaceTextInFile(filePath: String, searchString: String, replacement: String) : void
+ pmViewStatus(rbIntake: JRadioButton, rbIndividual: JRadioButton, cbIntake: JComboBox, cbIndividual: JComboBox, statusTable: JTable) : void
+ parseFile(fileName: String, searchStr: String) : String
+ parseFile(fileName: String, searchStr: String, searchStr2: String) : String
View
Book
Check
Write and submit
View
View and Mark
Assign
Presentation
- lecturerName: String
- presentDate: String
- presentTime: String
- presentStatus: String
+ Presentation()
+ Presentation(lecturerName: String, presentDate: String, presentTime: String, presentStatus: String)
+ LoadAssessment(tpNumber: String, assessmentTypeComboBox: JComboBox, filteredData: List<String[]>)
+ LoadAssessment(cbAssessment: JComboBox, tpNumber: String)
+ loadLecturer(tpNumber: String, cbAssessment: JComboBox, lblLecturer: JLabel)
+ isValidDay(day: int, month: String): boolean
+ isValidTime(time: String): boolean
+ getPresentationData(tpNumber: String): List<String[]>
+ populatePresentationTable(PresentationTb: JTable, tpNumber: String)
+ loadPresentationRequestsForLecturer(presentationRequest: JTable): void
+ readPresentationDetails(fileName: String): List<String[]>
+ getPresentationStatus(tpNum: String, assessmentType:String, presentationDetails: List<String[]>): String
+ readStudentNames(filename: String): Map<String, String>
Effect
Result
- tpNumber: String
- assessmentType: String
- finalResult: String
+ Result()
+ Result(tpNumber: String, assessmentType: String, finarResult: String)
+ getResultData(tpNumber: String): List<String[]>
Effect
Contain
Report
- reportDate: String
- assignDate: String
- assessmentType: String
- reportStatus: String
- reportTitle: String
- lecturerName: String
- reportContent: String
+ Report(reportDate: String, assignDate: String, reportTitle: String, assessmentType: String, reportStatus: String, lecturerName: String)
+ updateReportWriteDetails(tpNumber: String, reportDate: String, reportStatus: String, selectedAssessment: String): void
+ ReportWrite(tpNumber: String, assessmentType: String, reportTitle: String, reportContent: String): void
+ getSubmissionData(tpNumber: String): List<String[]>
+ loadAssessmentType(comboBox: JComboBox<String>, fileName: String): void
+ loadLecturer(comboBox: JComboBox<String>, fileName: String, selectedAssessment: String): void
+ readReportDetails(fileName: String): List<String[]>

Figure 1.3.1: Class Diagram

The diagram above represents the class diagram of the entire system. The primary class is User, which serves as both a superclass and an abstract class. The subclasses include admin, lecturer, project manager, and student, all of which inherit attributes and methods from the User class. The relationship between session and User is one of composition. Additionally, admin is associated with project manager, lecturer, and student because these roles are created by admin. There is an aggregation relationship between Project Manager and Lecturer because Project Manager needs to assign supervisors and second markers to lecturers. Furthermore, Project Manager is responsible for allocating assessments to students, hence the connection with report. Lecturer is associated with students and needs to modify reports and view presentations, thus having associations with both of these classes. Lastly, students are associated with presentation, result, and report, while result is influenced by both report and presentation.

2.0 Outputs of the Program

2.1 Admin

2.1.1 Login Page

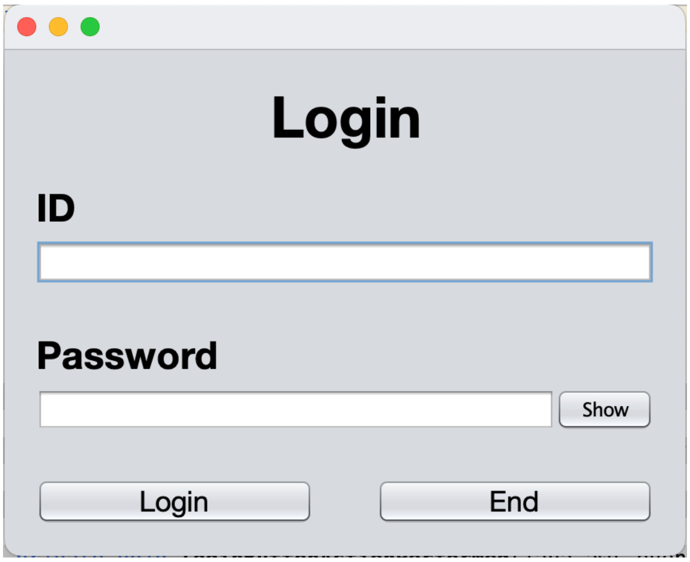
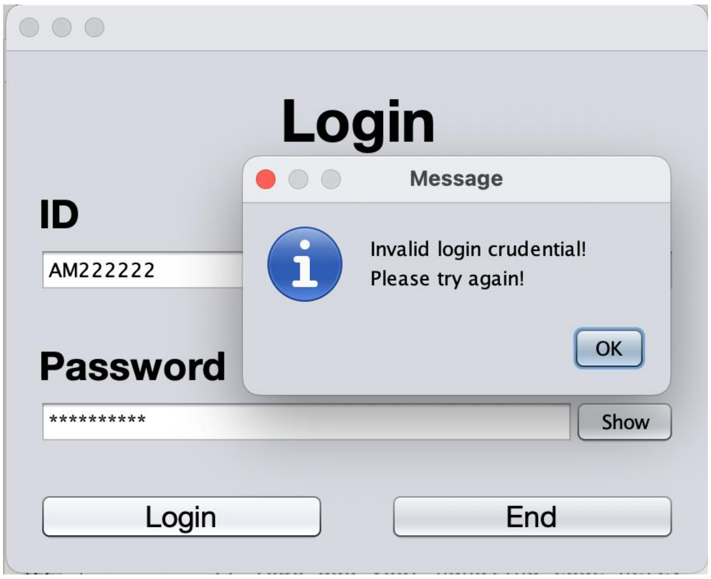


Figure 2.1.1.1: GUI of login page

Figure 2.1.1.1 depicts the login interface that will be executed immediately upon launching the program. It prompts users to input their ID along with the account password. Additionally, an "End" button is implemented to provide users with the option to terminate the program at any time. In the event of incorrect input for either the ID or password, a window as illustrated in figure 2.1.1.2, will appear, indicating invalid login attempt and prompting them to retry.



AM222222

****

Invalid login crudential!
Please try again!

Figure 2.1.1.2: Invalid credentials notification

Upon successfully login, the program will direct the user to a designated menu page based on their account role, such as admin or project manager.

2.1.2 Admin Menu Page

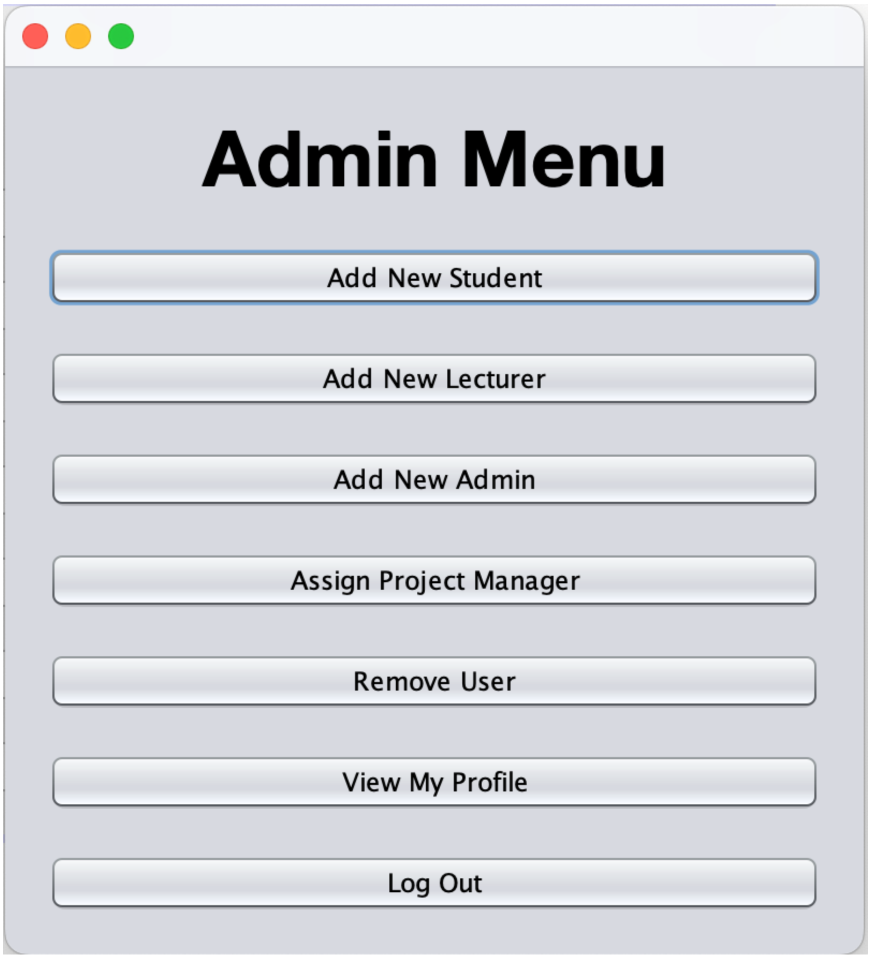
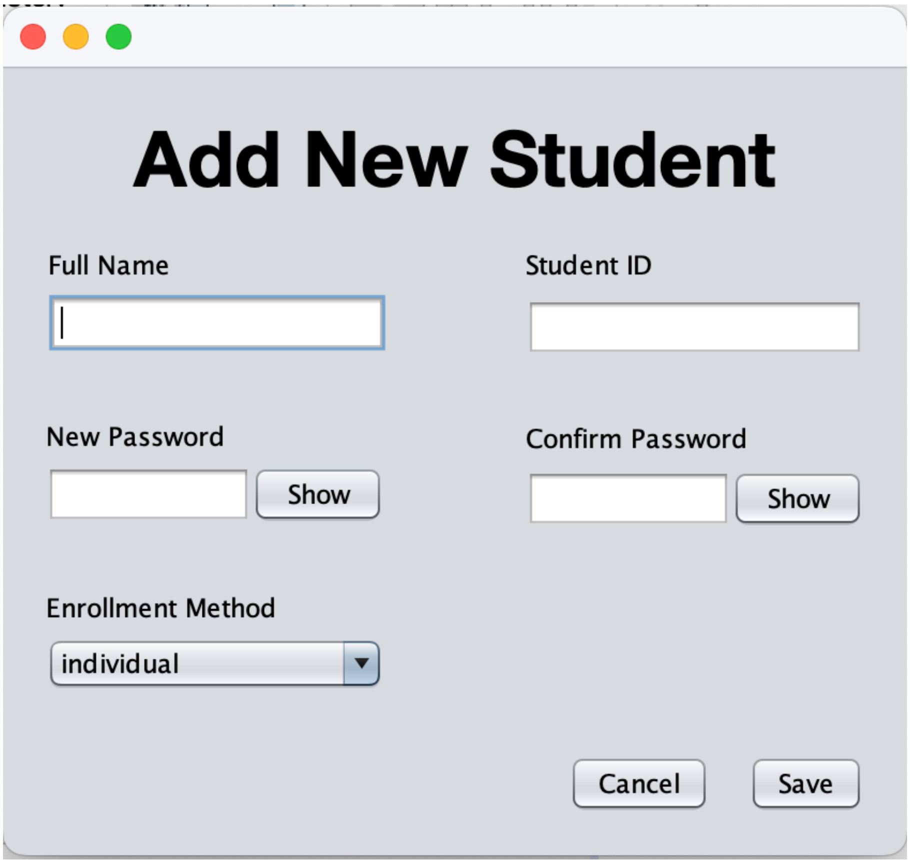


Figure 2.1.2.1: Admin menu

Figure 2.1.2.1 shows the admin menu that appears after a user with an admin role successfully logs into the system. On this page, the admin can initiate various functions or actions by clicking the available buttons, which will direct navigate to the designated windows. These functions will be further explained in this documentation. Moreover, the "Log Out" button allows the user to log out of their account and return to the login page.

2.1.3 Add New Student Page



Add New Student

Full Name:

Student ID:

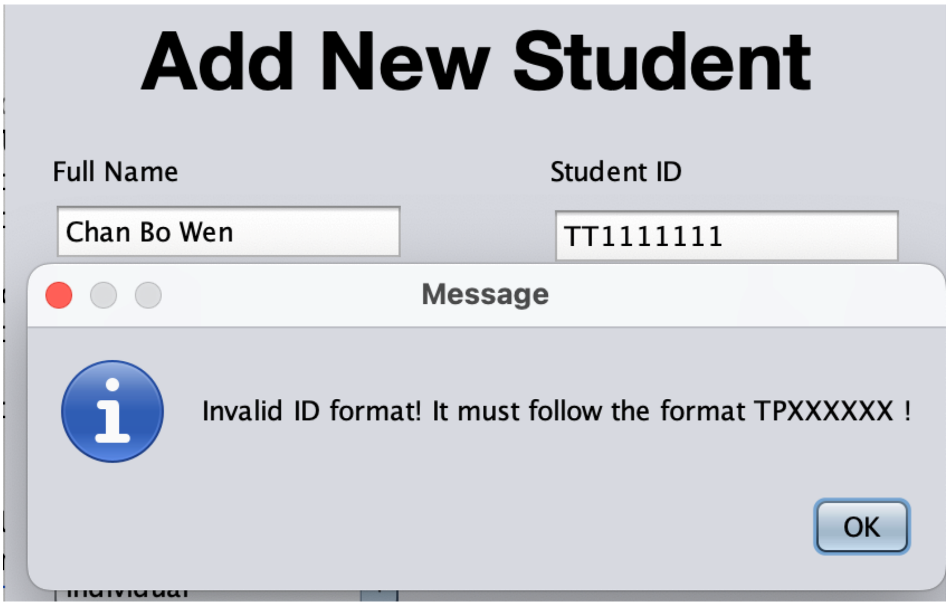
New Password:

Confirm Password:

Enrollment Method: individual

Figure 2.1.3.1: Add new student

If the "Add New Student" button is clicked by the admin, an "Add New Student" page will show out, prompting the admin to fill the details of the new student to enable them to gain access to the system. All fields are mandatory; if any field is left empty, an error notification will alert the user to complete all fields. Furthermore, validation checks are performed for the new student ID and password. For ID, it must follows the pattern "TPXXXXXX" and must be unique. Whereas, password must contain one uppercase letter, one lowercase letter, be free of spaces and symbols, and be within 6 and 12 characters long. Additionally, a "Confirm Password" field is provided to ensure the new password is the same with the confirmation password. An example of the error message that will appear for invalid entries mentioned above is shown in figure 2.1.3.2.



Add New Student

Full Name: Chan Bo Wen

Student ID: TT1111111

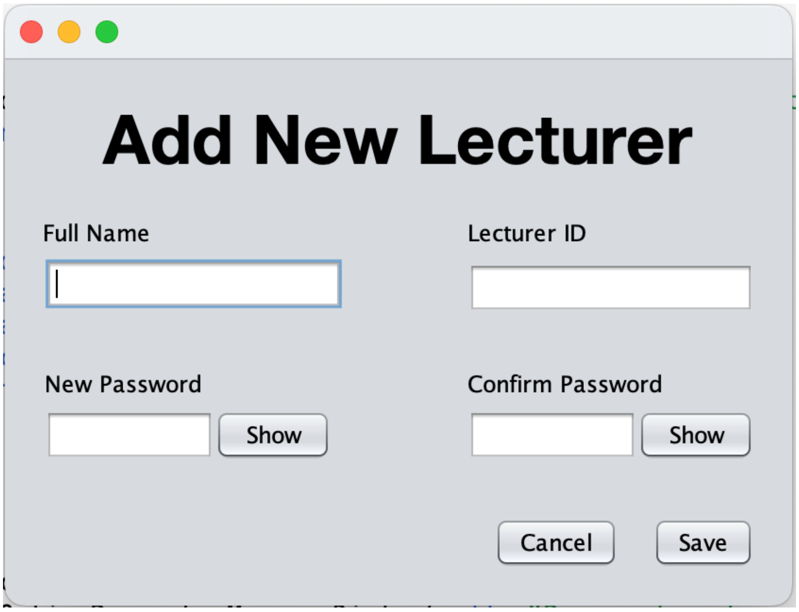
Message

Invalid ID format! It must follow the format TPXXXXXX !

Figure 2.1.3.2: Invalid ID assigned notification

After completing all required fields correctly and are valid, the admin should click the "Save" button to save the details to a text file that stores all student data. Upon successful saving, a confirmation message will appear, and the user will be returned to the admin menu. A "Cancel" button is also available, enabling the admin to cancel the action and return to the admin menu without saving any details.

2.1.4 Add New Lecturer Page



Add New Lecturer

Full Name:

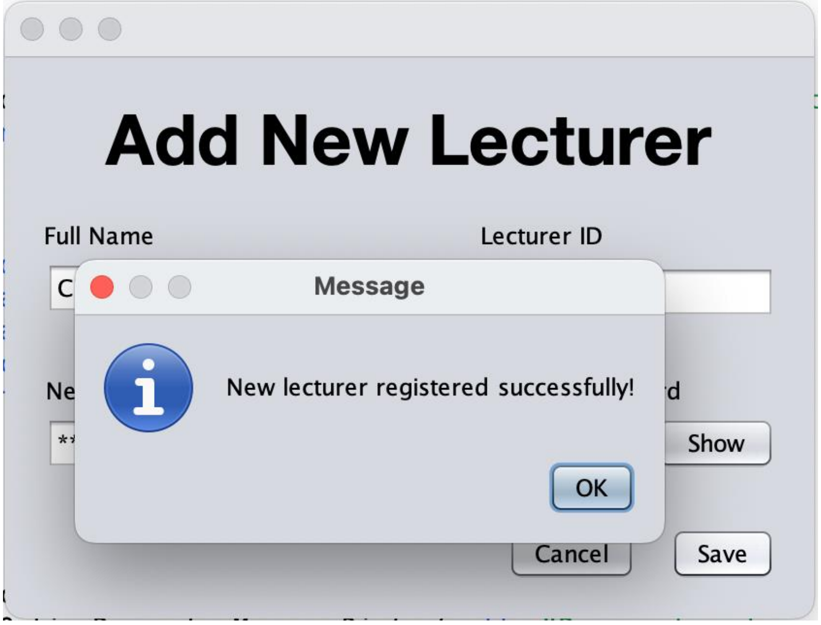
Lecturer ID:

New Password:

Confirm Password:

Figure 2.1.4.1: Add new lecturer

Admin has the opportunity to register a new lecturer account by selecting the "Add New Lecturer" function. Similar to adding a new student, all fields must be completed. The new lecturer ID must be unique and follow the "LRXXXXXX" pattern, as well as password must adhere to the same format as mentioned for adding a new student. Also, the "Save" and "Cancel" buttons will perform the same actions. Clicking "Save" will store the details in a text file containing all lecturer information, and a successful action message will appear, as illustrated in Figure 2.1.4.2, before returning the user to the admin menu. Clicking "Cancel" will abort the process and return the user to the admin menu without saving any details.



Add New Lecturer

Full Name:

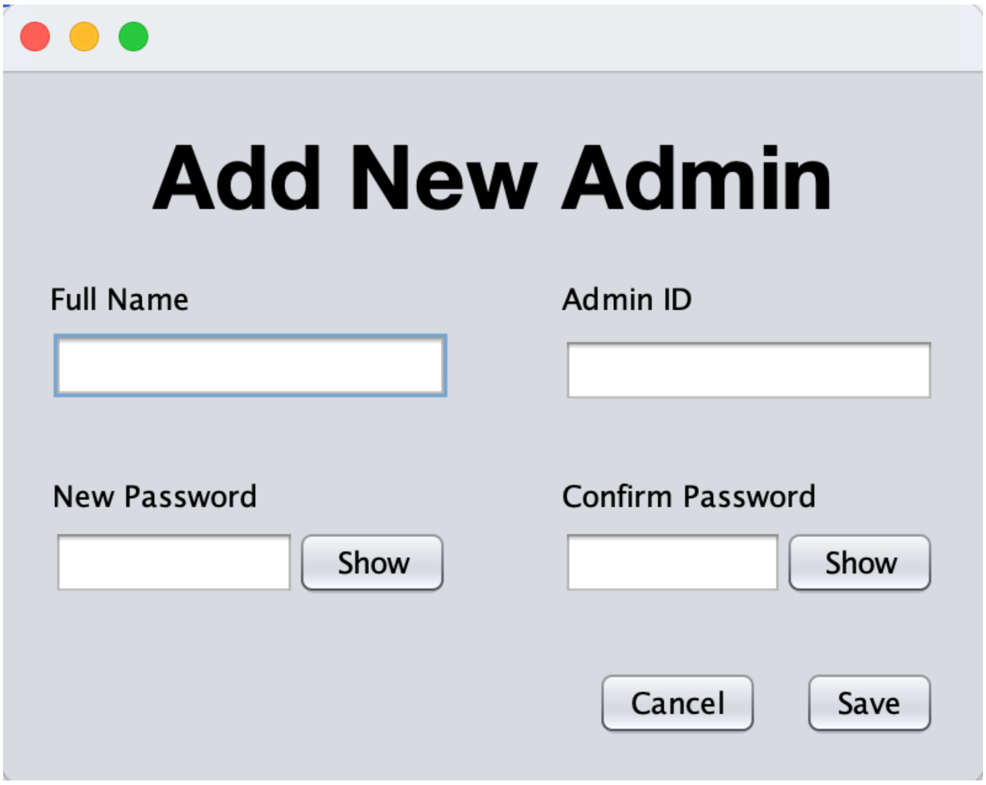
Lecturer ID:

Message

New lecturer registered successfully!

Figure 2.1.4.2: Successful registration

2.1.5 Add New Admin Page

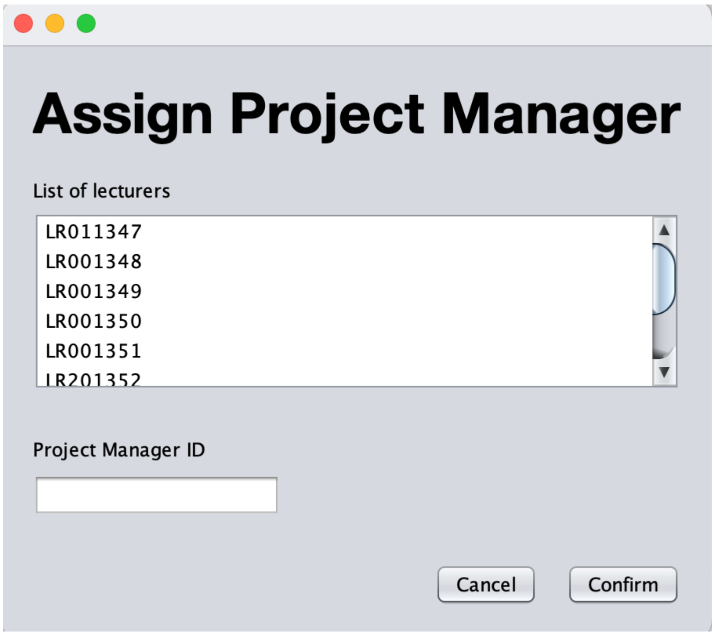


Add New Admin

Figure 2.1.5.1: Add new admin

The page shown in figure 2.1.5.1 will appear after admin clicks the “Add New Admin” button in the admin menu. Likewise, this page functions similarly to the "Add New Lecturer" feature, with the same validation requirements and buttons. The differences are that the admin ID must follow the "AMXXXXXX" pattern and must be unique. The "Save" button will save the new admin's details to a text file containing all admin information and display a successful action message before returning the user to the admin menu. The "Cancel" button will return the user to the admin menu without saving any details.

2.1.6 Assign Project Manager Page



Assign Project Manager

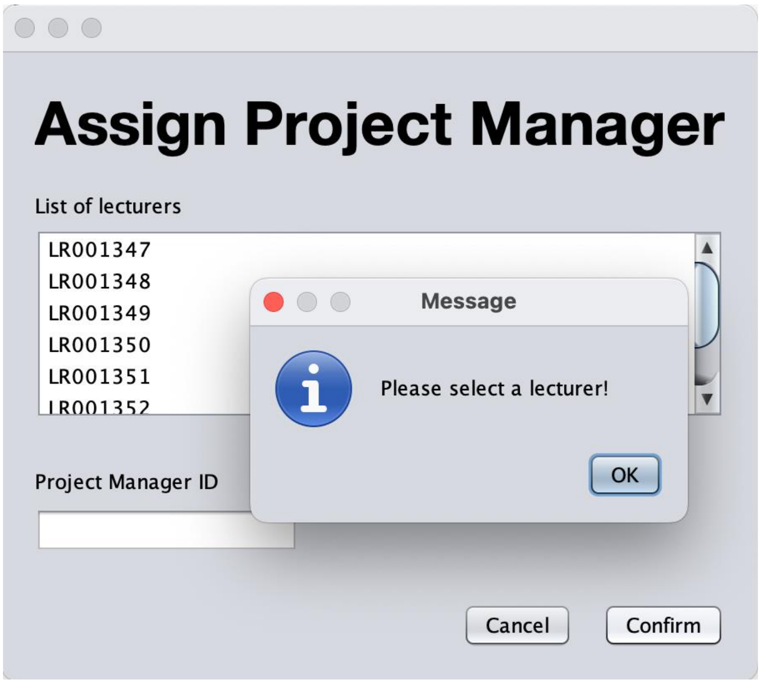
List of lecturers

LR011347
LR001348
LR001349
LR001350
LR001351
LR201352

Project Manager ID

Figure 2.1.6.1: Assign project manager

To assign an existing lecturer to the project manager role, admin can click the “Assign Project Manager” button, which directs them to the interface shown in figure 2.1.6.1. This page displays a list of existing lecturer IDs recorded in the text file, allowing the admin to select a lecturer for the project manager role. If no lecturer is selected and the "Confirm" button is clicked, an error message will appear. Additionally, a field is provided for entering the new project manager's ID, which must follow the "PMXXXXXX" pattern.



