DESIGN
Software interface

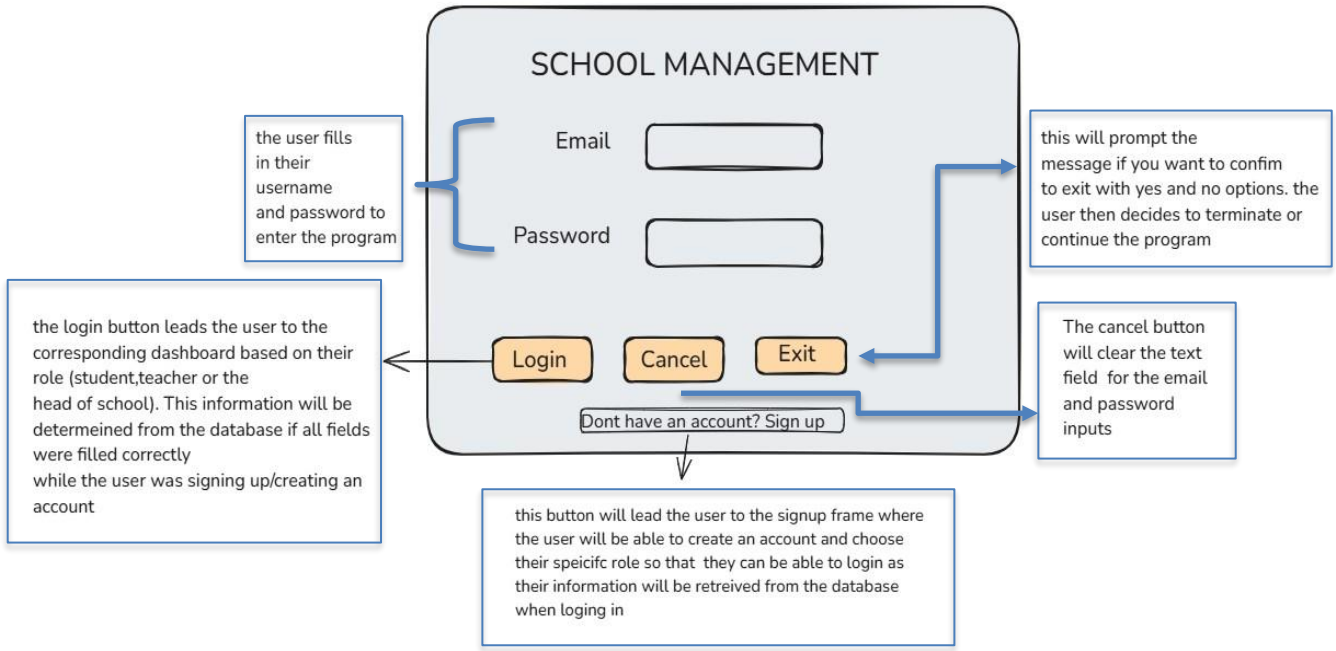


Figure 1; Login dashboard.