Assign Project Manager

List of lecturers

LR001347
LR001348
LR001349
LR001350
LR001351
LR001352

Message

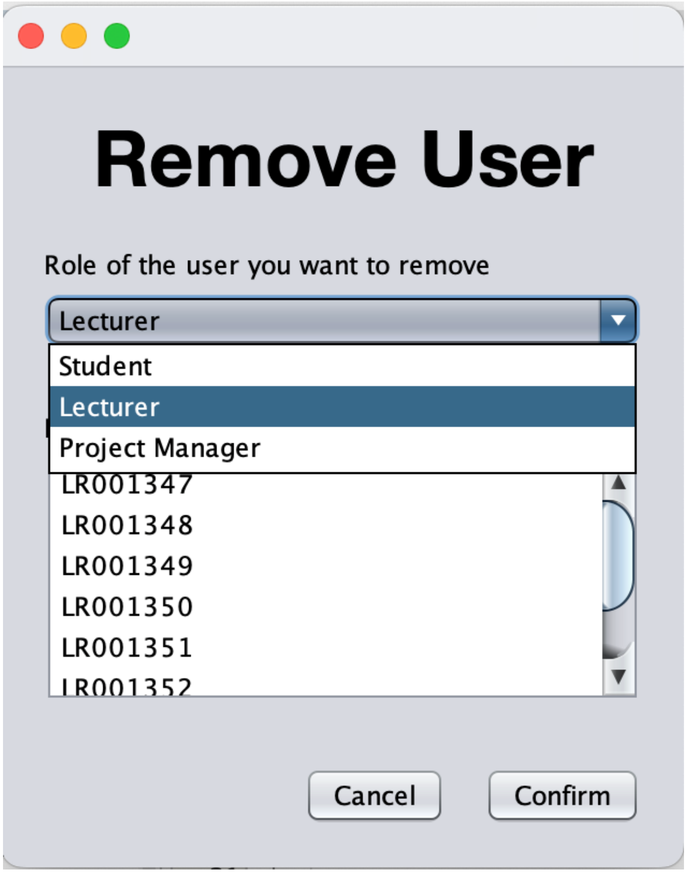
Please select a lecturer!

Project Manager ID

Figure 2.1.6.2: Error message when no user is selected

Upon successful assignment, the existing data for the lecturer, such as full name and password, will be retained. Only the user's ID and role will be updated to reflect their new position as a project manager. The details of this lecturer in the lecturer data files will be entirely removed and the newly data will be saved to project manager file.

2.1.7 Remove User Page



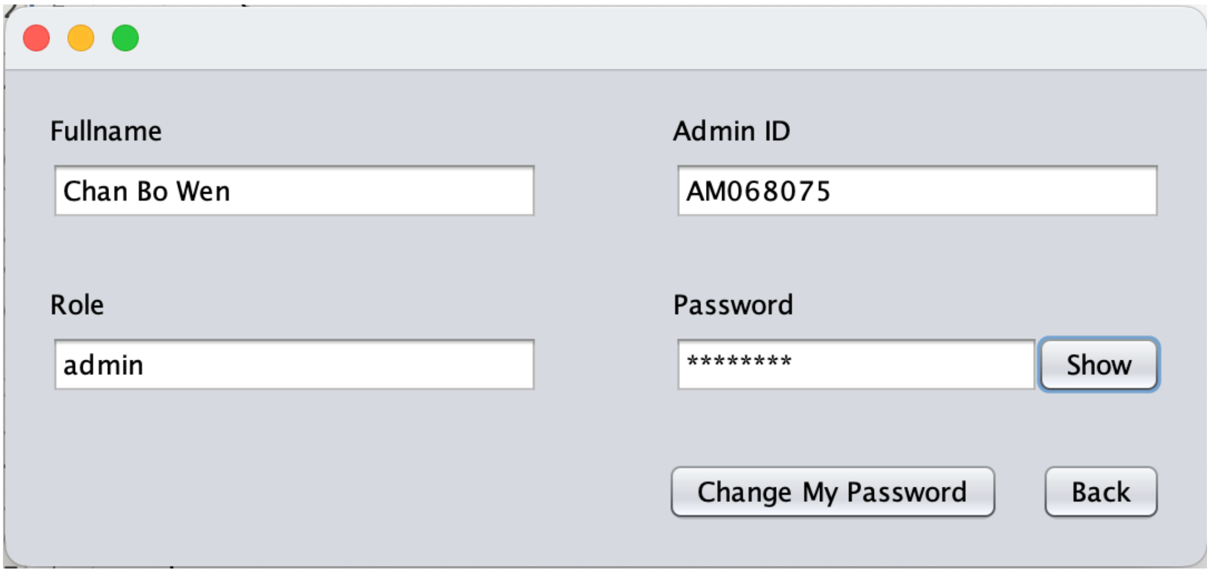
Role of the user you want to remove: Lecturer

LR001347
LR001348
LR001349
LR001350
LR001351
LR001352

Figure 2.1.7.1: Remove a user

Admins have the capability to remove users with student, lecturer, or project manager roles. When the admin selects the "Remove User" button, a dedicated page for this function, as shown in Figure 2.1.7.1, will appear. The admin must first select the role of the user to be deleted (student, lecturer, or project manager). Subsequently, a list of user IDs corresponding to the selected role will be displayed. The admin should then choose the user's ID from the list and click the "Confirm" button to remove the user. Then, these selected user’s data will be removed from the corresponding role data file.

2.1.8 View Admin Profile Page



Fullname

Chan Bo Wen

Admin ID

AM068075

Role

admin

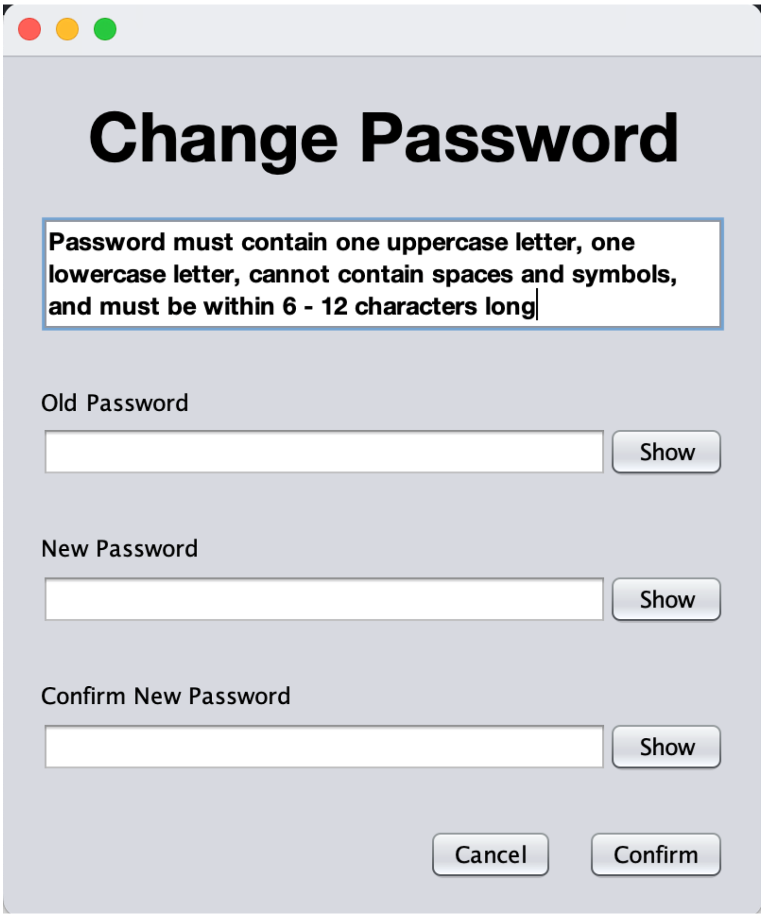
Password

****

Figure 2.1.8.1: Profile details

Current user with admin role can view their profile details, which include their full name, Admin ID, role, and account password, using the "View My Profile" feature. This page also contains a "Change My Password" button that directs the user to a page where they can change their password, as well as a "Back" button to return to the Admin menu.

2.1.9 Change Password Page



Change Password

Password must contain one uppercase letter, one lowercase letter, cannot contain spaces and symbols, and must be within 6 - 12 characters long

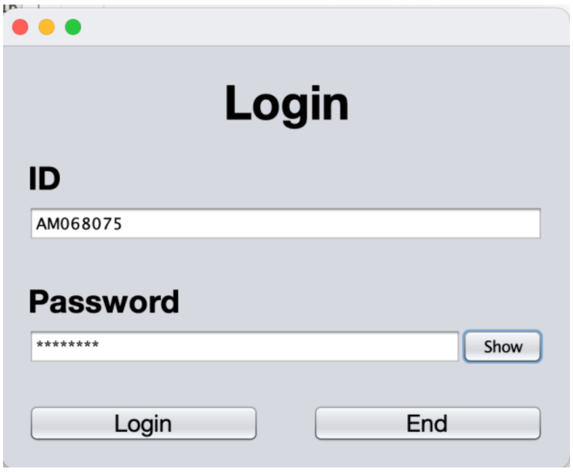
Figure 2.1.9.1: Change password

Users of any role can change their account password by clicking the "Change My Password" button from their profile page, which directs them to a window where they can make modifications. The window provides clear instructions regarding valid password criteria. Users are required to fill in three fields: the current (or old) password, the new password, and confirmation of the new password. If the old password is entered incorrectly or does not match the current password on record, an error message will be displayed. Similarly, if the new password and confirmation password do not match, an error message will appear. Once all required fields are correctly filled out, users can confirm their new password by clicking the "Confirm" button. The new password will then be recorded in the respective roles data file. The program will automatically update the user's profile, allowing them to view their newly created password from the profile page.

2.1.10 Additional Features

Show/Hide Password Field

Initially, all password fields, including those in the login and change password screens, are encrypted with asterisks ("*") to ensure robust security. An additional feature has been implemented to enhance user experience: the ability for users to view their password characters by clicking the "Show" button located beside each password field. Upon clicking the "Show" button, the password field will convert from asterisks to characters, displaying the actual password. Simultaneously, the button will change its label to "Hide". This allows users to view their password for verification or reference purposes. Should users wish to conceal their password again, they can simply click the "Hide" button. This action will revert the password field to its original asterisk encryption, while the button label changes back to "Show". Figure 2.1.10.1 and figure 2.1.10.2 demonstrate the functionality of this feature.

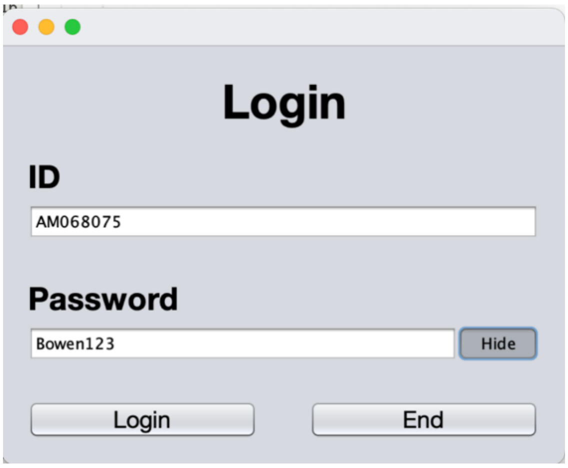


Login

ID: AM068075

Password: ********

Figure 2.1.10.1: Before clicking show



Login

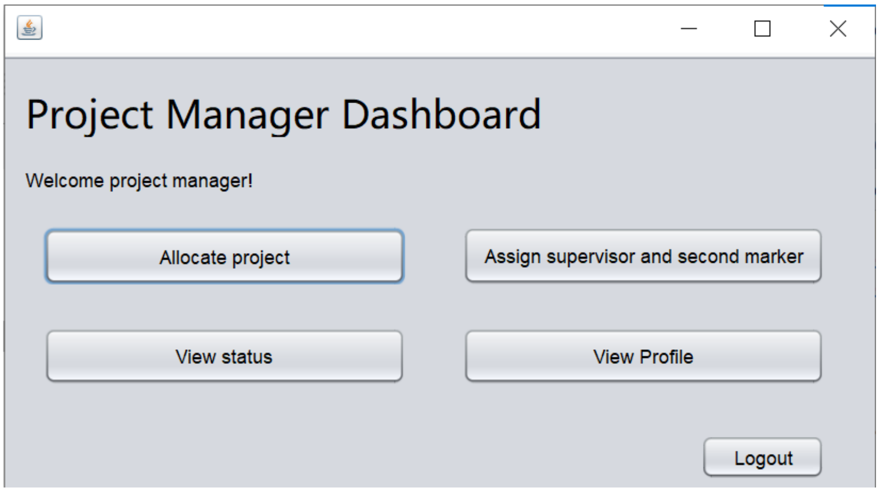
ID: AM068075

Password: Bowen123

Figure 2.1.10.2: After clicking show

2.2 Project Manager

2.2.1 Project Manager Menu Page



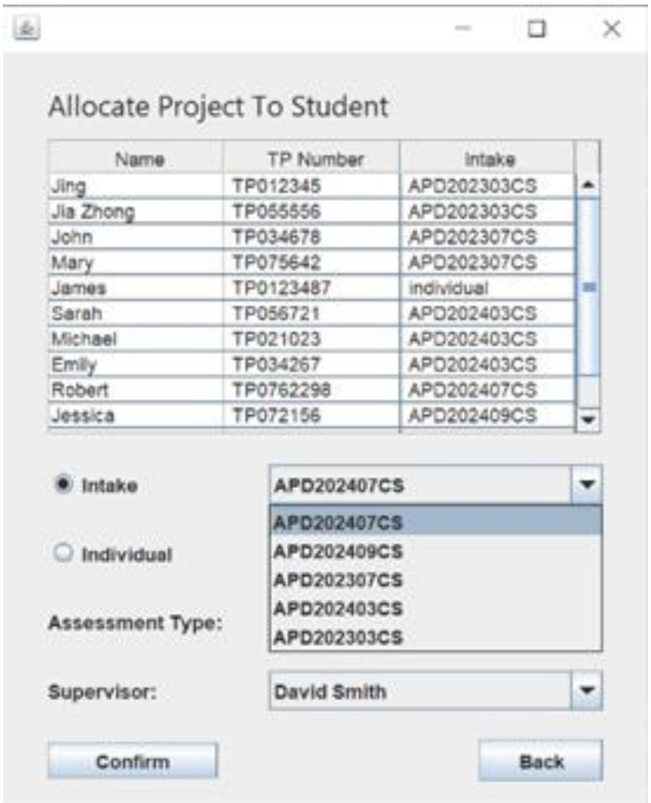
Project Manager Dashboard

Welcome project manager!

Figure 2.2.1.1: Project Manager Menu Page

The image above shows the menu of the Project Manager. This page covers the functionalities available to the Project Manager. At the top of the page, it is indicated that this is the Project Manager's dashboard, with a welcome message below the title to greet the user. The main section of the page features four buttons, each used to navigate to the corresponding page. In the top left corner is "allocate project," used to assign projects to students. The top right corner has "assign supervisor and second marker," used to assign lecturers to respective positions. The bottom left corner features "view status," used to check students' grades. The bottom right corner has "view profile," used to view the Project Manager's own information. At the bottom of the page, there is a "Logout" button that allows the user to exit the current page and return to the login page.

2.2.2 Allocate Project to Student



Allocate Project To Student

Name	TP Number	Intake
Jing	TP012345	APD202303CS
Jia Zhong	TP055556	APD202303CS
John	TP034678	APD202307CS
Mary	TP075642	APD202307CS
James	TP0123487	individual
Sarah	TP056721	APD202403CS
Michael	TP021023	APD202403CS
Emily	TP034267	APD202403CS
Robert	TP0762298	APD202407CS
Jessica	TP072156	APD202409CS

● Intake: APD202407CS

○ Individual

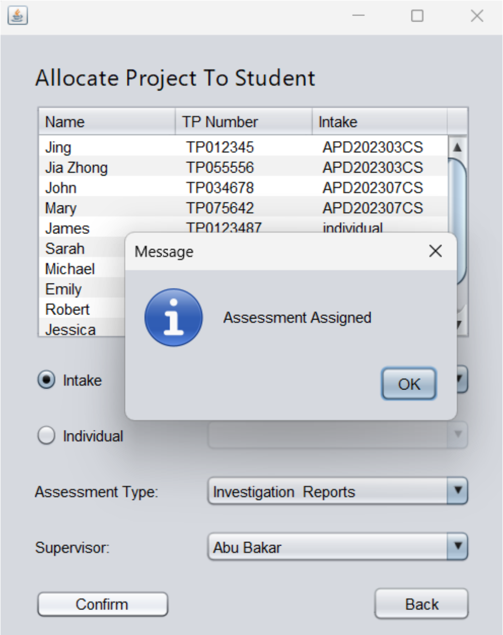
Assessment Type:

Supervisor: David Smith

Figure 2.2.2.1: Select allocate options

The image above depicts the “Allocate Project to Student” page. The table within displays all student names, TP Numbers, and intakes for the project manager’s review. Below, there are two radio buttons: “Intake” and “Individual.” These allow the project manager to choose the method of distributing reports. “Intake” enables the allocation of the same report to all students within a particular intake, while “Individual” allows for individualized allocation.

The combo box beside “Intake” displays all available intakes, while beside “Individual”, it displays all student TP Numbers for the project manager’s selection. Once selected, the project manager needs to choose the assessment type and specify who their supervisor is.



Allocate Project To Student

Name	TP Number	Intake
Jing	TP012345	APD202303CS
Jia Zhong	TP055556	APD202303CS
John	TP034678	APD202307CS
Mary	TP075642	APD202307CS
James	TP0123487	individual
Sarah		
Michael		
Emily		
Robert		
Jessica		

Message

Assessment Assigned

● Intake

○ Individual

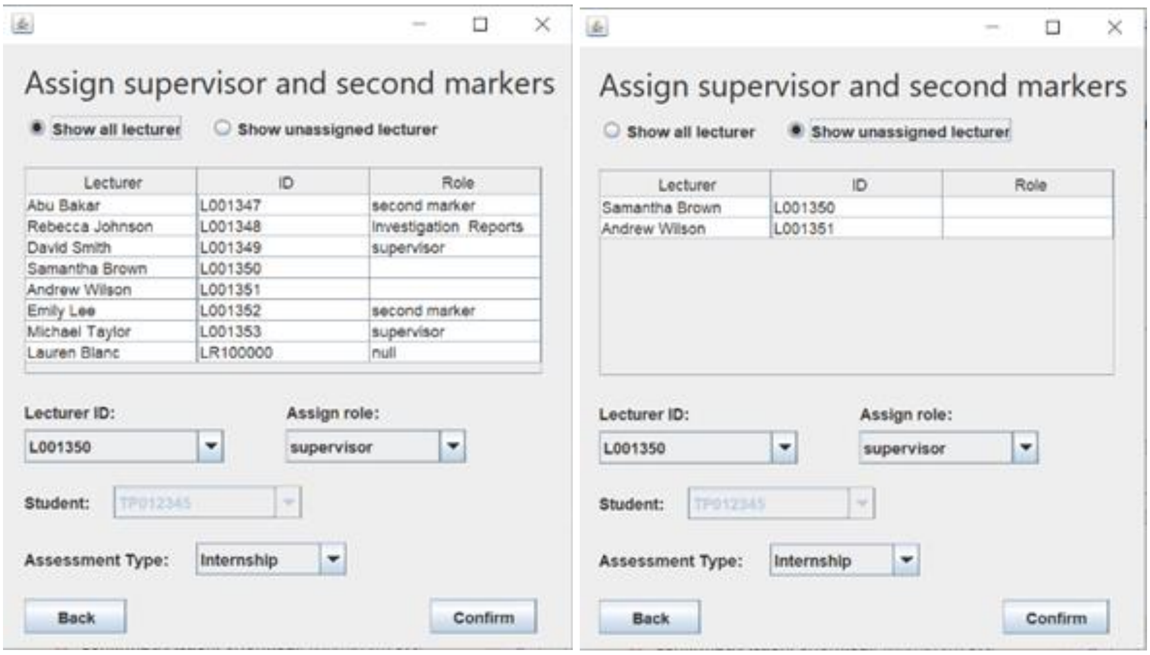
Assessment Type: Investigation Reports

Supervisor: Abu Bakar

Figure 2.2.2.2: Assigned successful message

Finally, clicking “Confirm” completes the allocation process, and a message box will pop up to show the data is recorded accordingly.

2.2.3 Assign Supervisor and Second Markers



Assign supervisor and second markers

● Show all lecturer ○ Show unassigned lecturer

Lecturer	ID	Role
Abu Bakar	L001347	second marker
Rebecca Johnson	L001348	Investigation Reports
David Smith	L001349	supervisor
Samantha Brown	L001350	
Andrew Wilson	L001351	
Emily Lee	L001352	second marker
Michael Taylor	L001353	supervisor
Lauren Blanc	LR100000	null

Lecturer ID: L001350

Assign role: supervisor

Student: TP012345

Assessment Type: Internship

Assign supervisor and second markers

○ Show all lecturer ● Show unassigned lecturer

Lecturer	ID	Role
Samantha Brown	L001350	
Andrew Wilson	L001351	

Lecturer ID: L001350

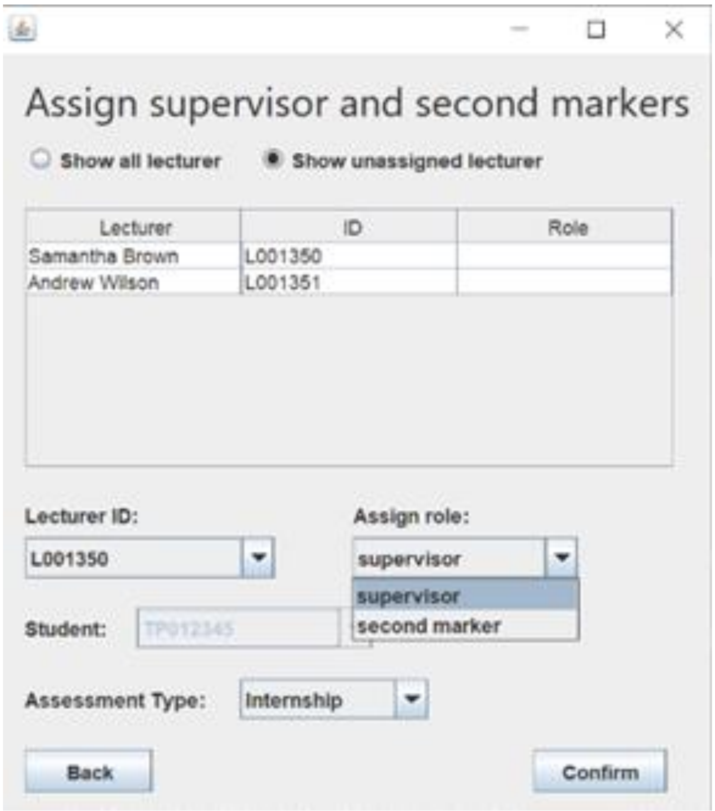
Assign role: supervisor

Student: TP012345

Assessment Type: Internship

Figure 2.2.3.1: Show all lecturer and unassigned lecturer

The image above depicts a page where registered lecturers are arranged as supervisors or second markers. Below the title, there are two radio buttons: “Show all lecturers” and “Show unassigned lecturers.” Selecting the first button will display all registered lecturers in the table below, showing their names, IDs, and assigned positions. Opting for the second button will only show lecturers who have not been assigned a position.



Assign supervisor and second markers

○ Show all lecturer ● Show unassigned lecturer

Lecturer	ID	Role
Samantha Brown	L001350	
Andrew Wilson	L001351	

Lecturer ID: L001350

Assign role: supervisor

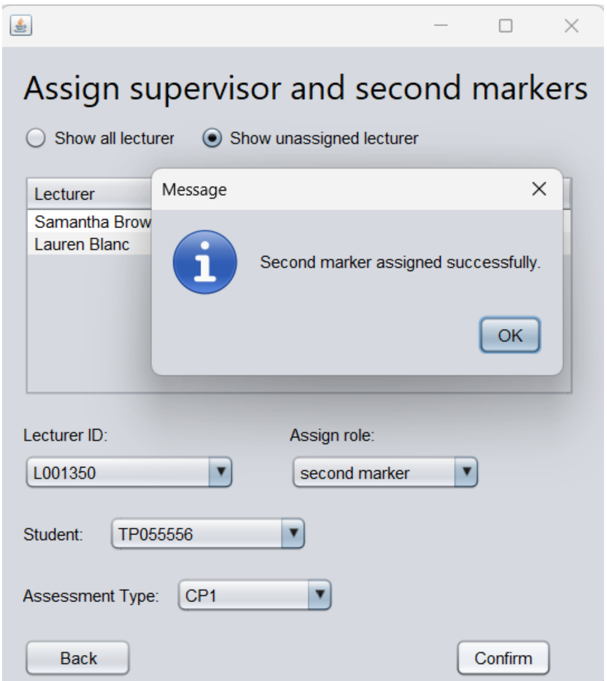
supervisor
second marker

Student: TP012345

Assessment Type: Internship

Figure 2.2.3.2: Select assign options

Below the table, there is a combo box for users to fill in options for assigning supervisors and second markers. The “Lecturer ID” box is used to specify the person for whom you want to assign a position. Clicking on the “Lecturer ID” box allows you to select lecturers who have not yet been assigned a position; other lecturers who have been assigned positions will not appear as options. The “assign role” option is for assigning a position to the specified person. If the assigned position is a supervisor, there is no need to assign students, and the student table cannot be selected. If the position is a second marker, students need to be assigned to the second marker for assessment, and you also need to select the items to be assessed by the second marker.



Assign supervisor and second markers

○ Show all lecturer ● Show unassigned lecturer

Lecturer
Samantha Brow
Lauren Blanc

Message

Second marker assigned successfully.