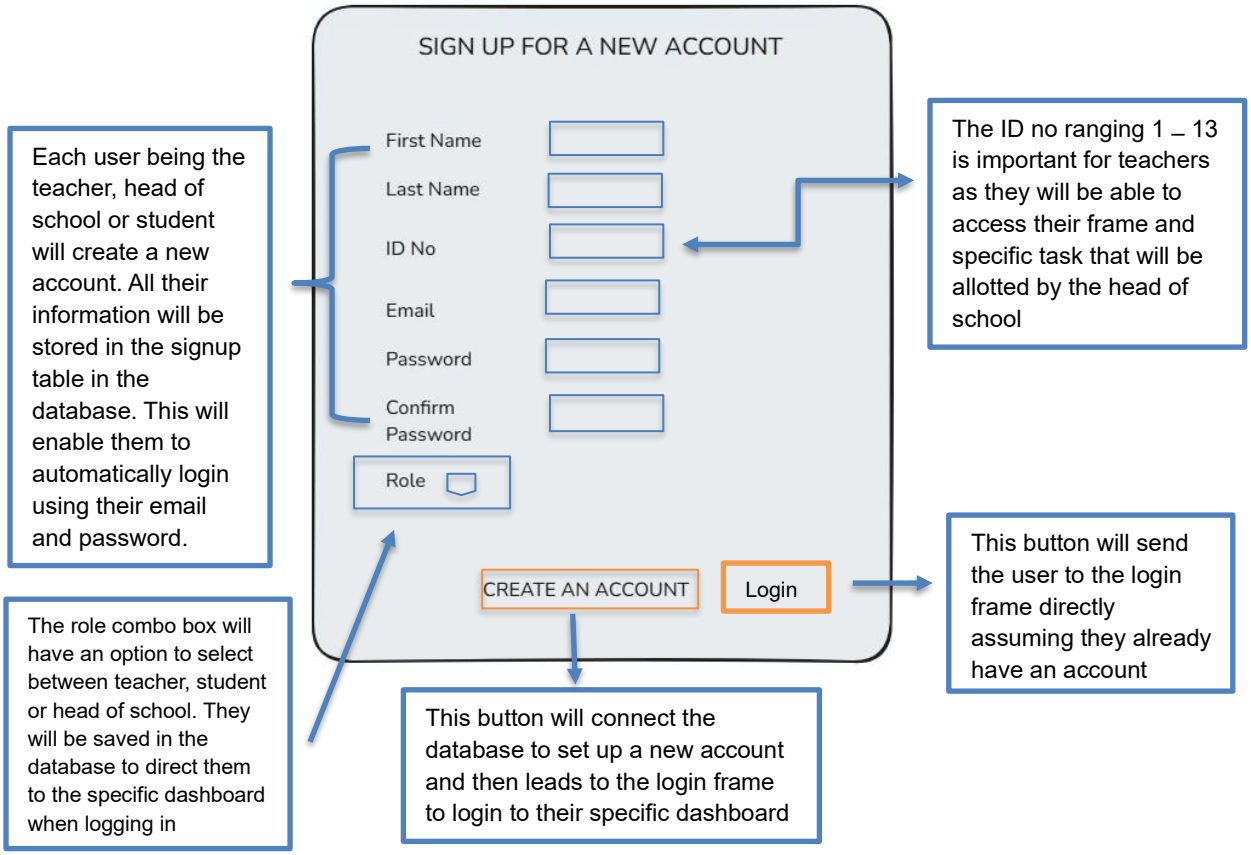


Figure 2; Sign up frame

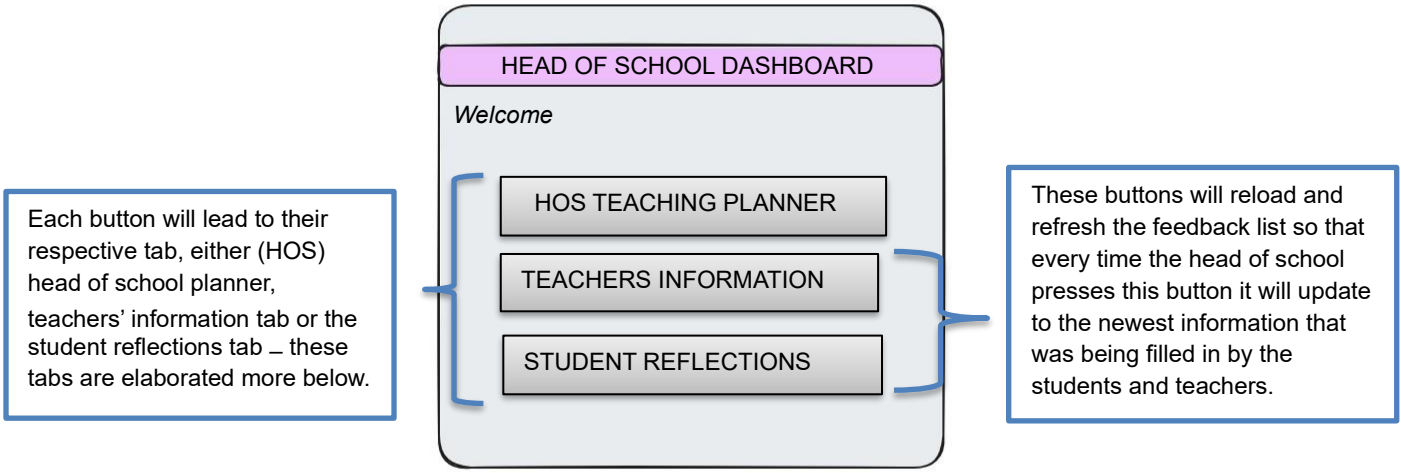