Lecturer ID: L001350

Assign role: second marker

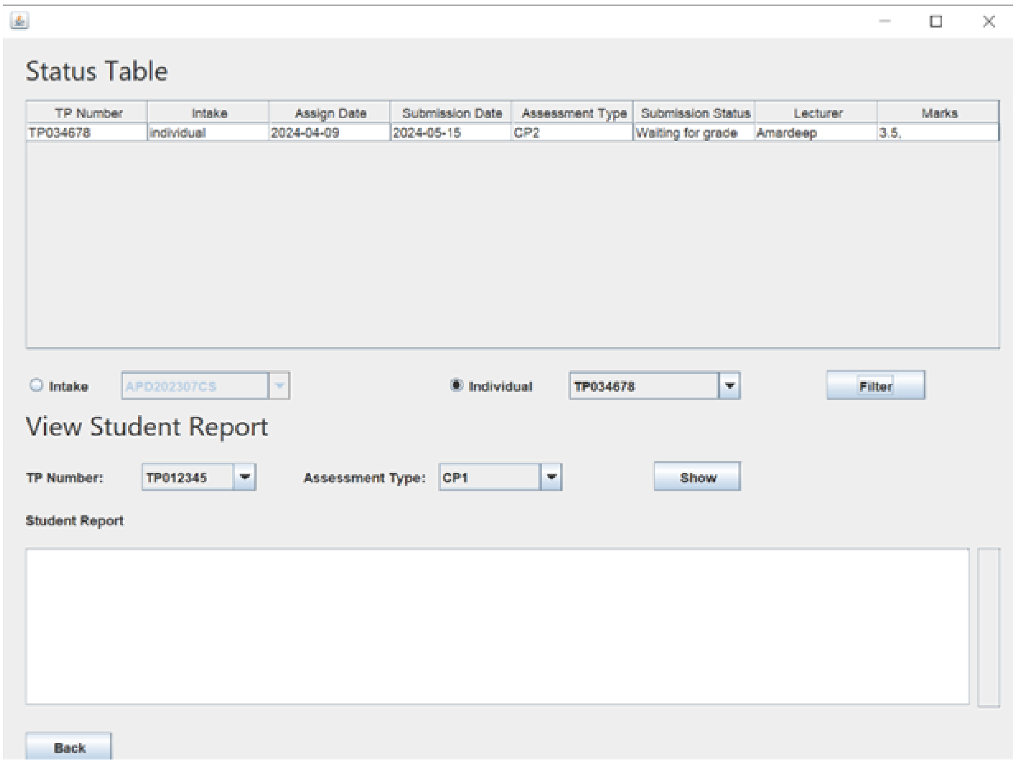
Student: TP055556

Assessment Type: CP1

Figure 2.2.3.2: Assigned successful message

While project manager finish to fill in the options, project manager can click “Confirm” button to assign the role to lecturer, and a message box will pop up, then the data will be recorded. If the role is second marker, the message that assign will be sent to the lecturer for waiting the lecturer to accept the distribute.

2.2.4 View Status



Status Table

TP Number	Intake	Assign Date	Submission Date	Assessment Type	Submission Status	Lecturer	Marks
TP034678	individual	2024-04-09	2024-05-15	CP2	Waiting for grade	Amardeep	3.5,

○ Intake APD202307CS ● Individual TP034678

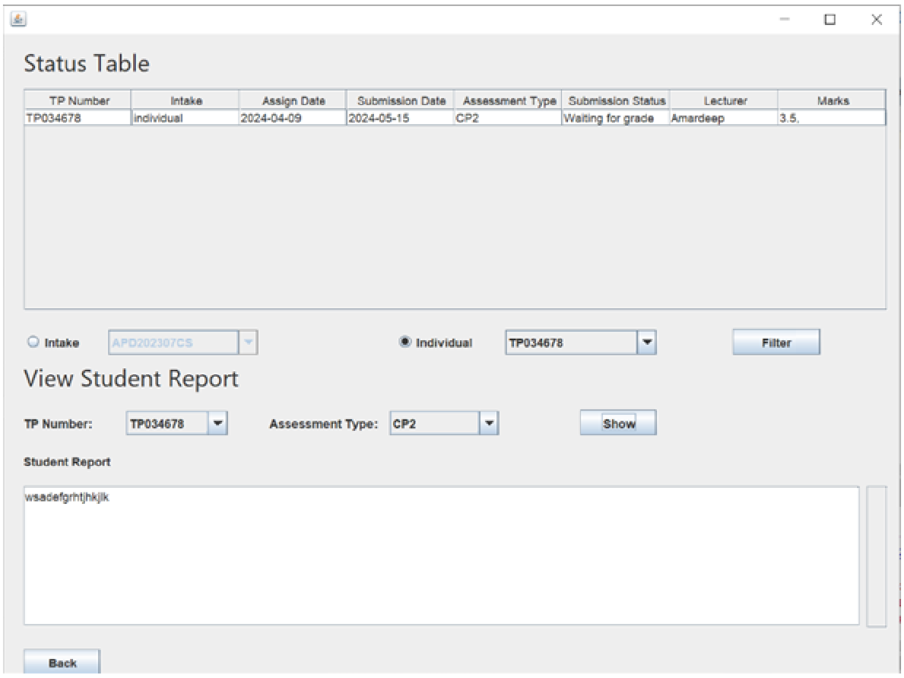
View Student Report

TP Number: TP012345 Assessment Type: CP1

Student Report

Figure 2.2.4.1: Show student's report status

The image provided is of the page where project managers can review students' grades. Below the table, there are two options: "Intake" and "Individual." Once selected, the project manager needs to choose the intake or TP Number they wish to review from the adjacent combo box. After making the selection, clicking "Filter" will display the relevant student data in the table. This data includes the student's TP Number, intake, date of report assignment, submission date, assignment type, submission status, lecturer's name, and their score.



Status Table

TP Number	Intake	Assign Date	Submission Date	Assessment Type	Submission Status	Lecturer	Marks
TP034678	individual	2024-04-09	2024-05-15	CP2	Waiting for grade	Amardeep	3.5.

○ Intake: APD202307CS ● Individual: TP034678

View Student Report

TP Number: TP034678 Assessment Type: CP2

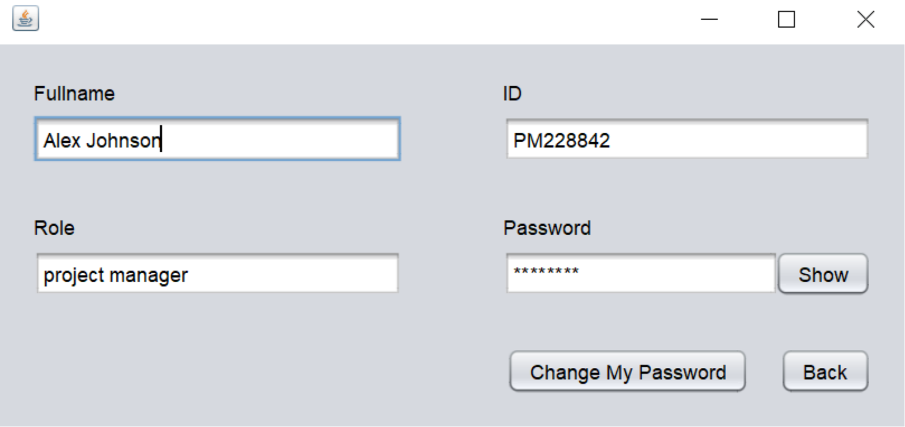
Student Report

wsadefgrhtjhkjlk

Figure 2.2.4.2: Show Student’s report

Below, there is an option to select the report of the desired student. After choosing the student and assignment type, clicking “Show” will display the contents of the student’s report in the text area below.

2.2.5 Project Manager View Profile



Fullname: Alex Johnson

ID: PM228842

Role: project manager

Password: ********

Figure 2.2.5.1: Profile Page of Project Manager

The image shows the Project Manager's Profile Page, which is used to view the current Project Manager's personal information. In the top left corner, the user's name is displayed. In the top right corner is the user's ID. In the bottom left corner is the user's position, and in the bottom right corner is the user's password. Next to the password field, there is a "Show" button that can be used to display the user's password. At the bottom of the page, there are two buttons: "Change My Password" and "Back." The former is used to navigate to the password change page to update the password, and the latter is used to return to the Project Manager's Menu.

2.3 Lecturer

2.3.1 Lecturer Menu Page

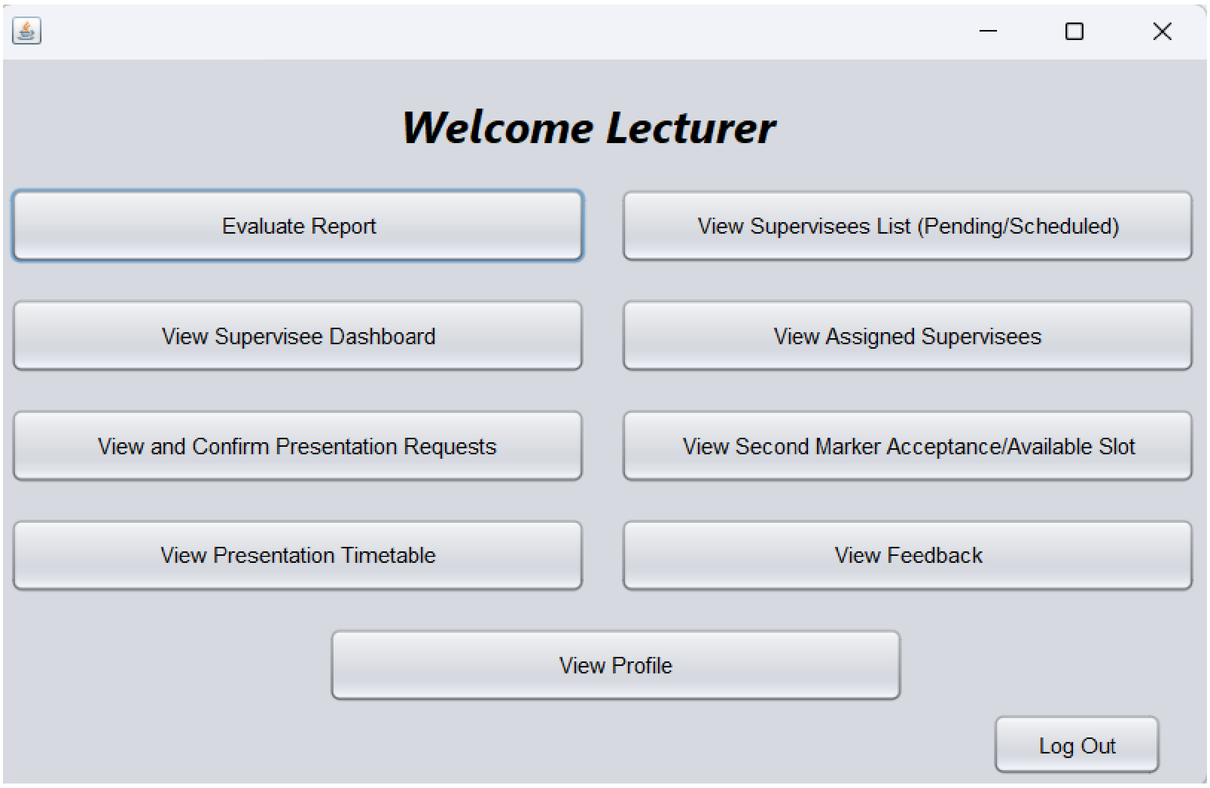

Welcome Lecturer
Evaluate Report
View Supervisees List (Pending/Scheduled)
View Supervisee Dashboard
View Assigned Supervisees
View and Confirm Presentation Requests
View Second Marker Acceptance/Available Slot
View Presentation Timetable
View Feedback
View Profile
Log Out

Figure 2.3.1.1: Lecturer Menu

Figure 2.3.1.1 shows the lecturer menu which will be shown when a lecturer logged in successfully. Lecturers will be able to perform various functions by clicking on the buttons. Each button will direct lecturers to the respective function page. The “Log Out” button will direct lecturers back to the “Login Page”.

2.3.2 Lecturer Evaluate Report

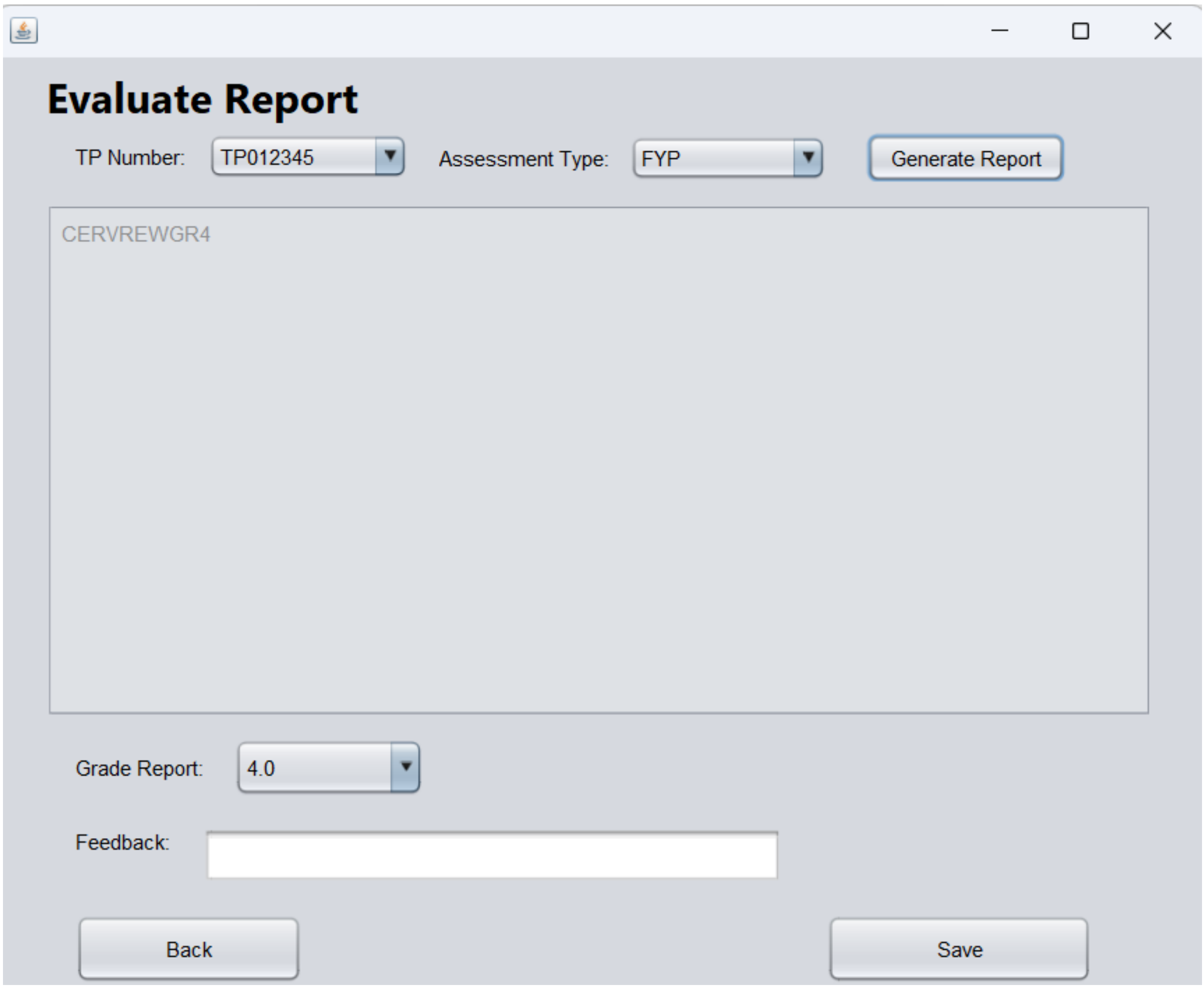
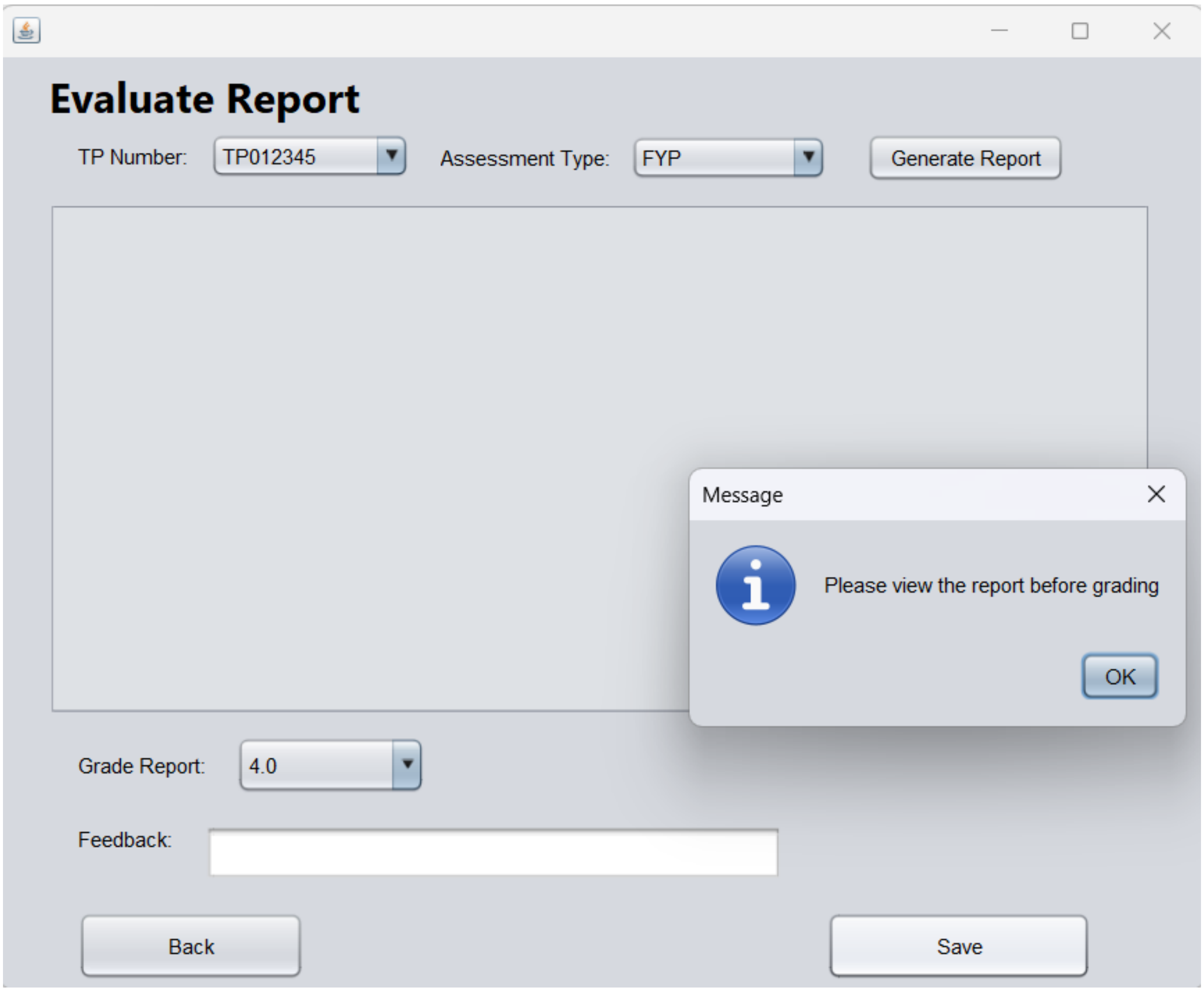


Figure 2.3.2.1: Lecturer load student report

If lecturer click on “Evaluate Report” button, Figure 2.3.2.1 will appear. Lecturers will make selection among the students that were assigned by project managers to them by selecting the student ID. After a student is selected, the list of assessments of the student which is associated with the lecturer in the “ReportDetails.txt” file will be loaded into the combo box for selection. Lecturer will then click “Generate Report” to view the answer script based on the TP Number and assessment selected.



Evaluate Report

TP Number: TP012345

Assessment Type: FYP

Grade Report: 4.0

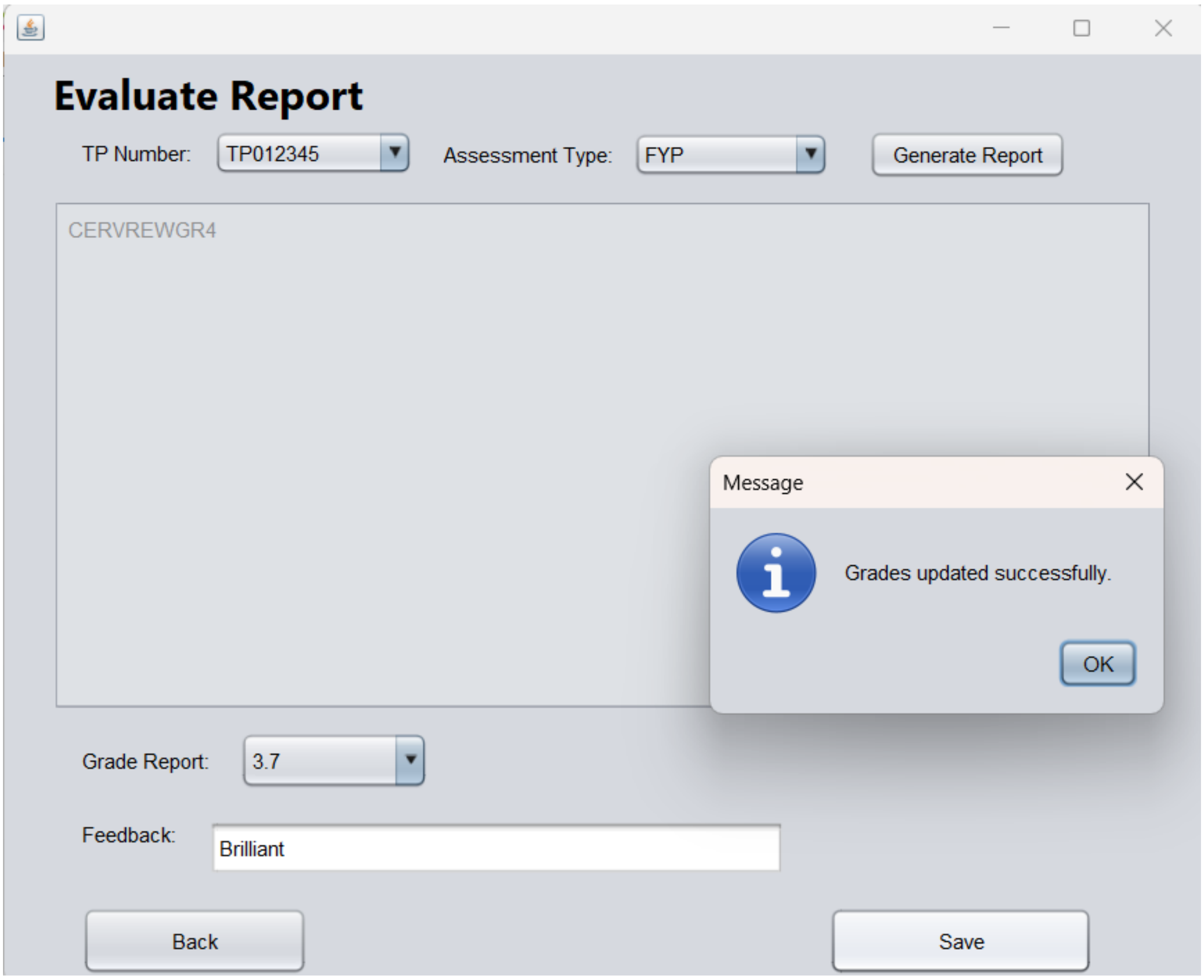
Feedback:

Message

Please view the report before grading

Figure 2.3.2.2: Grading when report not generated.

Based on Figure 2.3.2.2, an error message indicating to view the report before assigning grade to the student is shown to lecturer. This happens when lecturers save the grade of the assessment of the student without generating the report submitted by the student. This validation is to make sure that lecturers really view the student's report before grading and not simply giving a grade.



Evaluate Report

TP Number: TP012345

Assessment Type: FYP

CERVREWGR4

Message

Grades updated successfully.

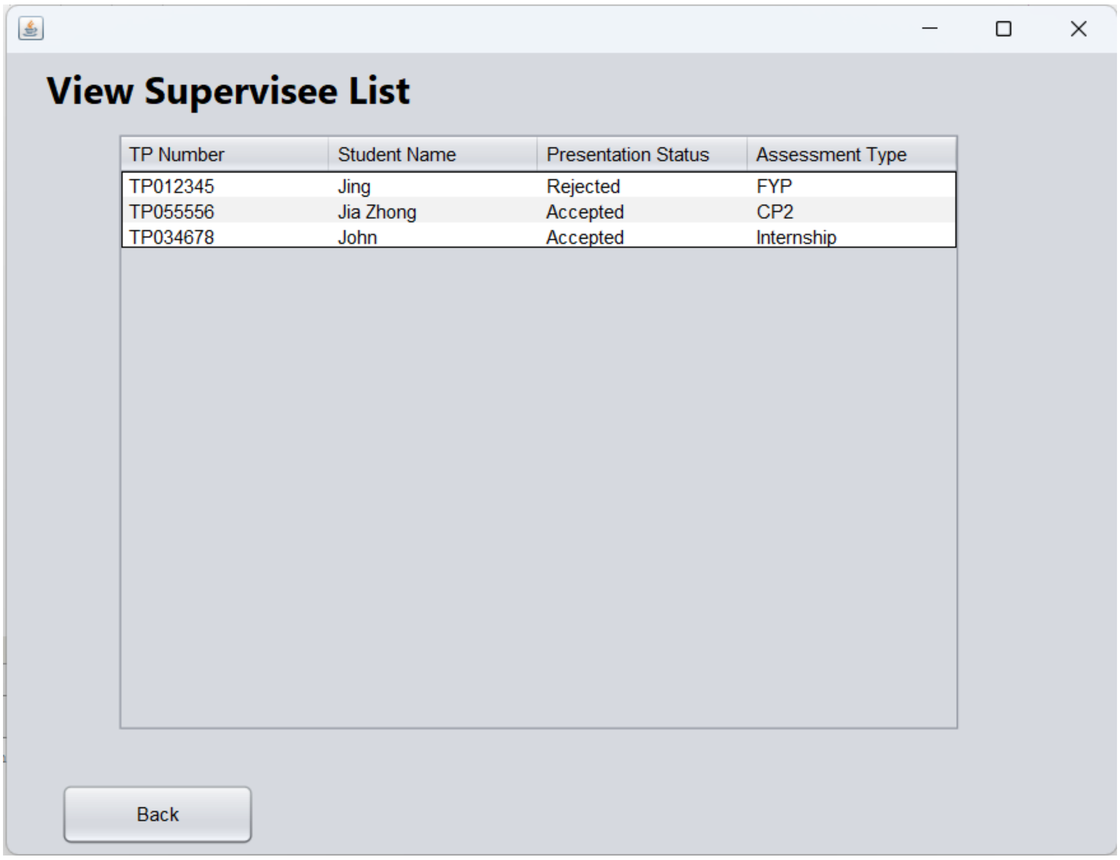
Grade Report: 3.7

Feedback: Brilliant

Figure 2.3.2.3: Evaluate Report successfully.

Based on Figure 2.3.2.3, after lecturers have generated the report of a particular assessment of the student, lecturers will view the report, give grade and provide feedback on the report if any. Then, clicking on "Save" will show a message indicating the grade is successfully appended into the "Result.txt" file. At the same time, the status of the report will be updated to "Graded" in the "ReportDetails.txt" file. There are 3 marks contained within a report, which are first grade, second grade and final grade. If the lecturer is the second marker, the mark will be updated into the second grade and final grade will be calculated by finding the average of mark between first grade and second grade, else the mark will be updated only into first grade. Lastly, clicking "Back" directs lecturers back to the lecturer menu.

2.3.3 View Supervisee List



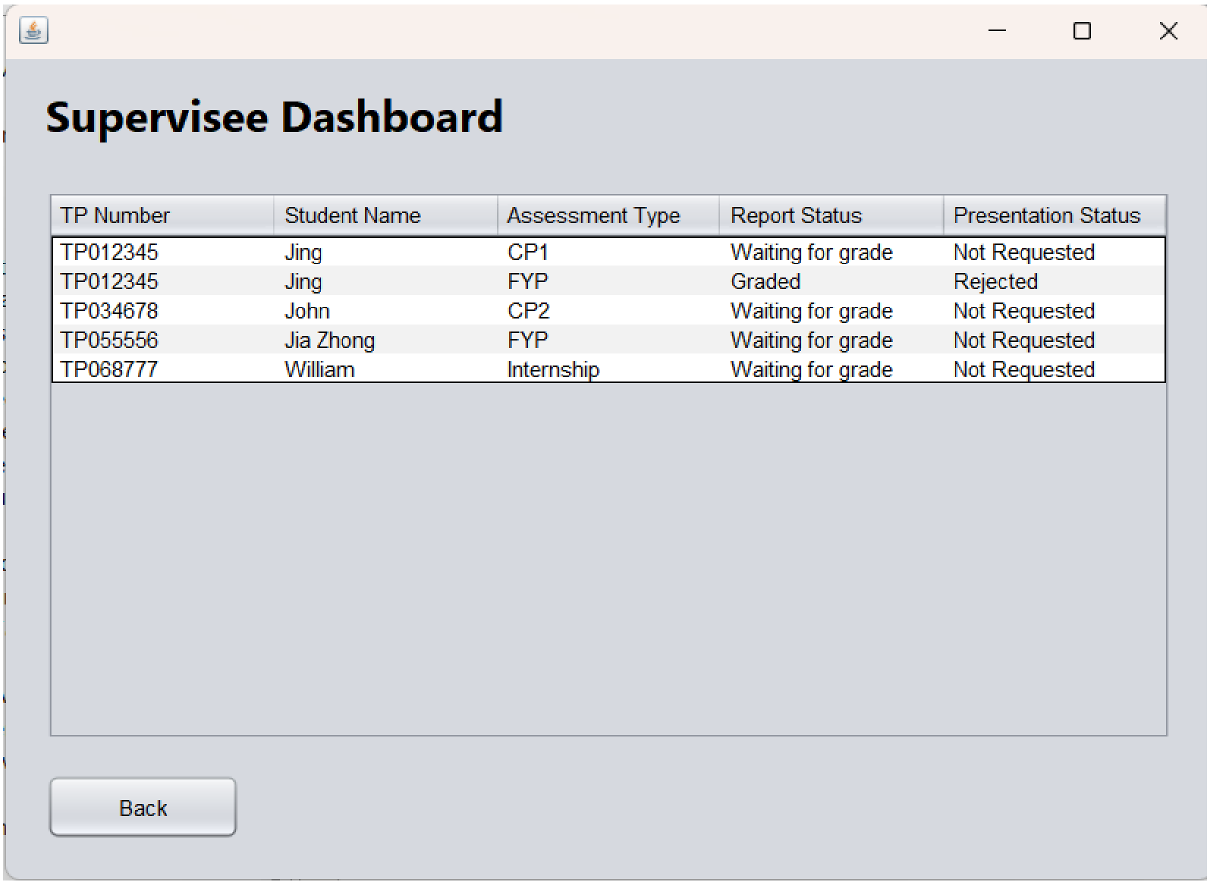
View Supervisee List

TP Number	Student Name	Presentation Status	Assessment Type
TP012345	Jing	Rejected	FYP
TP055556	Jia Zhong	Accepted	CP2
TP034678	John	Accepted	Internship

Figure 2.3.3.1: View Supervisee List

Based on figure 2.3.3.1, lecturers who are supervisors will be able to view a list of supervisees based on their presentation details such as assessment type and status. The presentation data of the supervisee are only visible to the lecturer if the current session's lecturer supervise the presentation. Supervisors can then press the back button to return to lecturer menu.

2.3.4 View Supervisee Dashboard



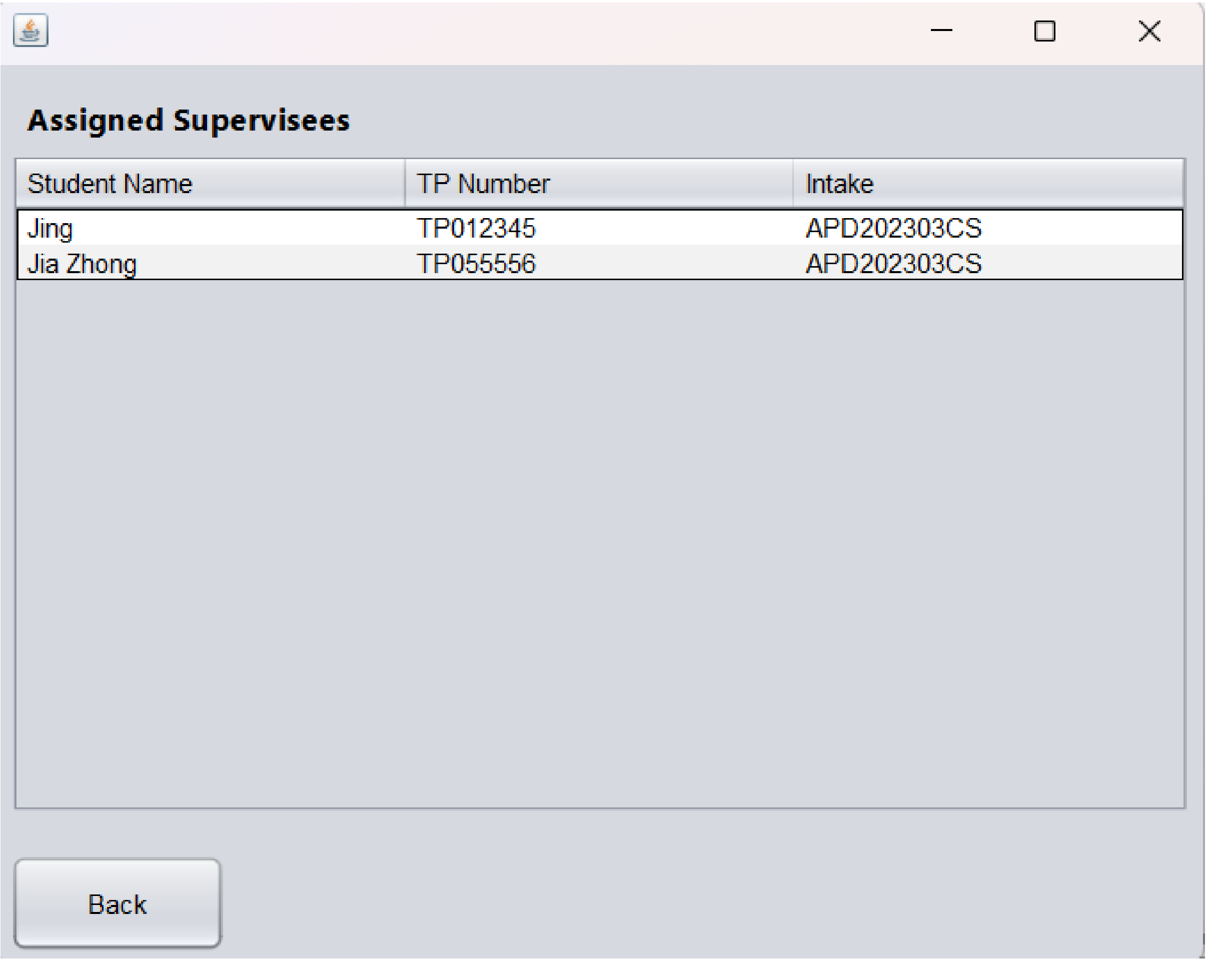
Supervisee Dashboard

TP Number	Student Name	Assessment Type	Report Status	Presentation Status
TP012345	Jing	CP1	Waiting for grade	Not Requested
TP012345	Jing	FYP	Graded	Rejected
TP034678	John	CP2	Waiting for grade	Not Requested
TP055556	Jia Zhong	FYP	Waiting for grade	Not Requested
TP068777	William	Internship	Waiting for grade	Not Requested

Figure 2.3.4.1: View Supervisee Dashboard

Based on figure 2.3.4.1, the table allows lecturers to view the overall progress of students in their assessments. Lecturers will be able to know whether a student has submitted the report and waiting for grade, or the report is not submitted or the report of the student is graded or requested for presentation for the assessment. Pressing the back button would direct the lecturer back to the lecturer menu.

2.3.5 View Assigned Supervisees



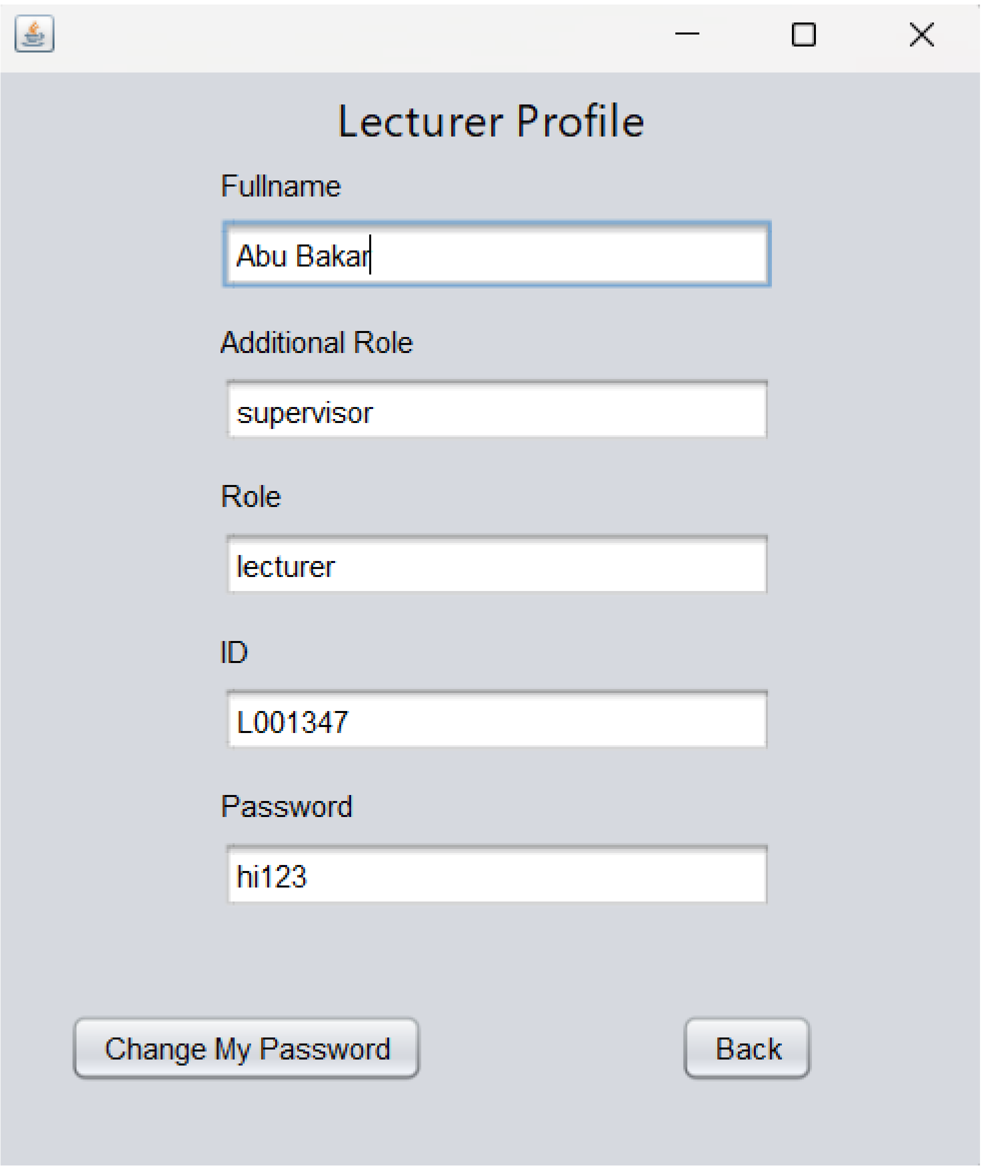
Assigned Supervisees

Student Name	TP Number	Intake
Jing	TP012345	APD202303CS
Jia Zhong	TP055556	APD202303CS

Figure 2.3.5.1: View Assigned Supervisee

Based on figure 2.3.5.1, lecturers who are supervisors can view their assigned supervisees from the table. The supervisee is assigned to the supervisor when the project manager assign projects to a student and assign a lecturer to become supervisor and be responsible for supervising the project of the student. Pressing the back button would direct the supervisor back to the lecturer menu.

2.3.6 Lecturer View Profile



Lecturer Profile

Fullname: Abu Bakar

Additional Role: supervisor

Role: lecturer

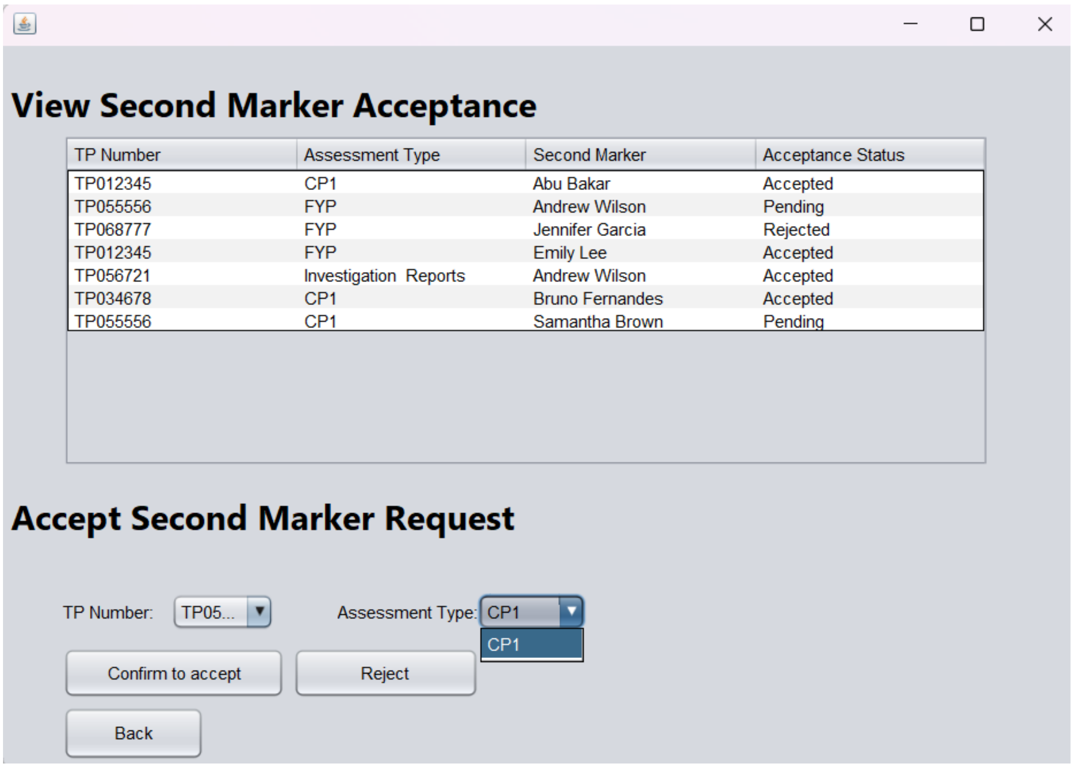
ID: L001347

Password: hi123

Figure 2.3.6.1: Lecturer View Profile

Similar to other roles, lecturer can also view their profile with an additional information which is the additional role. Lecturer can be either a supervisor or a second marker. All the text fields in the lecturer profile page are not available for editing. The only detail that can be changed is the password, but it cannot be changed directly in the text field, lecturer will have to press on the “Change My Password” button in order to access the change password page to change password as shown previously. Pressing the “Back” button would direct lecturers back to lecturer menu.

2.3.7 View Second Marker Acceptance and Accept Second Marker Request



View Second Marker Acceptance

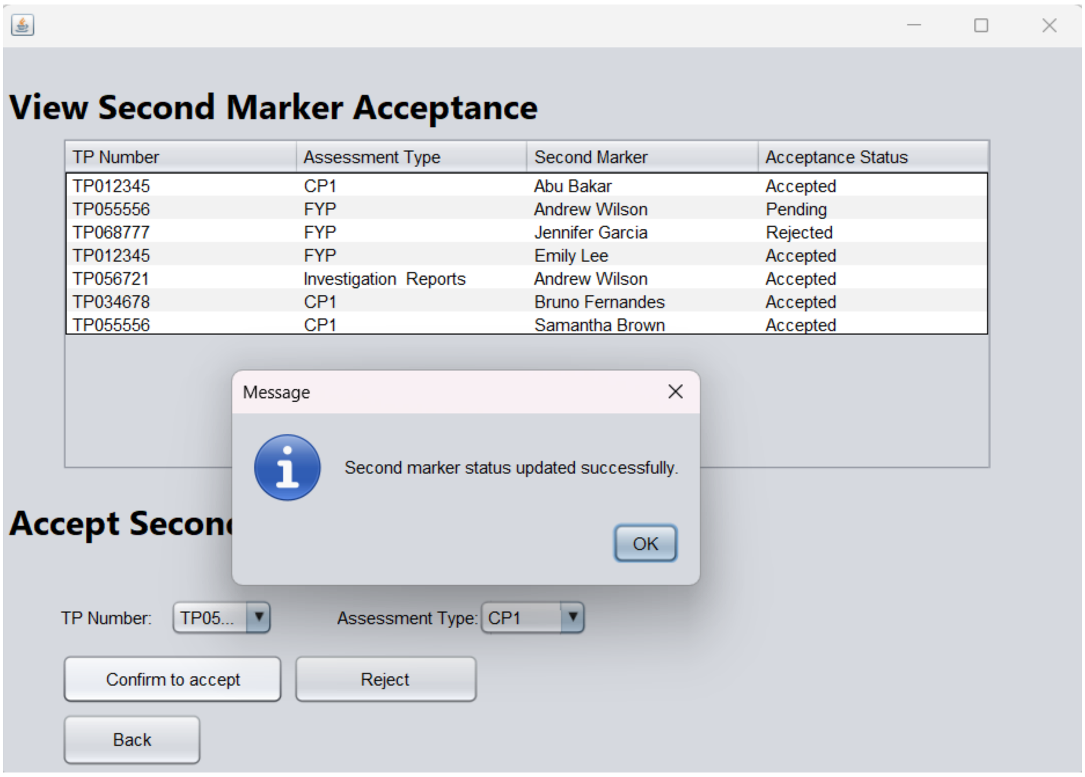
TP Number	Assessment Type	Second Marker	Acceptance Status
TP012345	CP1	Abu Bakar	Accepted
TP055556	FYP	Andrew Wilson	Pending
TP068777	FYP	Jennifer Garcia	Rejected
TP012345	FYP	Emily Lee	Accepted
TP056721	Investigation Reports	Andrew Wilson	Accepted
TP034678	CP1	Bruno Fernandes	Accepted
TP055556	CP1	Samantha Brown	Pending

Accept Second Marker Request

TP Number: TP05...

Assessment Type: CP1

Figure 2.3.7.1: View Second Marker Acceptance and Accept Second Marker Request



View Second Marker Acceptance

TP Number	Assessment Type	Second Marker	Acceptance Status
TP012345	CP1	Abu Bakar	Accepted
TP055556	FYP	Andrew Wilson	Pending
TP068777	FYP	Jennifer Garcia	Rejected
TP012345	FYP	Emily Lee	Accepted
TP056721	Investigation Reports	Andrew Wilson	Accepted
TP034678	CP1	Bruno Fernandes	Accepted
TP055556	CP1	Samantha Brown	Accepted

Message

Second marker status updated successfully.

TP Number: TP05...

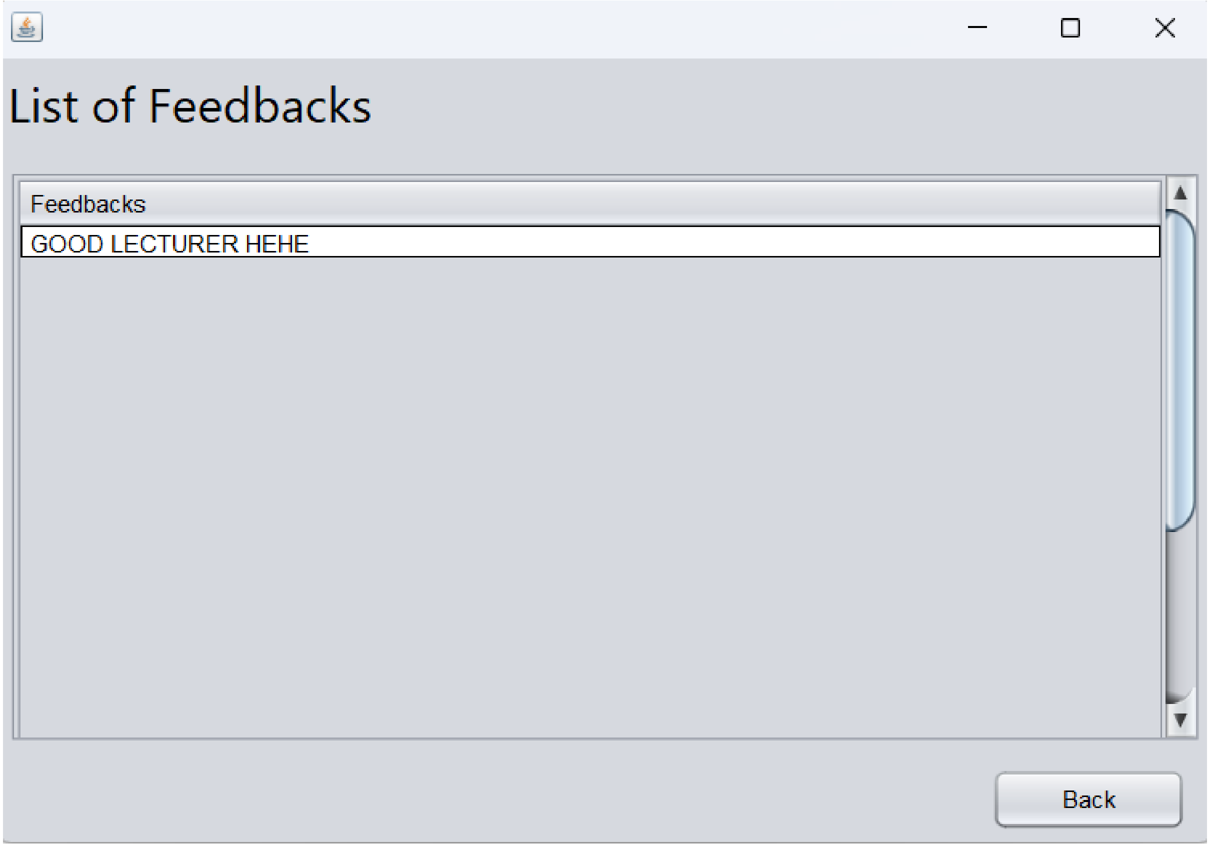
Assessment Type: CP1

Figure 2.3.7.2: Second marker request accepted

When lecturers clicked on the “View Second Marker Acceptance/Available Slot” button, the page in figure 2.3.7.1 will show up. The logged in lecturer will be able to view all the second marker acceptance record in a table, regardless of the lecturer. Then at the bottom half part, it is the section where the lecturer can see if there is any requests for them to be second marker. If the combo boxes are not empty and loaded with TP number with its respective assessment type, it means there is request for the logged in lecturer to be a second marker.

The lecturer will then choose a TP number, which is student ID and upon choosing the student ID, its respective assessment type will be shown in the assessment type combo box. After selecting one of the requests, the lecturer will have to choose to accept the second marker request or reject. Either accepting or rejecting the second marker request will output a pop-up message indicating the acceptance status is updated for the request as shown in figure 2.3.7.2. Once accept, additional role of second marker will be assigned to the lecturer, rejecting the request would not assign second marker role to the lecturer. Similar to other pages, pressing back button would return back to lecturer menu.

2.3.8 View Feedback



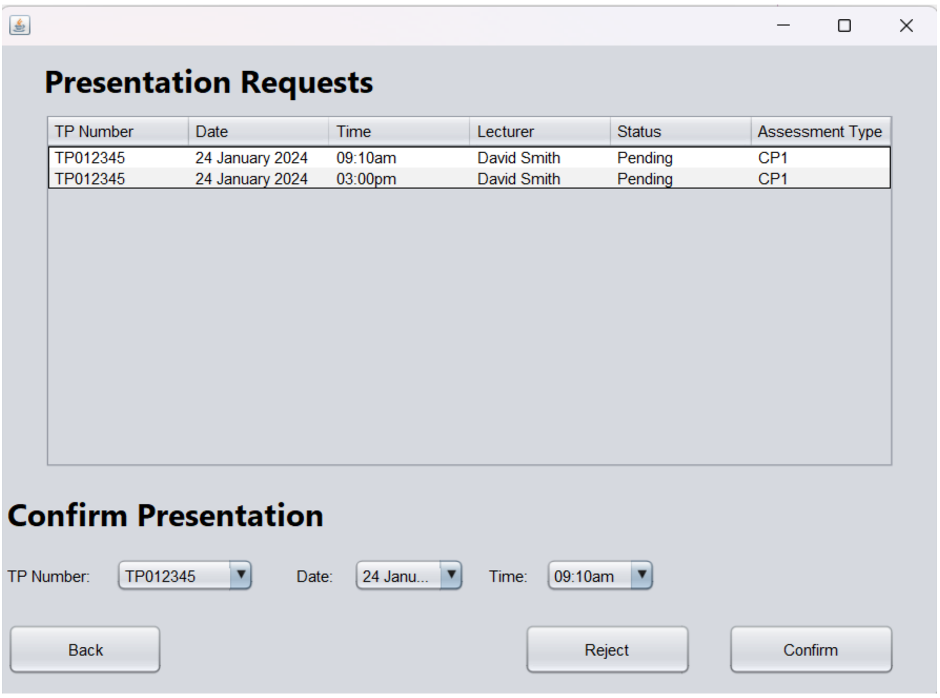
List of Feedbacks

Feedbacks
GOOD LECTURER HEHE

Figure 2.3.8.1: List of Feedback

When lecturer click on the “View Feedback” button, the page in figure 2.3.8.1 will show up. Lecturer will be able to see all the feedback from students so that they can improve their teaching methods or materials. The feedback is anonymous for lecturer point of view. Lecturers are not allowed to see which specific student provide them the feedback. This is to ensure the privacy of students.

2.3.9 View and Confirm Presentation Request



Presentation Requests

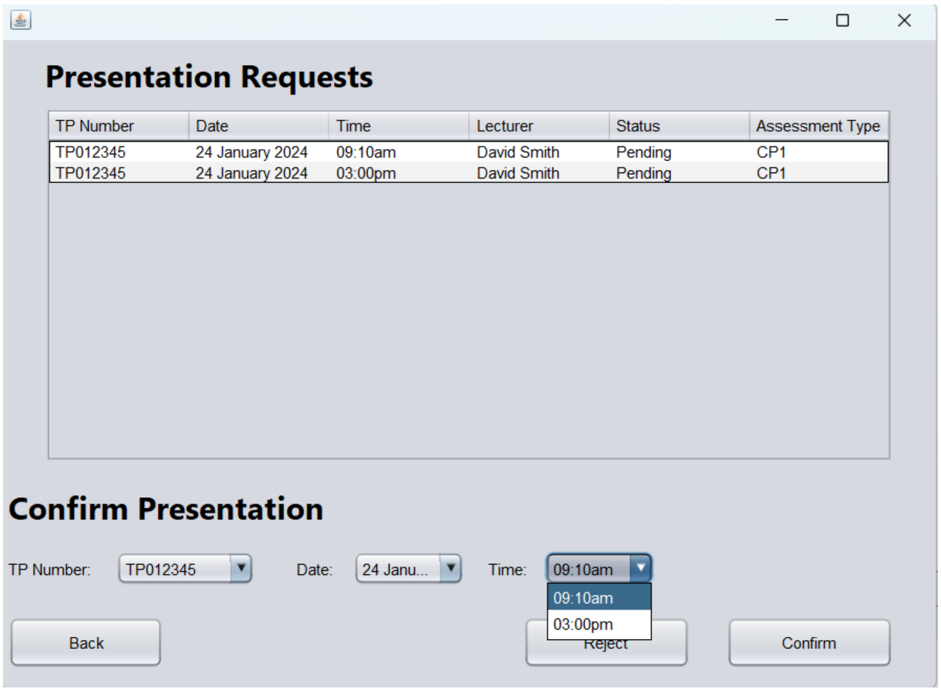
TP Number	Date	Time	Lecturer	Status	Assessment Type
TP012345	24 January 2024	09:10am	David Smith	Pending	CP1
TP012345	24 January 2024	03:00pm	David Smith	Pending	CP1

Confirm Presentation

TP Number: TP012345 Date: 24 Janu... Time: 09:10am

Figure 2.3.9.1: View and Confirm Presentation Request

When lecturers click on the "View and Confirm Presentation Request" button, lecturers will be directed to the page as shown in figure 2.3.9.1. They will be able to view all the presentation requests made by students towards them. A student can make several requests for a presentation with different time so that lecturers can accept the one that fits their schedule and reject the ones which doesn't fit their schedule.



Presentation Requests

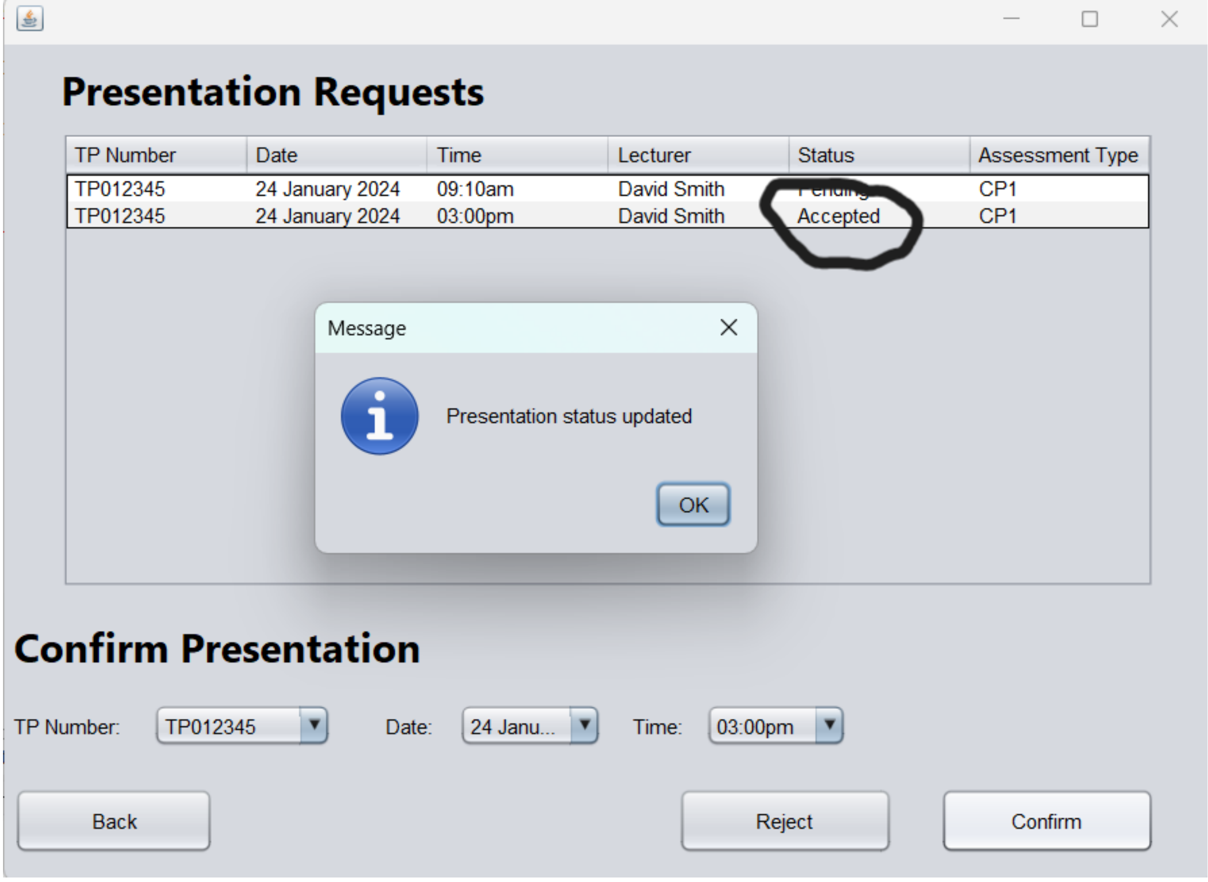
TP Number	Date	Time	Lecturer	Status	Assessment Type
TP012345	24 January 2024	09:10am	David Smith	Pending	CP1
TP012345	24 January 2024	03:00pm	David Smith	Pending	CP1

Confirm Presentation

TP Number: TP012345 Date: 24 Janu... Time: 09:10am

Figure 2.3.9.2: Selecting presentation request

At the confirm presentation section, lecturer will first select the TP number of the student. Then, the date and time of the presentation requests made by the student will appear in the date and time combo boxes. Lecturers will then select the date and time and proceed by accepting or rejecting the request.



Presentation Requests

TP Number	Date	Time	Lecturer	Status	Assessment Type
TP012345	24 January 2024	09:10am	David Smith	Pending	CP1
TP012345	24 January 2024	03:00pm	David Smith	Accepted	CP1

Message

Presentation status updated

Confirm Presentation

TP Number: TP012345

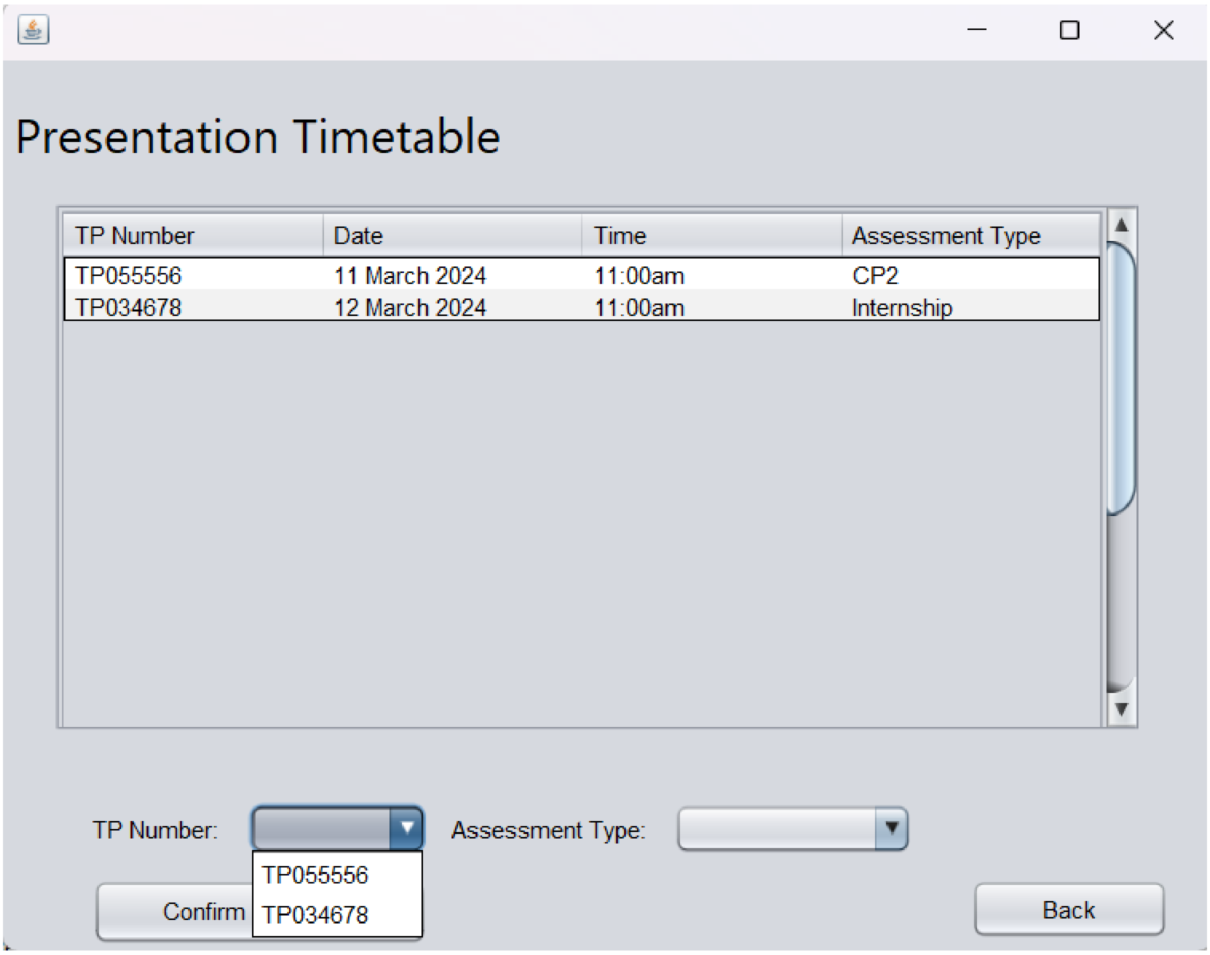
Date: 24 Janu...

Time: 03:00pm

Figure 2.3.9.3: Presentation request status updated

Based on figure 2.3.9.3, a pop-up message indicating presentation status updated will be outputted. In this case, the lecturer accepted the presentation request by clicking the "Confirm" button. The status is updated to "Accepted" as you can see in the figure 2.3.9.3.

2.3.10 View Presentation Timetable

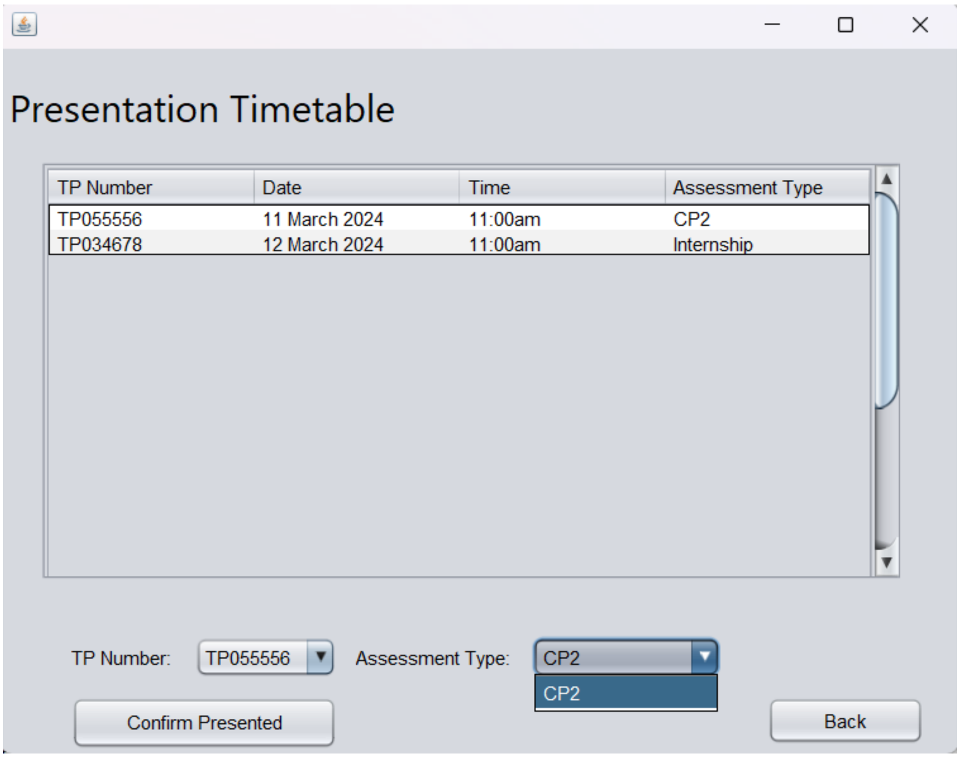


Presentation Timetable

TP Number	Date	Time	Assessment Type
TP055556	11 March 2024	11:00am	CP2
TP034678	12 March 2024	11:00am	Internship

Figure 2.3.10.1: Presentation Timetable

If a lecturer accepted the assessment presentation request of student. It will appear in the lecturer's presentation timetable. The bottom section is where the lecturers select the student and the respective assessment type and update the status of the presentation of the student to "Presented" if the student has already done the presentation to the lecturer. Clicking the TP number combo box should show a list of student ID to choose from.



Presentation Timetable

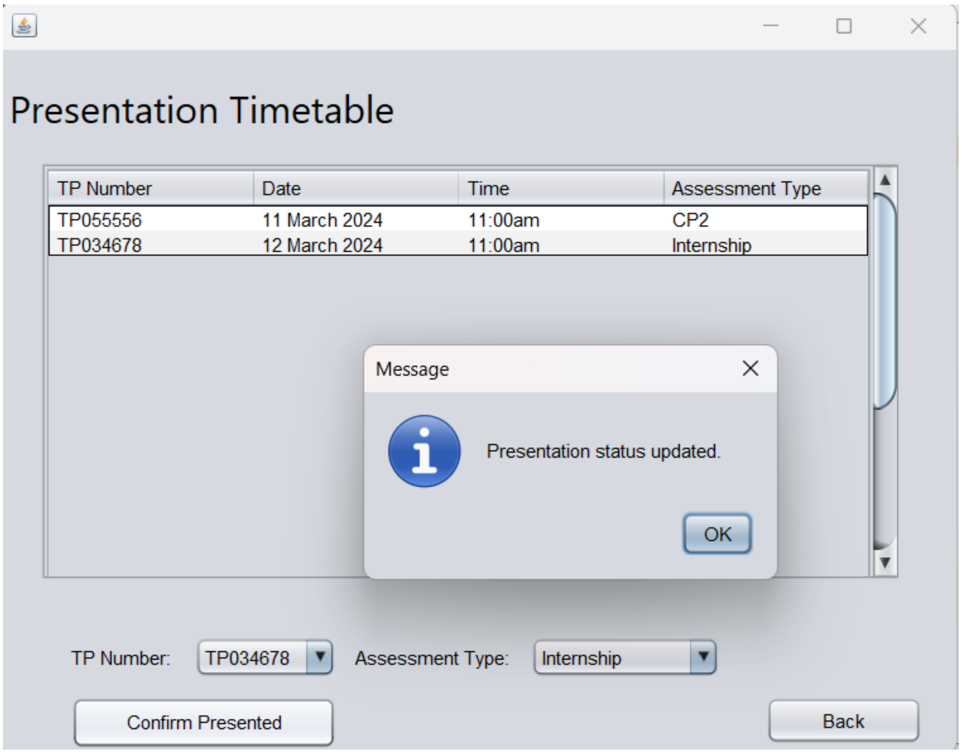
TP Number	Date	Time	Assessment Type
TP055556	11 March 2024	11:00am	CP2
TP034678	12 March 2024	11:00am	Internship

TP Number: TP055556

Assessment Type: CP2

Figure 2.3.10.2: Presentation Timetable

After student ID which is the TP number is selected, the assessment type of the student which the presentation is accepted by the lecturer will be shown in a list in the combo box. Then, lecturers can proceed by clicking the “Confirm Presented” button to update the presentation status of the assessment type of a particular student to “Presented”.



Presentation Timetable

TP Number	Date	Time	Assessment Type
TP055556	11 March 2024	11:00am	CP2
TP034678	12 March 2024	11:00am	Internship

Message

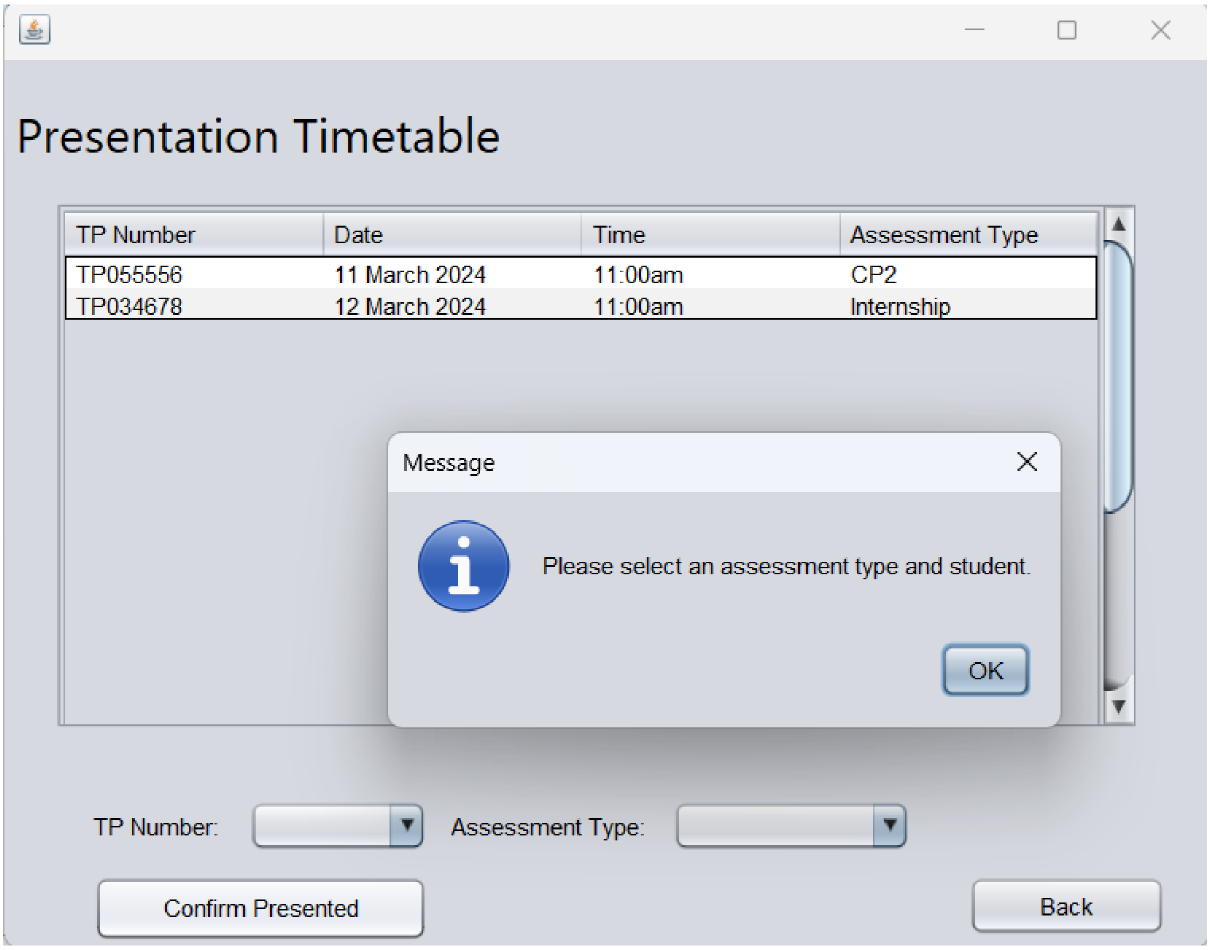
Presentation status updated.

TP Number: TP034678

Assessment Type: Internship

Figure 2.3.10.3: Presentation Timetable

Once lecturers clicked the "Confirm Presented" button, a pop-up message will be shown indicating that the status of the presentation is updated to "Presented".



Presentation Timetable

TP Number	Date	Time	Assessment Type
TP055556	11 March 2024	11:00am	CP2
TP034678	12 March 2024	11:00am	Internship

Message

Please select an assessment type and student.

Figure 2.3.10.4: Presentation Timetable

Validation is made where lecturers do not select any student. If the case happens, a pop-up message will show up telling the lecturer to select a student and their respective assessment type before updating the presentation status.

2.4 Student

2.4.1 Student Menu Page

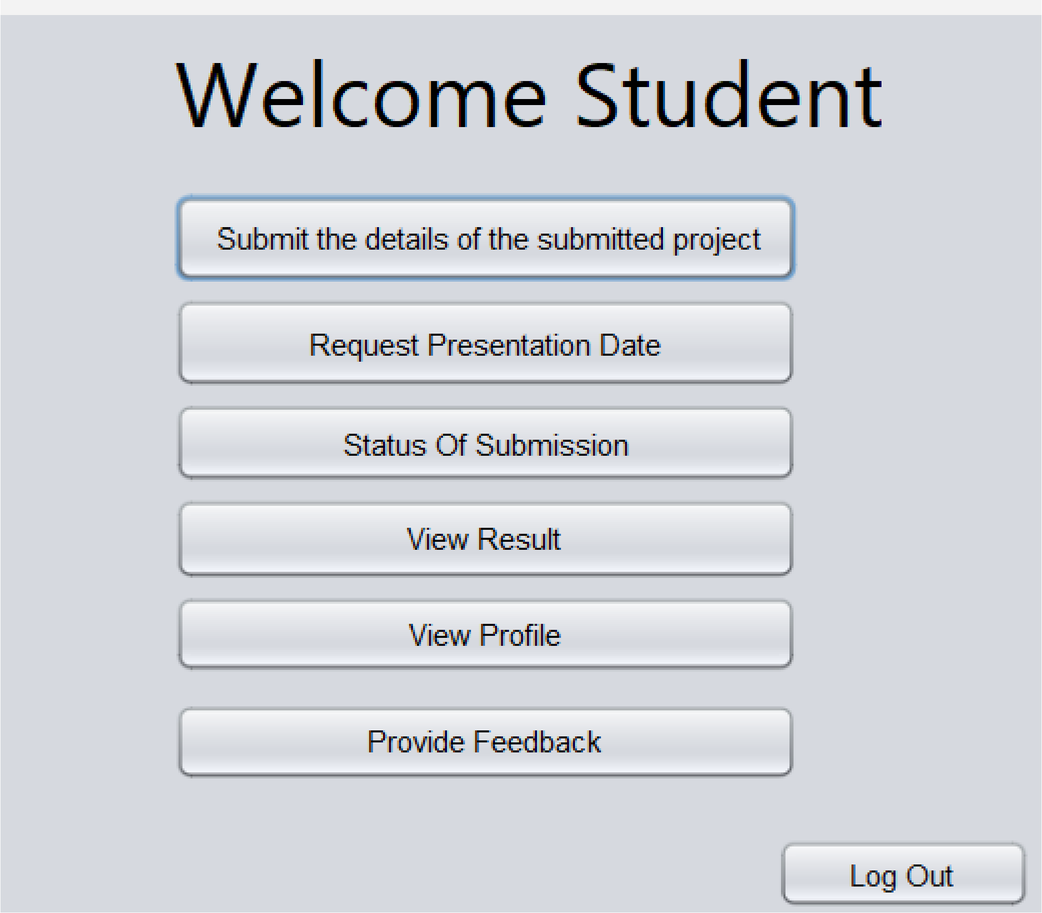


Figure 2.4.1.1: Student Menu Page

Figure 2.4.1.1 shows that the student menu that appear after a user with student role successfully login into the system. On this page, the student contains various function that's show on the menu page. When student click the button, it will direct go to the window which will be explain on this document. The "Log Out" will allows students to logout their account and return to the Login page.

2.4.2 Submit the details of the submitted project

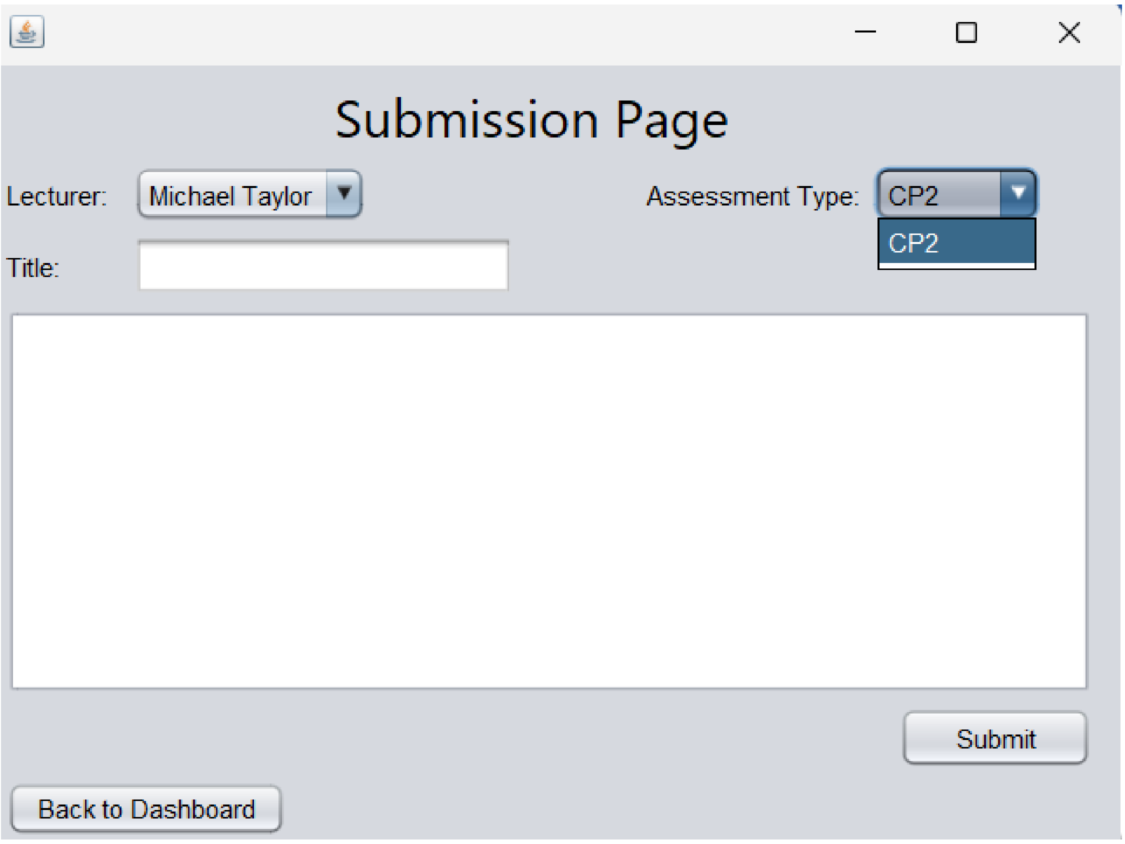
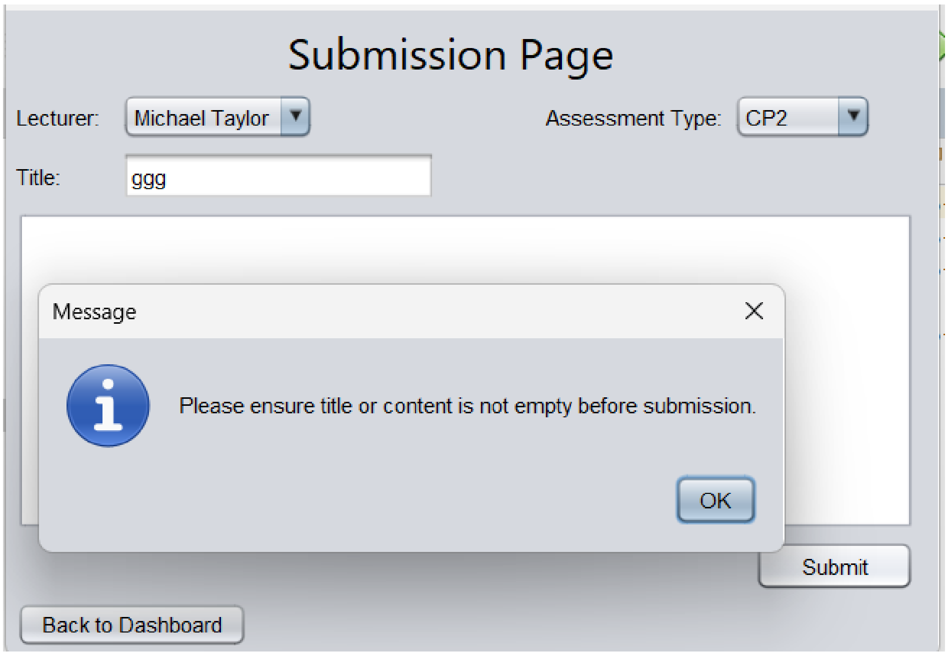


Figure 2.4.2.1: Submission page

After clicking the button “Submit the details of the submitted project”, “Submission page” will appear like Figure 2.4.2.1. Students can choose the assessment type and submit the report. After choosing the assessment type, the lecturer’s name will display automatically on the combo box.

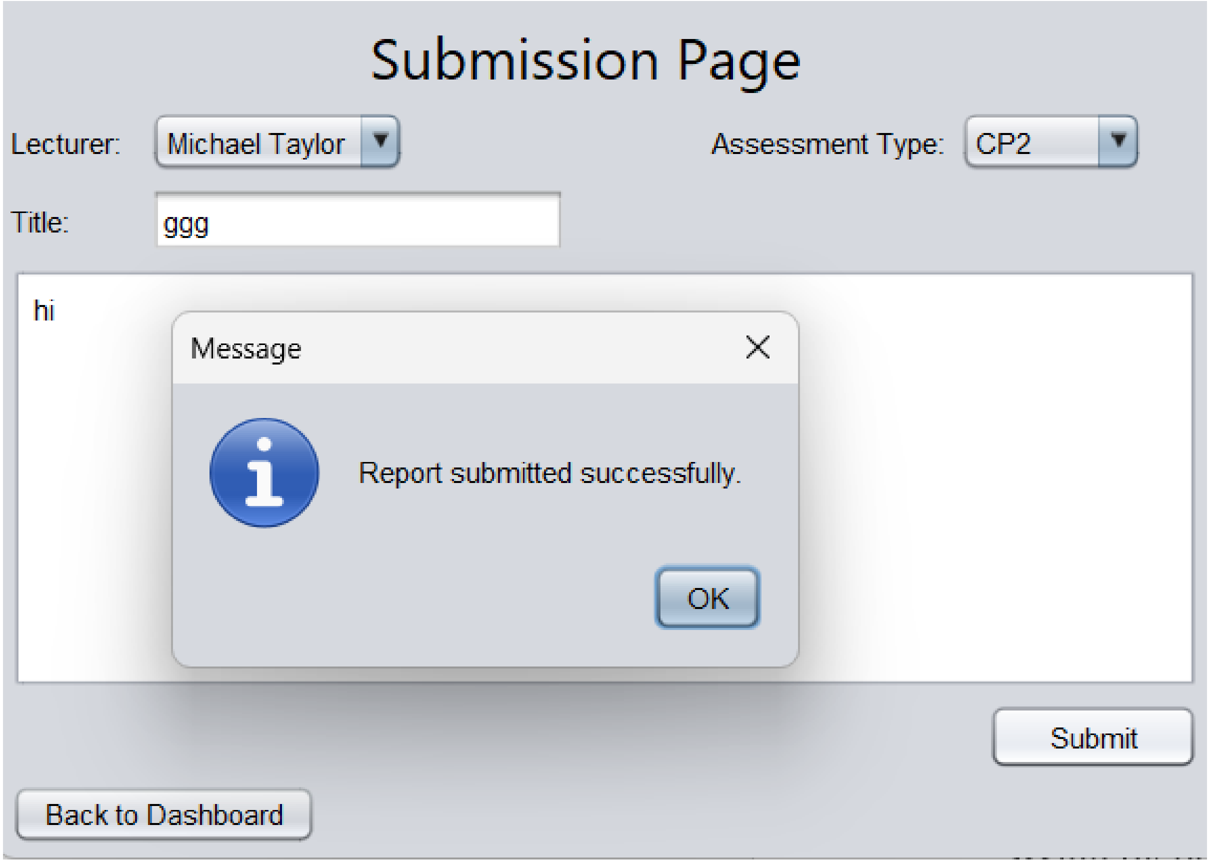


Title: ggg

Please ensure title or content is not empty before submission.

Figure 2.4.2.2: Validation of the content

If there is empty on the title and content column, it will cause an error to remind students that need to ensure that it will not empty just like Figure 2.4.2.2.



Submission Page

Lecturer: Michael Taylor

Assessment Type: CP2

Title: ggg

hi

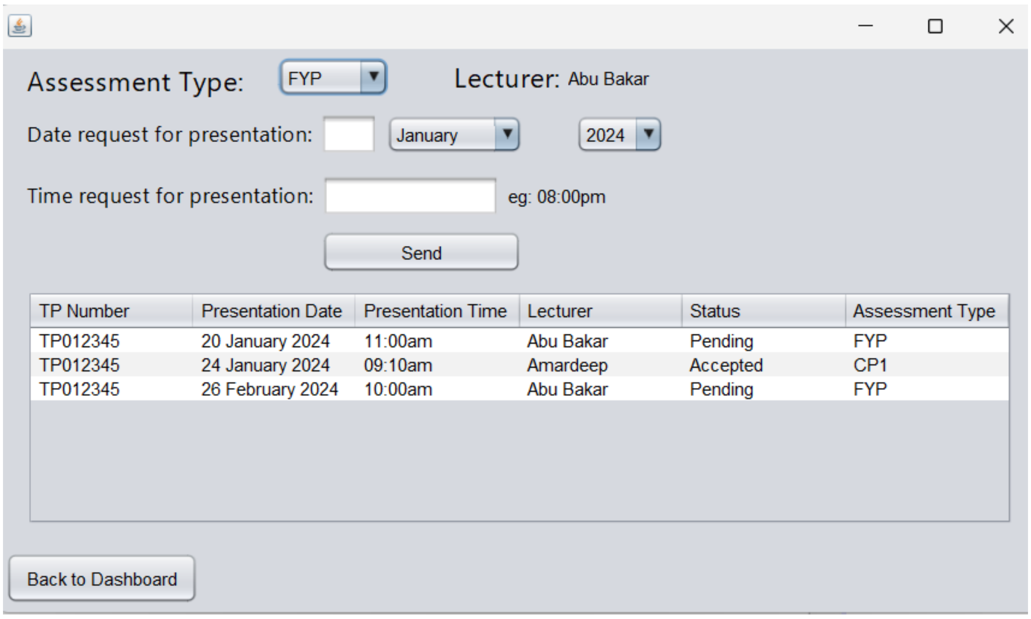
Message

Report submitted successfully.

Figure 2.4.2.3: Report submitted successfully

After writing the title and the content, click “Submit” button and the project will submit to the lecturer and show the message report submitted successfully like Figure 2.4.2.3. Click “Back to Dashboard” button will back to the student menu page.

2.4.3 Request Presentation Date



Assessment Type: FYP

Lecturer: Abu Bakar

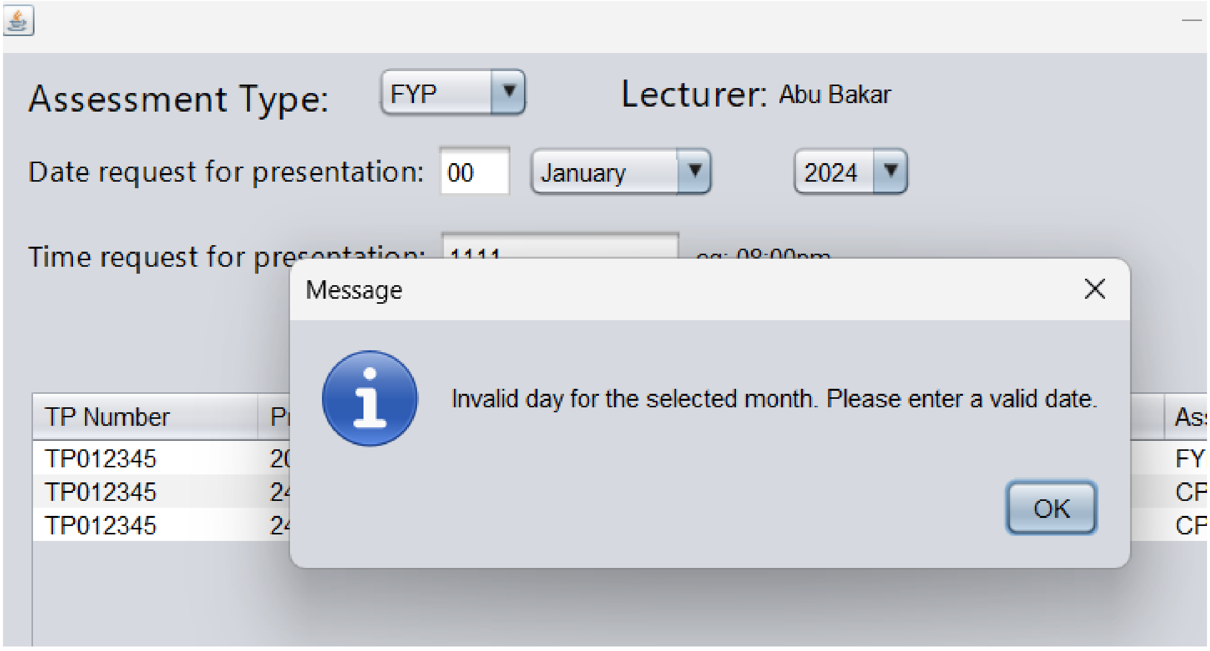
Date request for presentation: January 2024

Time request for presentation: eg: 08:00pm

TP Number	Presentation Date	Presentation Time	Lecturer	Status	Assessment Type
TP012345	20 January 2024	11:00am	Abu Bakar	Pending	FYP
TP012345	24 January 2024	09:10am	Amardeep	Accepted	CP1
TP012345	26 February 2024	10:00am	Abu Bakar	Pending	FYP

Figure 2.4.3.1: Request Presentation Date Page

Student can request presentation date to the lecturer to book a time for their presentation like figure 2.4.3.1. Choose the assessment type on the combo box and the lecturer in charge will be display. Fill in and choose the date student want to request for presentation. Then, fill in the time to request for presentation based on the format mentioned beside.



Assessment Type: FYP

Lecturer: Abu Bakar

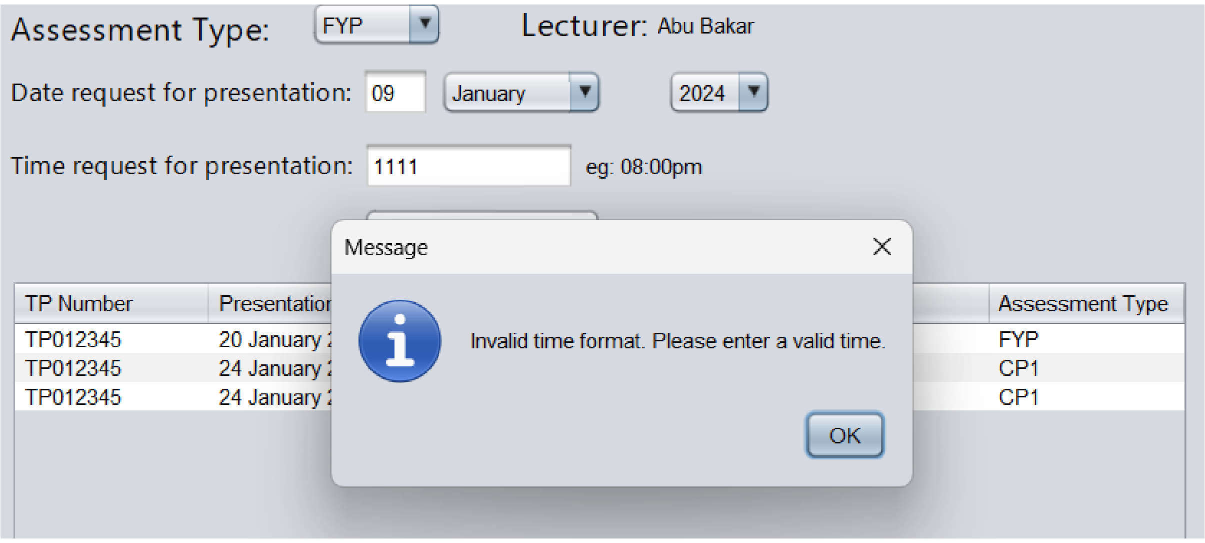
Date request for presentation: 00 January 2024

Time request for presentation: 1111 eg: 08:00pm

Message

Invalid day for the selected month. Please enter a valid date.

Figure 2.4.3.2: Invalid date



Assessment Type: FYP

Lecturer: Abu Bakar

Date request for presentation: 09 January 2024

Time request for presentation: 1111 eg: 08:00pm

Message

Invalid time format. Please enter a valid time.

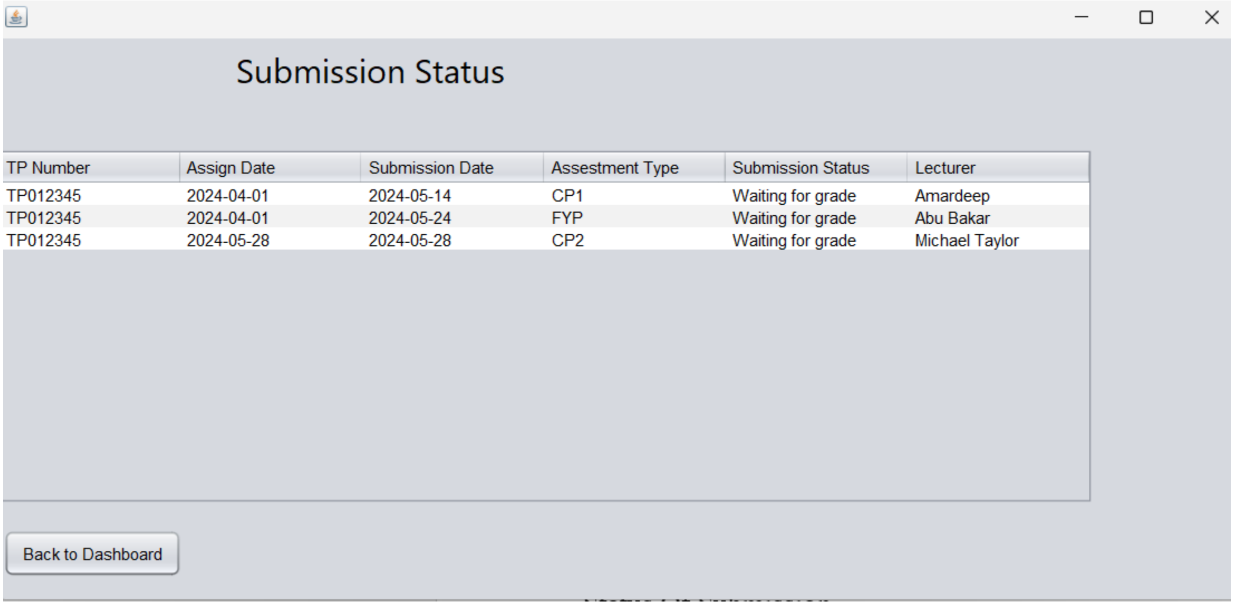
TP Number	Presentatior	Assessment Type
TP012345	20 January	FYP
TP012345	24 January	CP1
TP012345	24 January	CP1

Figure 2.4.3.3: Invalid time

There is a validation for date and time request for presentation shows on figure 2.4.3.2 and figure 2.4.3.3. When student fill in an invalid date and time request, it will cause an error to remind students that was invalid of date or time.

After filling in the valid date and time, then click “Send” button, it will send the presentation date and time to the lecturer who in charge. The presentation details will show in the table. When the presentation request doesn’t have respond by lecturer, the status will be “Pending”, so student need to for the confirmation from lecturer. If lecturer had accepted the request, the status will change to “Accepted” and if lecturer rejected the request, it will change to “Rejected”.

2.4.4 Status Of Submission



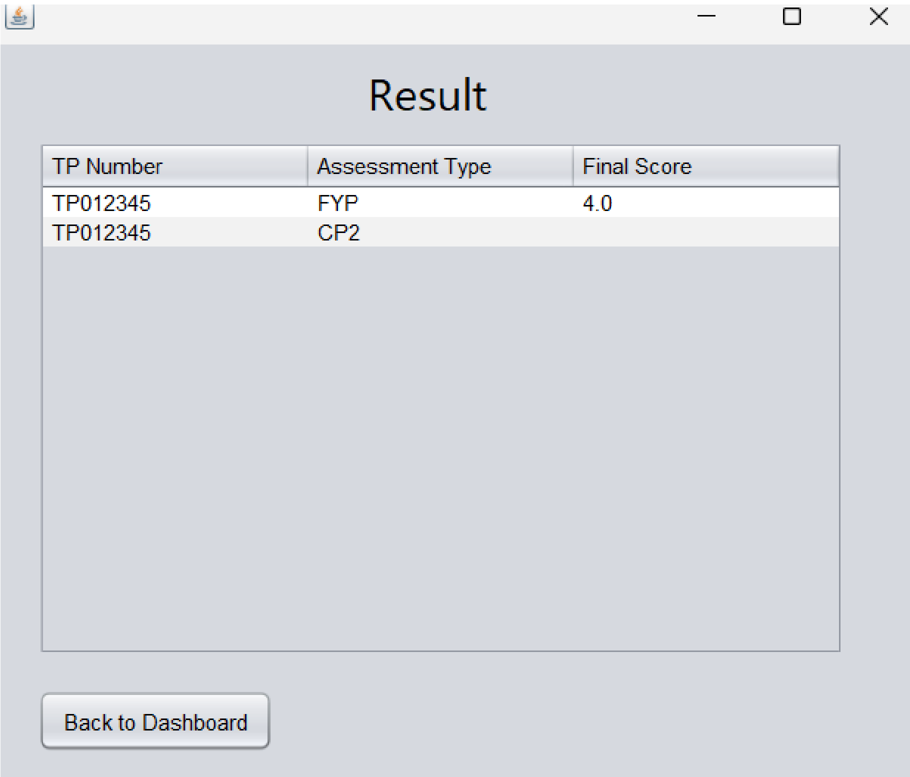
Submission Status

TP Number	Assign Date	Submission Date	Assestment Type	Submission Status	Lecturer
TP012345	2024-04-01	2024-05-14	CP1	Waiting for grade	Amardeep
TP012345	2024-04-01	2024-05-24	FYP	Waiting for grade	Abu Bakar
TP012345	2024-05-28	2024-05-28	CP2	Waiting for grade	Michael Taylor

Figure 2.4.4.1: Status of Submission

Student can check the submission status through clicking the “Status Of Submission” on Student Menu Page and it will bring students to Status Of Submission page, the page is like figure 2.4.4.1. In this page, students can check the submission status of their project, they can see the assign date, submission date, assessment type, submission status and the project had submitted to which lecturer. It can help students to check the status of submission of the project.

2.4.5 View Result



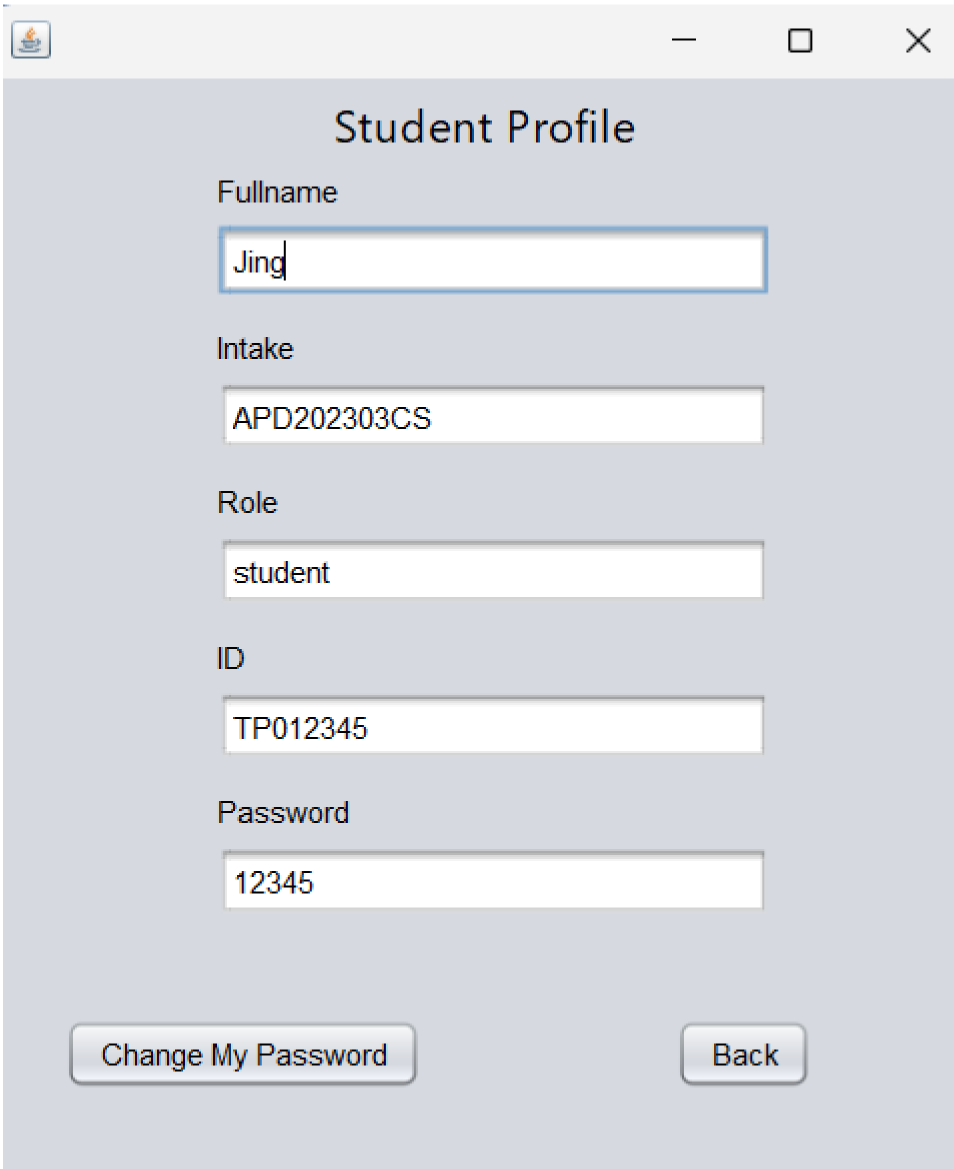
Result

TP Number	Assessment Type	Final Score
TP012345	FYP	4.0
TP012345	CP2	

Figure 2.4.5.1 Result page

Students can check their results when they click on the button "View Result" on the Student Menu Page, then it will go to the Result Page like figure 2.4.5.1. When the lecturer finish marking, the result will be come out at here. The table had displayed the assessment type that student had done and the final score.

2.4.6 View Profile



Student Profile

Fullname: Jing

Intake: APD202303CS

Role: student

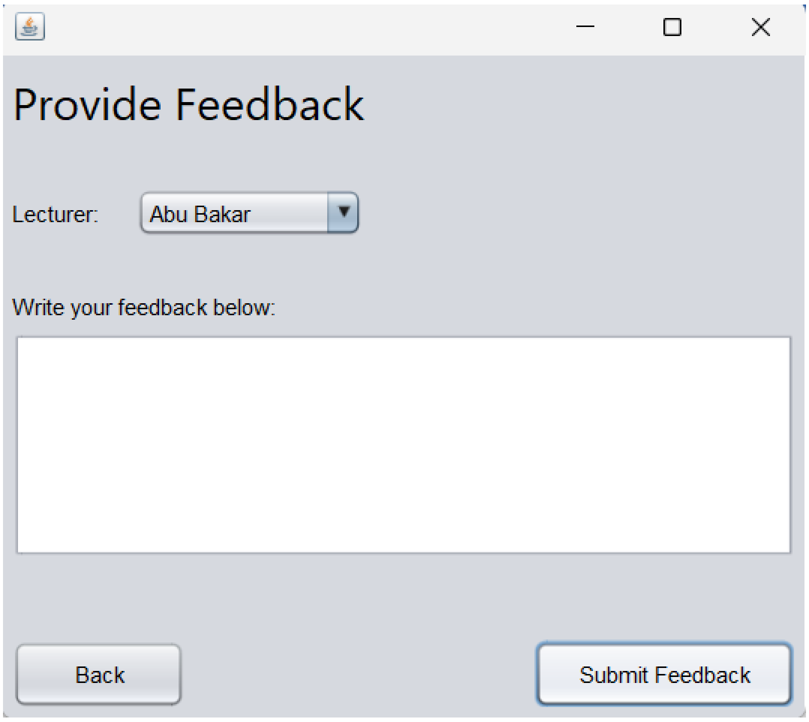
ID: TP012345

Password: 12345

Figure 2.4.6.1 View Profile Page

Student can click the button “View Profile” on the student menu and go to the View Profile Page show on figure 2.4.6.1. Student can view their profile like their full name, intake, role, ID number and password. If student want to change their password, they can click the button “Change My Password”. It will bring students to other page call “Change Password”. The page had mentioned earlier.

2.4.7 Provide Feedback



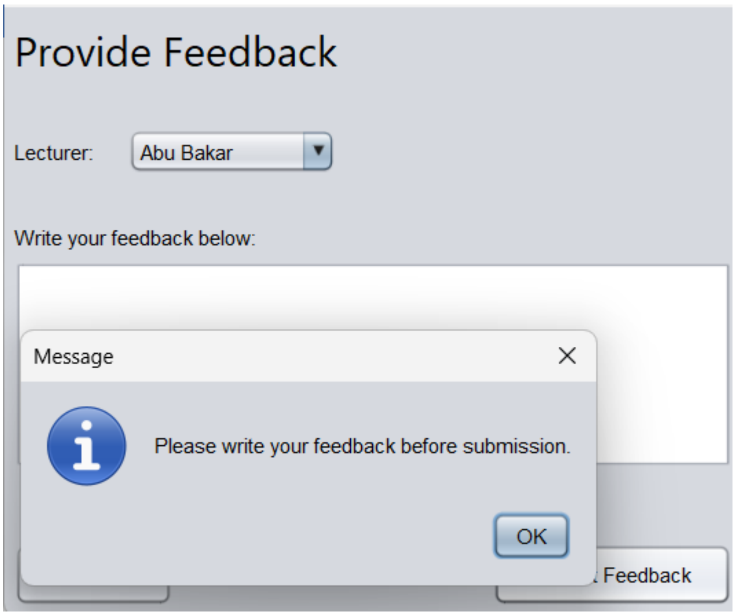
Provide Feedback

Lecturer: Abu Bakar

Write your feedback below:

Figure 2.4.7.1 Provide Feedback Page

When students want to give some feedback to the lecturer, they can choose the button "Provide Feedback" and go to Provide Feedback Page that show in figure 2.4.7.1. Students can choose which lecturer they want to provide feedback.



Provide Feedback

Lecturer: Abu Bakar

Write your feedback below:

Message

Please write your feedback before submission.

Figure 2.4.7.2 Validation of feedback

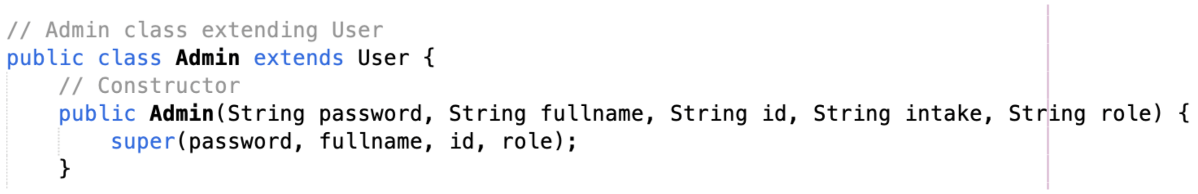
The validation had shown on figure 2.4.7.2. If there are empty on the text area, it will occur an error. After providing feedback, click the "Submit Feedback" button and the feedback will submit to lecturer anonymously.

3.0 Object-Oriented Concepts Incorporated

3.1 Inheritance

Inheritance is a fundamental concept in object-oriented programming where a new class (subclass or derived class) inherits properties and behaviours from an existing class (superclass or base class). In this case, “Admin” class, “Lecturer class”, “Project Manager” class and “Student” class are all child class that extend the “User” class which is the parent class, inheriting its properties and methods. This avoids code duplication and promotes code reusability. Since each of these classes is a specialized type of User, it makes sense to inherit from the User class and add or override functionalities specific to each class.

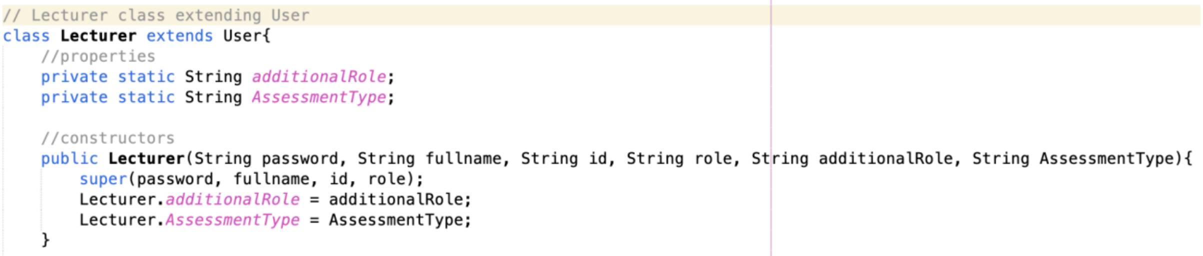
The “Admin” class, for example, not only inherits basic user information such as full name, id, password, and role and the methods of getter and setters from the “User” class but also adds administrative capabilities functions like adding new users, removing users, and managing roles.



```
// Admin class extending User
public class Admin extends User {
    // Constructor
    public Admin(String password, String fullname, String id, String intake, String role) {
        super(password, fullname, id, role);
    }
```

Figure 3.1.1: Admin class extending User

Similarly, the “Lecturer” class inherits common “User” properties while includes additional attributes such as additional Role and assessment Type to handle lecturer-specific details.



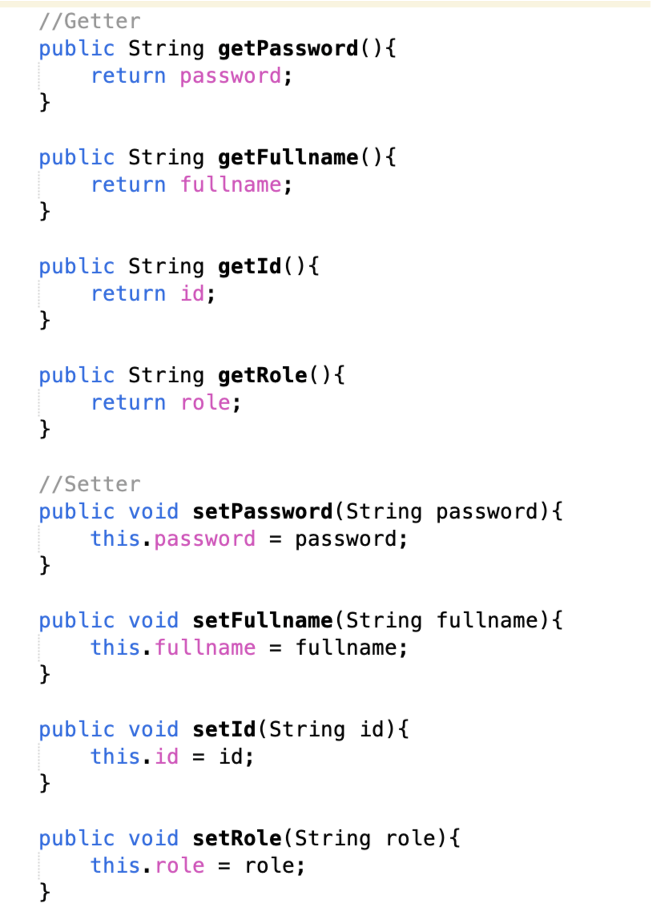
```
// Lecturer class extending User
class Lecturer extends User{
    //properties
    private static String additionalRole;
    private static String AssessmentType;

    //constructors
    public Lecturer(String password, String fullname, String id, String role, String additionalRole, String AssessmentType){
        super(password, fullname, id, role);
        Lecturer.additionalRole = additionalRole;
        Lecturer.AssessmentType = AssessmentType;
    }
```

Figure 3.1.2: Lecturer class extending User

The “Project Manager” class, while inheriting from “User”, can include functionalities and properties relevant to managing projects, and the “Student” class extends “User” to include student-specific details like intake.

3.2 Encapsulation

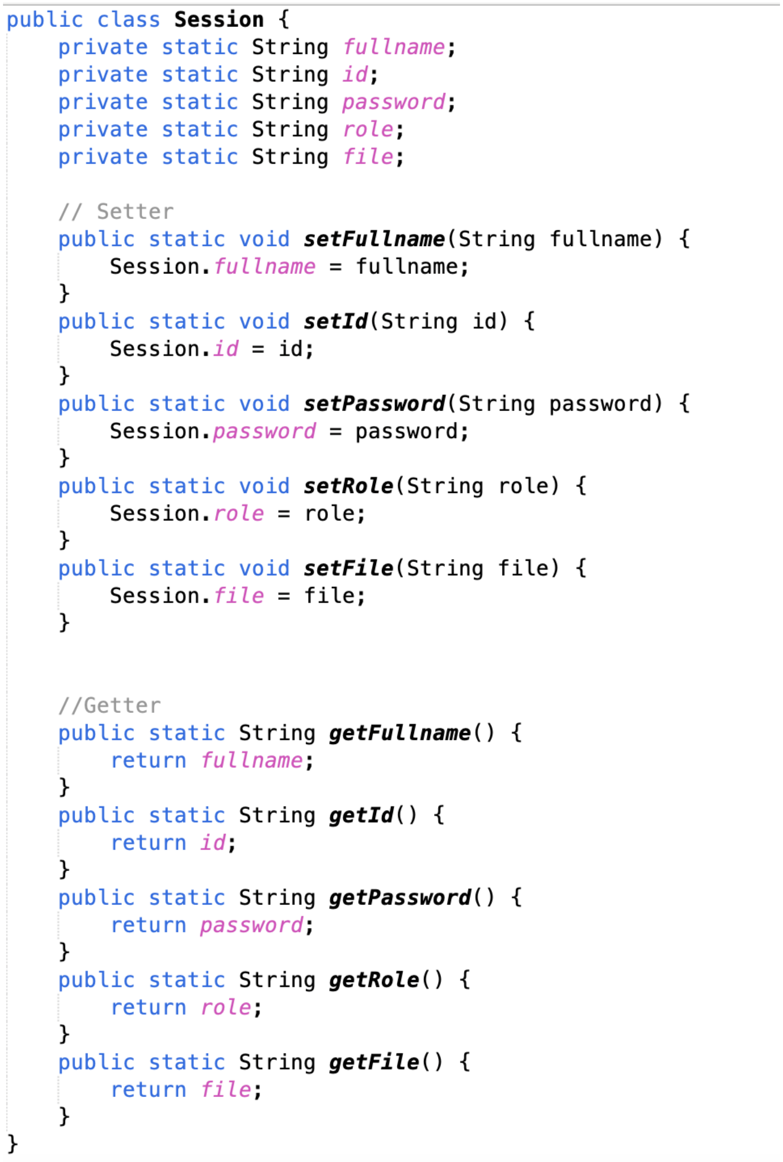
Encapsulation is a principle that bundle of data (attributes) and methods (behaviours) that operate on the data into a single class. In other words, encapsulation is a practice that keep fields within a class private, then providing access to those fields via public methods, usually getter and setter. This hides the internal state of objects from external access, preventing unauthorized manipulation and ensuring data integrity. In the provided solution, attributes such as password, full name, id, and role in the “User” class are declared as protected, restricting direct access from outside the class. Furthermore, it enables code reusability by encapsulating common functionalities within classes, which can be reused across different parts of the system. For instance, the “User” class encapsulates user-related attributes and methods, promoting reuse in subclasses such as Admin, Student, and Lecturer.



```java
//Getter
public String getPassword(){
    return password;
}

public String getFullname(){
    return fullname;
}

public String getId(){
    return id;
}

public String getRole(){
    return role;
}

//Setter
public void setPassword(String password){
    this.password = password;
}

public void setFullname(String fullname){
    this.fullname = fullname;
}

public void setId(String id){
    this.id = id;
}

public void setRole(String role){
    this.role = role;
}
```

Figure 3.2.1: Java code for getter and setter in “User” class

Figure 3.2.1 shows the code of getter and setter in the “User” class in order to achieve encapsulation. These methods allow encapsulated attributes to be accessed and modified.



```java
public class Session {
    private static String fullname;
    private static String id;
    private static String password;
    private static String role;
    private static String file;

    // Setter
    public static void setFullname(String fullname) {
        Session.fullname = fullname;
    }
    public static void setId(String id) {
        Session.id = id;
    }
    public static void setPassword(String password) {
        Session.password = password;
    }
    public static void setRole(String role) {
        Session.role = role;
    }
    public static void setFile(String file) {
        Session.file = file;
    }


    //Getter
    public static String getFullname() {
        return fullname;
    }
    public static String getId() {
        return id;
    }
    public static String getPassword() {
        return password;
    }
    public static String getRole() {
        return role;
    }
    public static String getFile() {
        return file;
    }
}
```

Figure 3.2.2: Java code of "Session" class

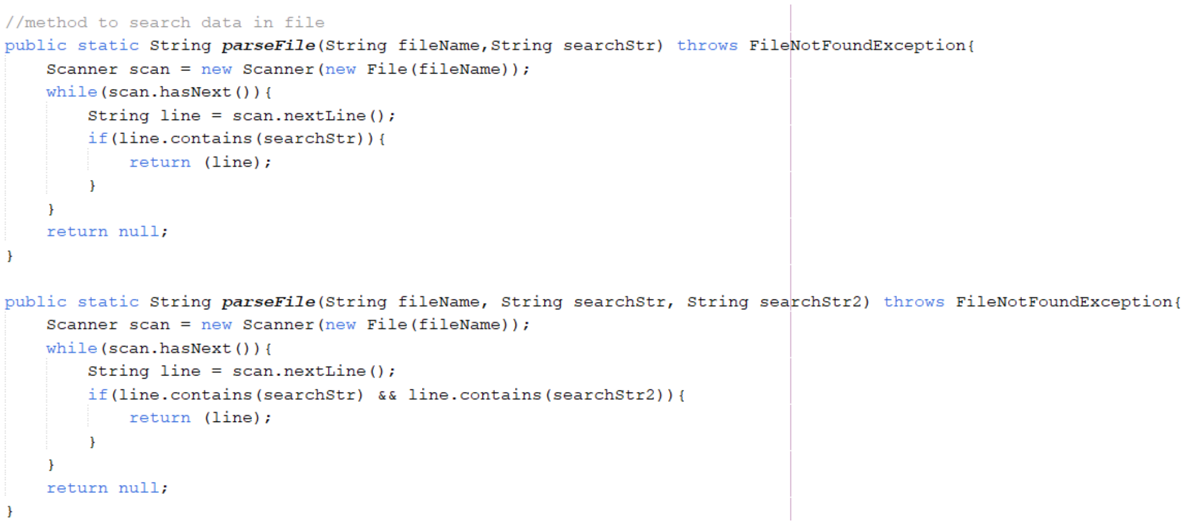
Additionally, a class called "Session" is also implemented that stores the current user's data that successfully logged in to the system. This class provides setter methods for setting session-related data (full name, id, password, role, file) after the user logged in. These data will remain unchanged throughout the whole system usage unless change password function is called which will uses the password setter to set the new password. User can access to these data to view their profile with the getter methods.

3.3 Polymorphism

Polymorphism is an OOP concept where one name can have many forms (Subramaniam.C.Valoo, 2022). It means to process objects differently based on their data type. There are 2 types of polymorphism which are commonly mentioned, which are static polymorphism which also known as compile time polymorphism and dynamic polymorphism which also known as runtime polymorphism. Both types of polymorphisms are implemented in our project.

Static polymorphism (Compile Time Polymorphism)

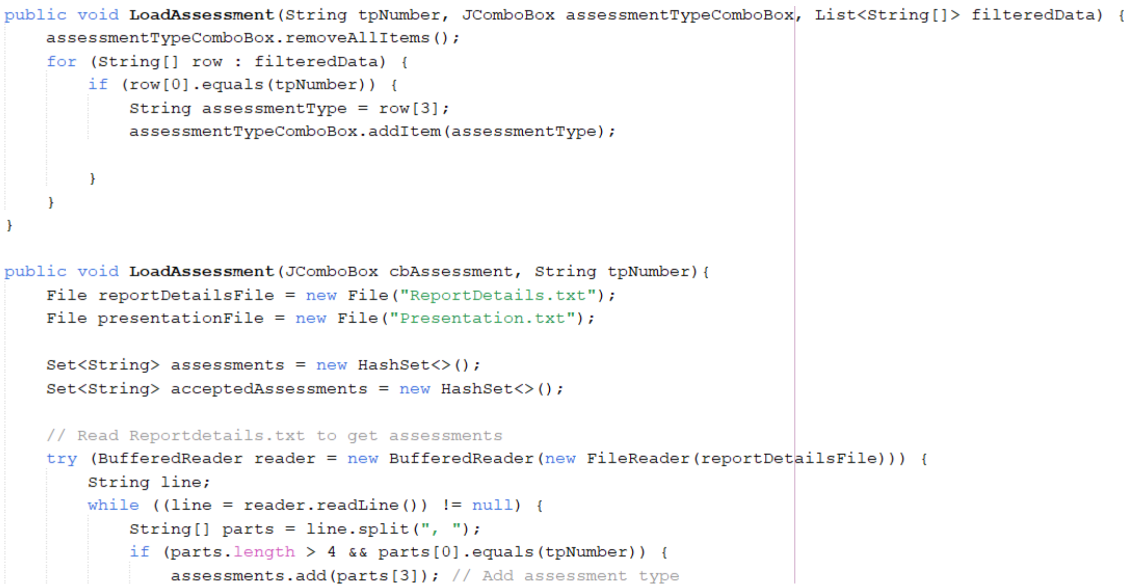
Static polymorphism allows us to implement multiple methods within a same class that use the same name but different set of parameters and this is called **method overloading**. The parameter sets can either have different number of parameters **or** types of parameters are different **or** parameters arranged in different order. Due to the different sets of parameters, that allows the compiler to identify which method is called as each method has a different signature.



```java
//method to search data in file
public static String parseFile(String fileName,String searchStr) throws FileNotFoundException{
    Scanner scan = new Scanner(new File(fileName));
    while(scan.hasNext()){
        String line = scan.nextLine();
        if(line.contains(searchStr)){
            return (line);
        }
    }
    return null;
}

public static String parseFile(String fileName, String searchStr, String searchStr2) throws FileNotFoundException{
    Scanner scan = new Scanner(new File(fileName));
    while(scan.hasNext()){
        String line = scan.nextLine();
        if(line.contains(searchStr) && line.contains(searchStr2)){
            return (line);
        }
    }
    return null;
}
```

Figure 3.3.1: Java code of "Project Manager" class

Based on Figure 3.3.1, the 2 methods demonstrate method overloading. They share the same method name "parseFile" and have different parameter lists. One takes 2 parameters and the other takes 3 parameters. In this case, the 2 methods read and retrieve data from file in different way by filtering the data from the files with different conditions.



```
public void LoadAssessment(String tpNumber, JComboBox assessmentTypeComboBox, List<String[]> filteredData) {
    assessmentTypeComboBox.removeAllItems();
    for (String[] row : filteredData) {
        if (row[0].equals(tpNumber)) {
            String assessmentType = row[3];
            assessmentTypeComboBox.addItem(assessmentType);

        }
    }
}

public void LoadAssessment(JComboBox cbAssessment, String tpNumber){
    File reportDetailsFile = new File("ReportDetails.txt");
    File presentationFile = new File("Presentation.txt");

    Set<String> assessments = new HashSet<>();
    Set<String> acceptedAssessments = new HashSet<>();

    // Read Reportdetails.txt to get assessments
    try (BufferedReader reader = new BufferedReader(new FileReader(reportDetailsFile))) {
        String line;
        while ((line = reader.readLine()) != null) {
            String[] parts = line.split(", ");
            if (parts.length > 4 && parts[0].equals(tpNumber)) {
                assessments.add(parts[3]); // Add assessment type
```

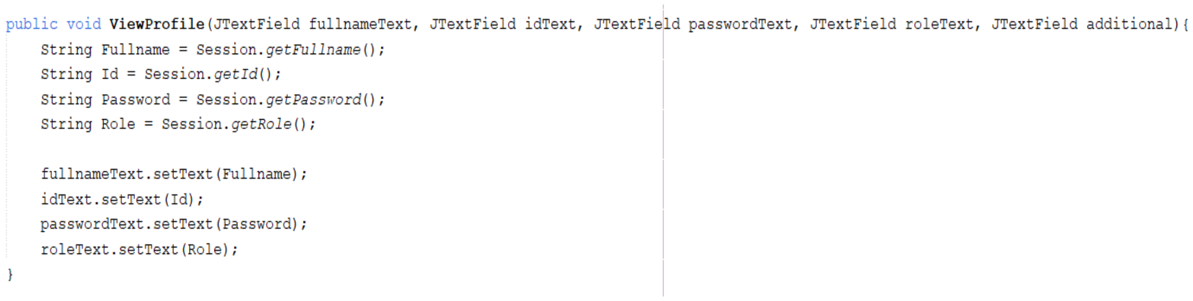
Figure 3.3.2: Java code of “Presentation” class

Based on Figure 3.3.2, this is another example of method overloading in our Java code. Both methods share the same name but with different parameters. In this case, both “loadAssessment” method populate the “JComboBox” based on different file sources and conditions.

Based on the 2 examples of method overloading implemented in our project, the overloaded methods provide different but very similar functionality. The method overloading approach makes the code more flexible as the method’s behaviour adapts to provided parameters.

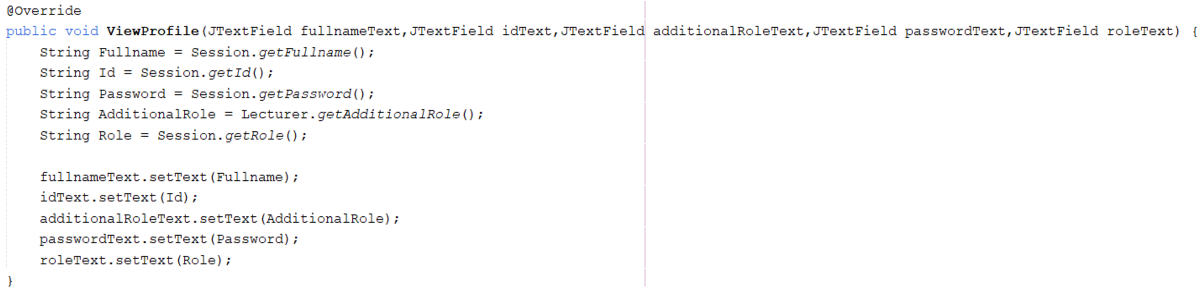
Dynamic polymorphism (Runtime Polymorphism)

In dynamic polymorphism, the compiler is not allowed to determine the executed method. The compiler has to do that at runtime. In an inheritance hierarchy, a method in the superclass can be overridden by the subclass. This enables the subclass to customize or completely replace the behaviour of the method in superclass. Similar to static polymorphism, the methods, implemented by the superclass and subclass share the same name. However, the parameters of the methods must be the same and provide different functionality. This situation is called **method overriding**.



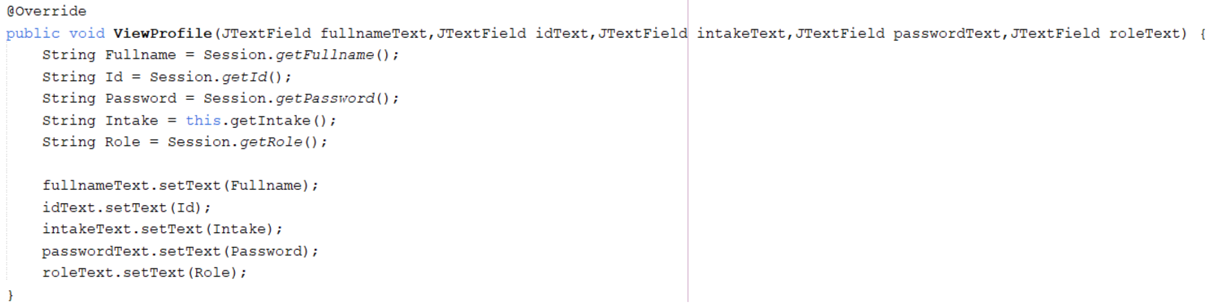
```
public void ViewProfile(JTextField fullnameText, JTextField idText, JTextField passwordText, JTextField roleText, JTextField additional){
    String Fullname = Session.getFullname();
    String Id = Session.getId();
    String Password = Session.getPassword();
    String Role = Session.getRole();

    fullnameText.setText(Fullname);
    idText.setText(Id);
    passwordText.setText(Password);
    roleText.setText(Role);
}
```

Figure 3.3.3: Java code of “User” class



```
@Override
public void ViewProfile(JTextField fullnameText,JTextField idText,JTextField additionalRoleText,JTextField passwordText,JTextField roleText) {
    String Fullname = Session.getFullname();
    String Id = Session.getId();
    String Password = Session.getPassword();
    String AdditionalRole = Lecturer.getAdditionalRole();
    String Role = Session.getRole();

    fullnameText.setText(Fullname);
    idText.setText(Id);
    additionalRoleText.setText(AdditionalRole);
    passwordText.setText(Password);
    roleText.setText(Role);
}
```

Figure 3.3.4: Java code of “Lecturer” class

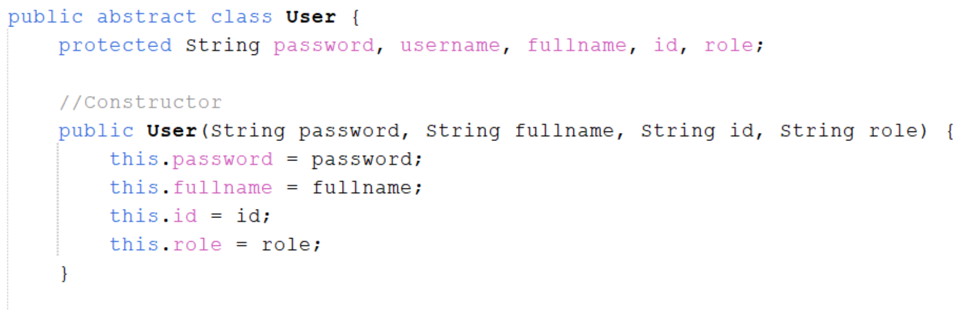


```
@Override
public void ViewProfile(JTextField fullnameText,JTextField idText,JTextField intakeText,JTextField passwordText,JTextField roleText) {
    String Fullname = Session.getFullname();
    String Id = Session.getId();
    String Password = Session.getPassword();
    String Intake = this.getIntake();
    String Role = Session.getRole();

    fullnameText.setText(Fullname);
    idText.setText(Id);
    intakeText.setText(Intake);
    passwordText.setText(Password);
    roleText.setText(Role);
}
```

Figure 3.3.5: Java code of “Student” class

Based on Figure 3.3.3, Figure 3.3.4 and Figure 3.3.5, method overriding happens as the “Lecturer” and “Student” classes override the “ViewProfile” method defined in the “User” class. The “@Override” annotation indicates that these methods are intended to override the superclass method.

3.4 Abstraction



```
public abstract class User {
    protected String password, username, fullname, id, role;

    //Constructor
    public User(String password, String fullname, String id, String role) {
        this.password = password;
        this.fullname = fullname;
        this.id = id;
        this.role = role;
    }
```

Figure 3.4.1: Abstract class “User”

Abstraction is a concept in Java that involves hiding complex implementation details and only showing the necessary features of an object. It is declared using the “abstract” keyword in Java and can contain both abstract and non-abstract methods. In the code, the User class is defined as abstract, which means it cannot be instantiated and serves as a blueprint for other classes.

The User class is abstract because it represents a generic user in the system. It defines properties and methods common to all types of users, but it does not implement any specific functionality that would allow it to be used on its own. While this class does not define any abstract methods, it serves as a base for subclasses, which will implement specific behaviours for different types of users, such as administrators, project managers, students, and lecturers.

4.0 Limitation and Conclusion

In conclusion, the Object-Oriented System developed for Academic Guidance Hub (AGH) effectively addresses the challenges faced by the institution such as managing student enrolment, project assignment, and evaluation processes. The program is accessible by four different role-based account, where each distinct role is able to perform respective functionalities. For the "Admin" role, typical functions such as new user registration, authenticate user log in and edit/remove user data are implemented. While for the "Lecturer" role, the functions include evaluate reports submitted by students, view presentation requests by students and confirm the presentation requests, view supervisee dashboard to see overall progress of students, view all the second marker acceptance and confirm second marker request if any, view assigned supervisees as assigned by project manager, view supervisee list to view the status of students' presentation requests, view presentation timetable and update the students' presentation status if presented. For the "Student" role, the function includes submit the details of the submitted project, request presentation date, check status of submission, view result, view profile and provide feedback. There are three main functions that the project manager can perform, one is to allocate student, which is used to assign assessment to the designated student, the other is to assign lecturer to assign supervisor and second marker positions to lecturer, and the last is to view status to view students' grades and reports.

However, it is important to acknowledge certain limitations and considerations too. One of the possible drawbacks for admin is that the system does not log admin activities, such as adding or removes user or updating user information. This lack of auditing makes it difficult to track changes and identify potential security breaches or misuse of admin privileges. Moreover, the system does not support heavy operations like adding or removing multiple users at once. For now, admin is required to perform these actions individually, which can be time-consuming when handling large user groups. By overall, firstly the system does not provide a comprehensive notification mechanism. This means lecturers might not receive timely updates about new submissions, presentation requests, or changes in supervisee status unless they manually check the system regularly. This can lead to delays in response times and affect the overall efficiency of the evaluation and supervision process. Additionally, the platform does not include a collaborative workspace for lecturers and students to interact in real-time. Such a feature could facilitate more dynamic supervision and feedback processes, enhancing the learning experience for students and making it easier for lecturers to manage their responsibilities. Lastly, the system lacks detailed analytics and reporting tools that could provide lecturers with insights into student performance and progress over time. This information could be valuable for identifying trends, addressing issues promptly, and improving the overall quality of guidance provided to students.

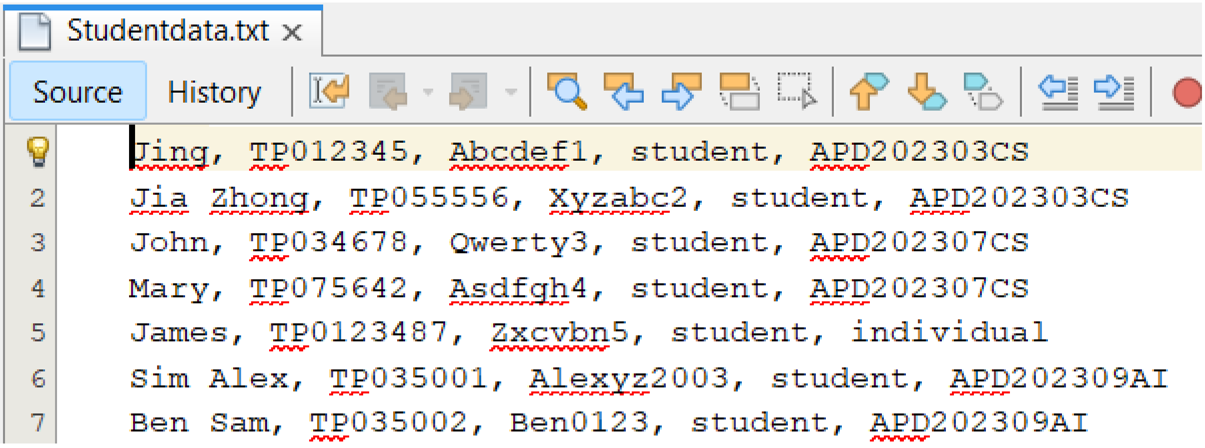
5.0 References

Revutska, N. (2023). *WHAT IS OBJECT-ORIENTED PROGRAMMING (OOP)? EXPLAINING FOUR MAJOR PRINCIPLES.* softserve. https://career.softserveinc.com/en-us/stories/what-is-object-oriented-programming-oop-explaining-four-major-principles

Stackify. (2024). *What Are OOP Concepts in Java? How They Work and More.* https://stackify.com/oops-concepts-in-java/

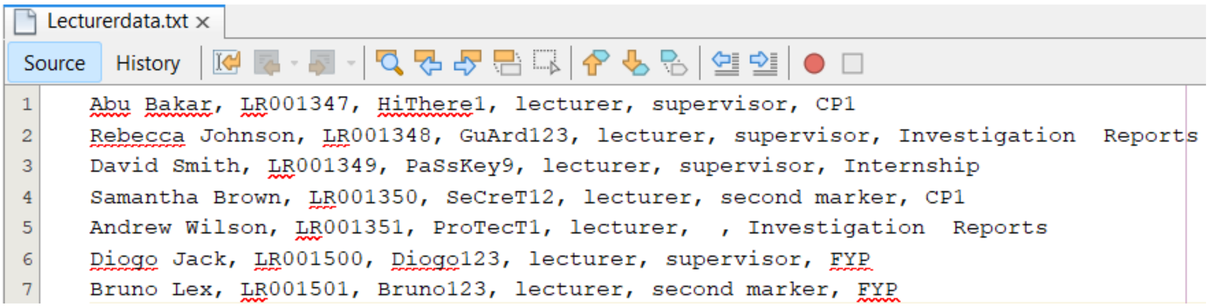
PREMMAURYA. (2019). *Polymorphism in Java - GeeksforGeeks*. GeeksforGeeks. https://www.geeksforgeeks.org/polymorphism-in-java/

6.0 Appendix



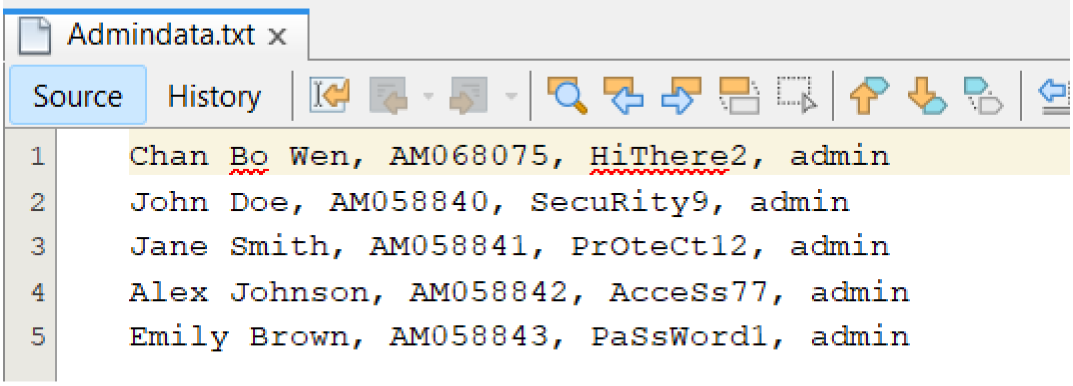
```
Jing, TP012345, Abcdef1, student, APD202303CS
Jia Zhong, TP055556, Xyzabc2, student, APD202303CS
John, TP034678, Qwerty3, student, APD202307CS
Mary, TP075642, Asdfgh4, student, APD202307CS
James, TP0123487, Zxcvbn5, student, individual
Sim Alex, TP035001, Alexyz2003, student, APD202309AI
Ben Sam, TP035002, Ben0123, student, APD202309AI
```

Figure 6.0.1: Studentdata.txt text file



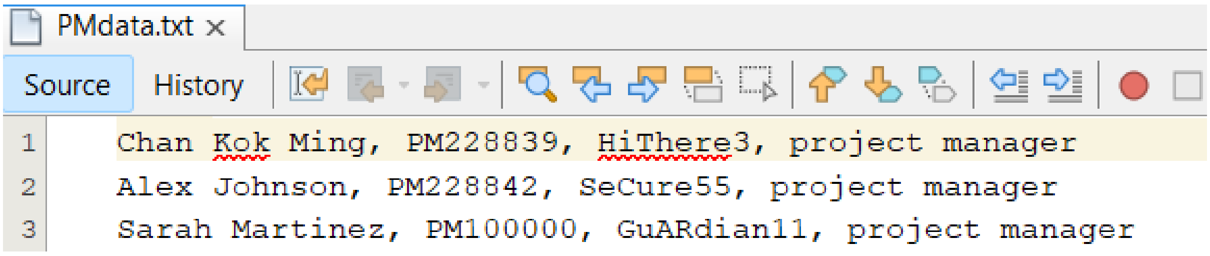
```
Abu Bakar, LR001347, HiThere1, lecturer, supervisor, CP1
Rebecca Johnson, LR001348, GuArd123, lecturer, supervisor, Investigation  Reports
David Smith, LR001349, PaSsKey9, lecturer, supervisor, Internship
Samantha Brown, LR001350, SeCreT12, lecturer, second marker, CP1
Andrew Wilson, LR001351, ProTecT1, lecturer,  , Investigation  Reports
Diogo Jack, LR001500, Diogo123, lecturer, supervisor, FYP
Bruno Lex, LR001501, Bruno123, lecturer, second marker, FYP
```

Figure 6.0.2: Lecturerdata.txt text file



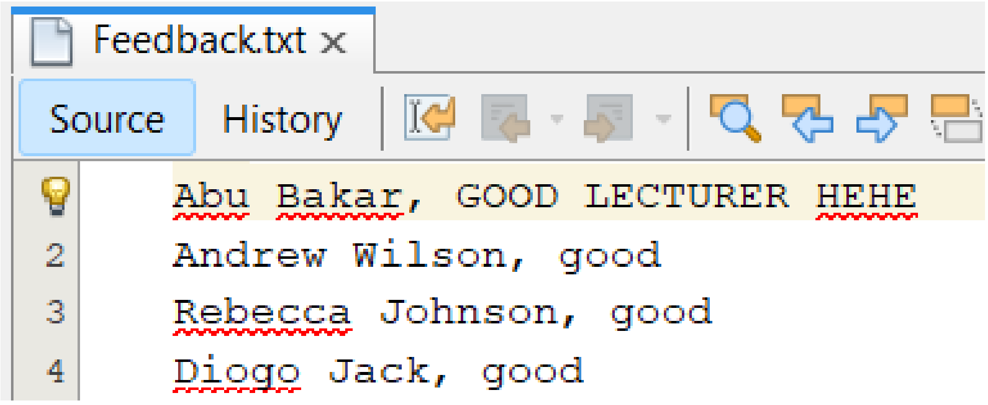
```
Chan Bo Wen, AM068075, HiThere2, admin
John Doe, AM058840, SecuRity9, admin
Jane Smith, AM058841, PrOteCt12, admin
Alex Johnson, AM058842, AcceSs77, admin
Emily Brown, AM058843, PaSsWord1, admin
```

Figure 6.0.3: Admindata.txt text file



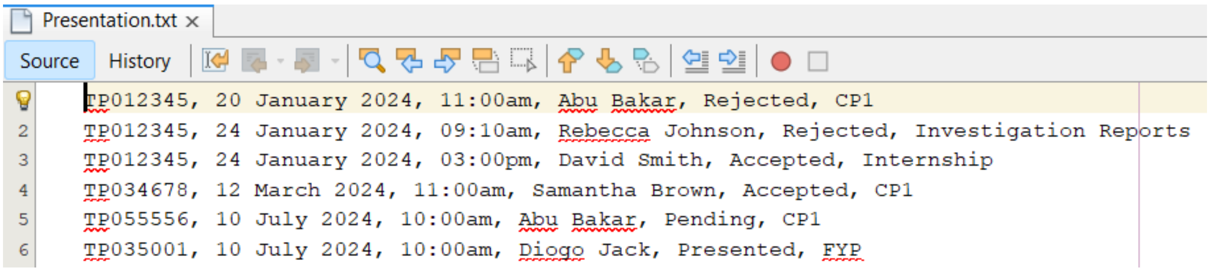
```
Chan Kok Ming, PM228839, HiThere3, project manager
Alex Johnson, PM228842, SeCure55, project manager
Sarah Martinez, PM100000, GuARdian11, project manager
```

Figure 6.0.4: PMdata.txt text file



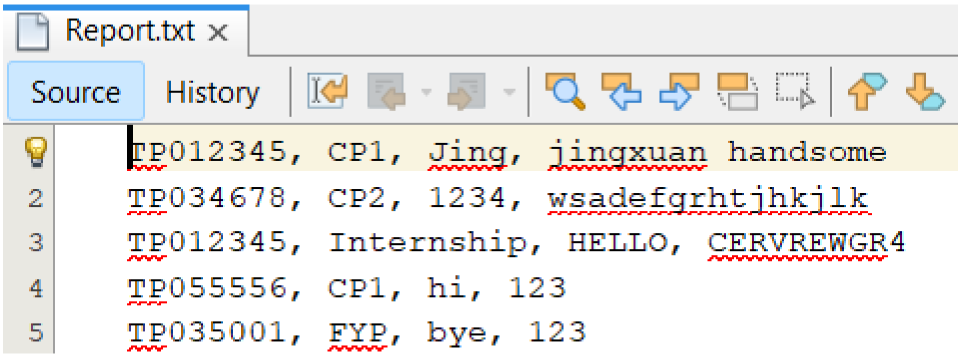
```
Abu Bakar, GOOD LECTURER HEHE
Andrew Wilson, good
Rebecca Johnson, good
Diogo Jack, good
```

Figure 6.0.5: Feedback.txt text file



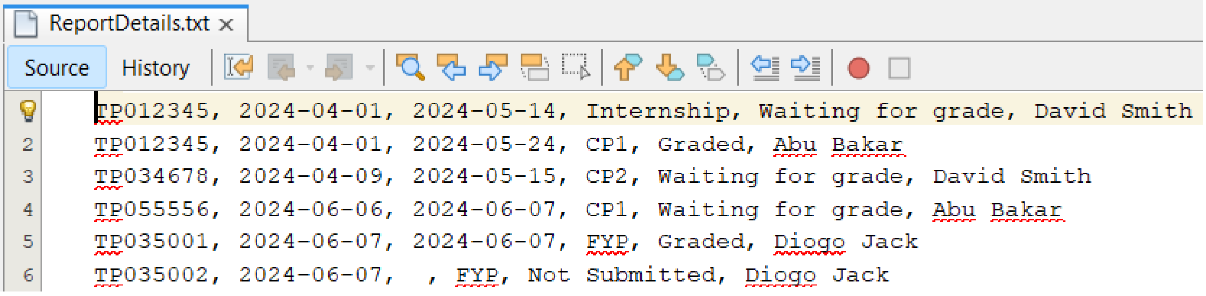
```
TP012345, 20 January 2024, 11:00am, Abu Bakar, Rejected, CP1
TP012345, 24 January 2024, 09:10am, Rebecca Johnson, Rejected, Investigation Reports
TP012345, 24 January 2024, 03:00pm, David Smith, Accepted, Internship
TP034678, 12 March 2024, 11:00am, Samantha Brown, Accepted, CP1
TP055556, 10 July 2024, 10:00am, Abu Bakar, Pending, CP1
TP035001, 10 July 2024, 10:00am, Diogo Jack, Presented, FYP
```

Figure 6.0.6: Presentation.txt text file



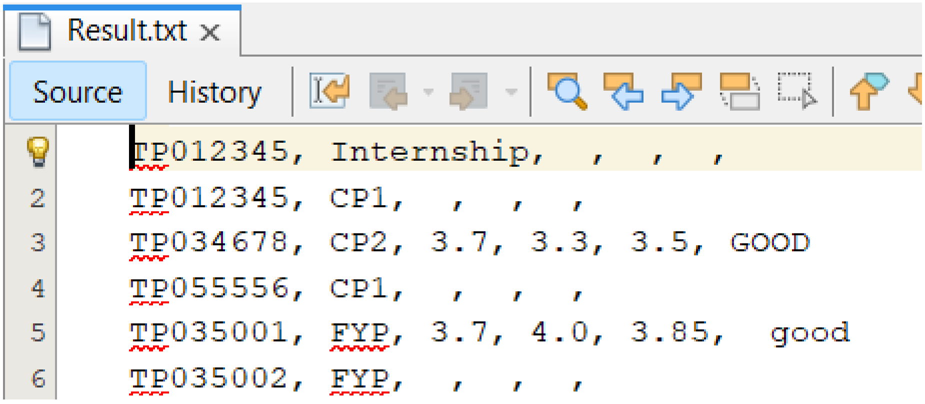
```
TP012345, CP1, Jing, jingxuan handsome
TP034678, CP2, 1234, wsadefgrhtjhkjlk
TP012345, Internship, HELLO, CERVREWGR4
TP055556, CP1, hi, 123
TP035001, FYP, bye, 123
```

Figure 6.0.7: Report.txt text file



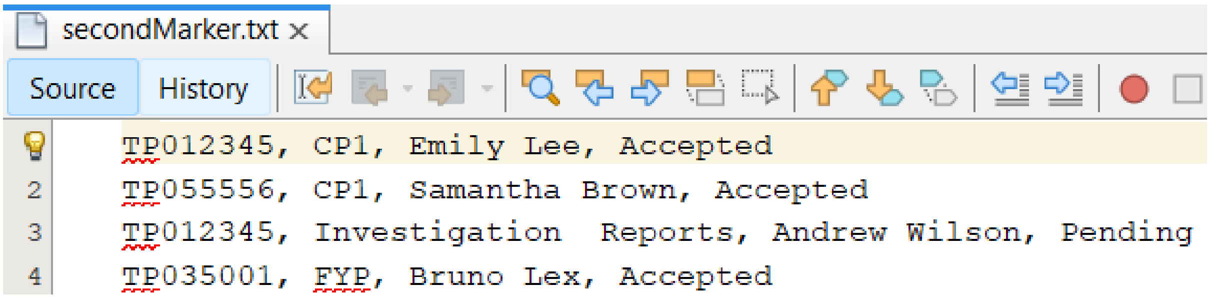
```
TP012345, 2024-04-01, 2024-05-14, Internship, Waiting for grade, David Smith
TP012345, 2024-04-01, 2024-05-24, CP1, Graded, Abu Bakar
TP034678, 2024-04-09, 2024-05-15, CP2, Waiting for grade, David Smith
TP055556, 2024-06-06, 2024-06-07, CP1, Waiting for grade, Abu Bakar
TP035001, 2024-06-07, 2024-06-07, FYP, Graded, Diogo Jack
TP035002, 2024-06-07,  , FYP, Not Submitted, Diogo Jack
```

Figure 6.0.8: ReportDetails.txt text file



```
TP012345, Internship,  ,  ,  ,
TP012345, CP1,  ,  ,  ,
TP034678, CP2, 3.7, 3.3, 3.5, GOOD
TP055556, CP1,  ,  ,  ,
TP035001, FYP, 3.7, 4.0, 3.85,  good
TP035002, FYP,  ,  ,  ,
```

Figure 6.0.9: Result.txt text file



```
TP012345, CP1, Emily Lee, Accepted
TP055556, CP1, Samantha Brown, Accepted
TP012345, Investigation  Reports, Andrew Wilson, Pending
TP035001, FYP, Bruno Lex, Accepted
```

Figure 6.0.10: secondMarker.txt text file