Figure 3; Head of School Dashboard

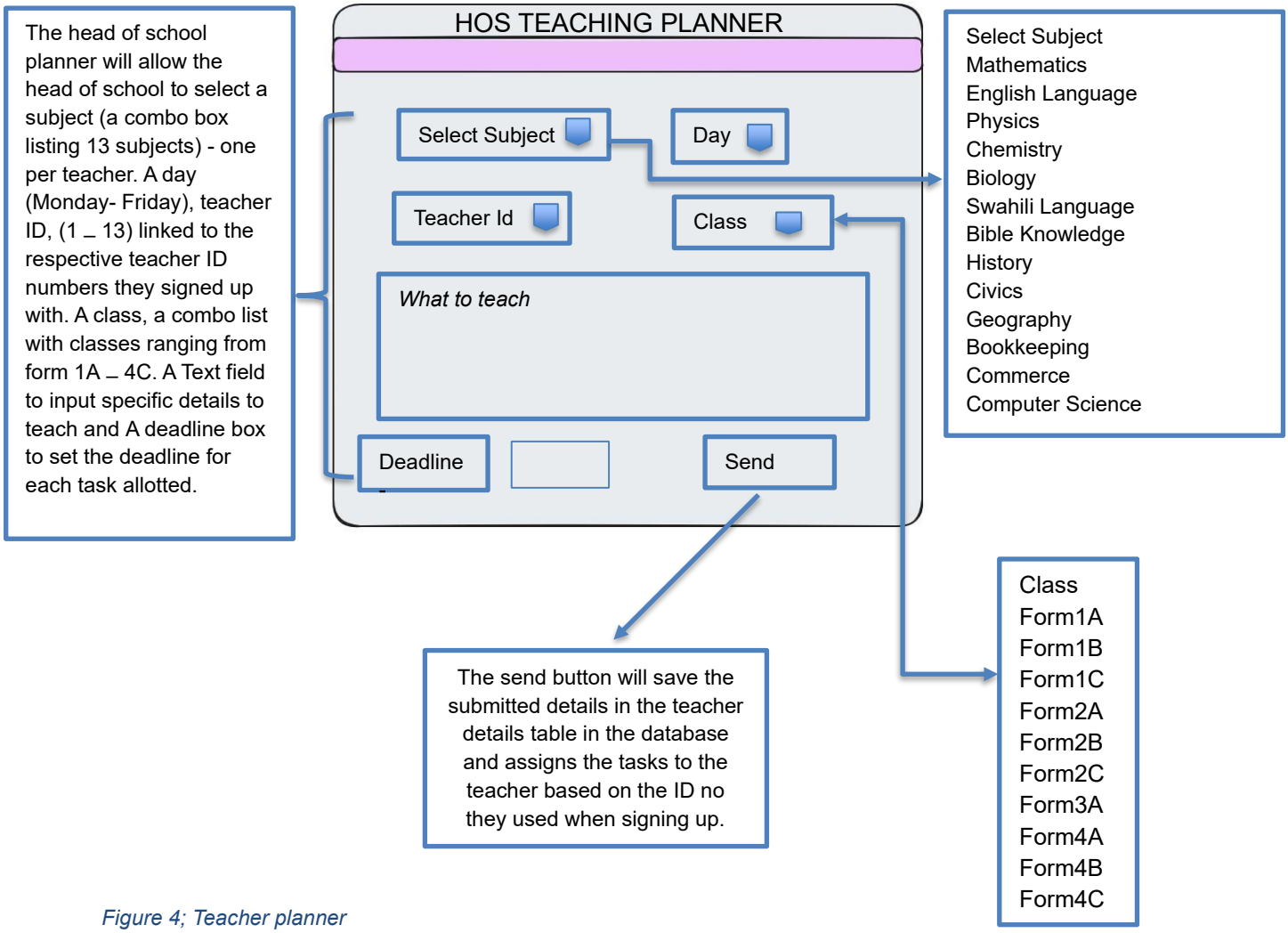


Figure 4; Teacher planner

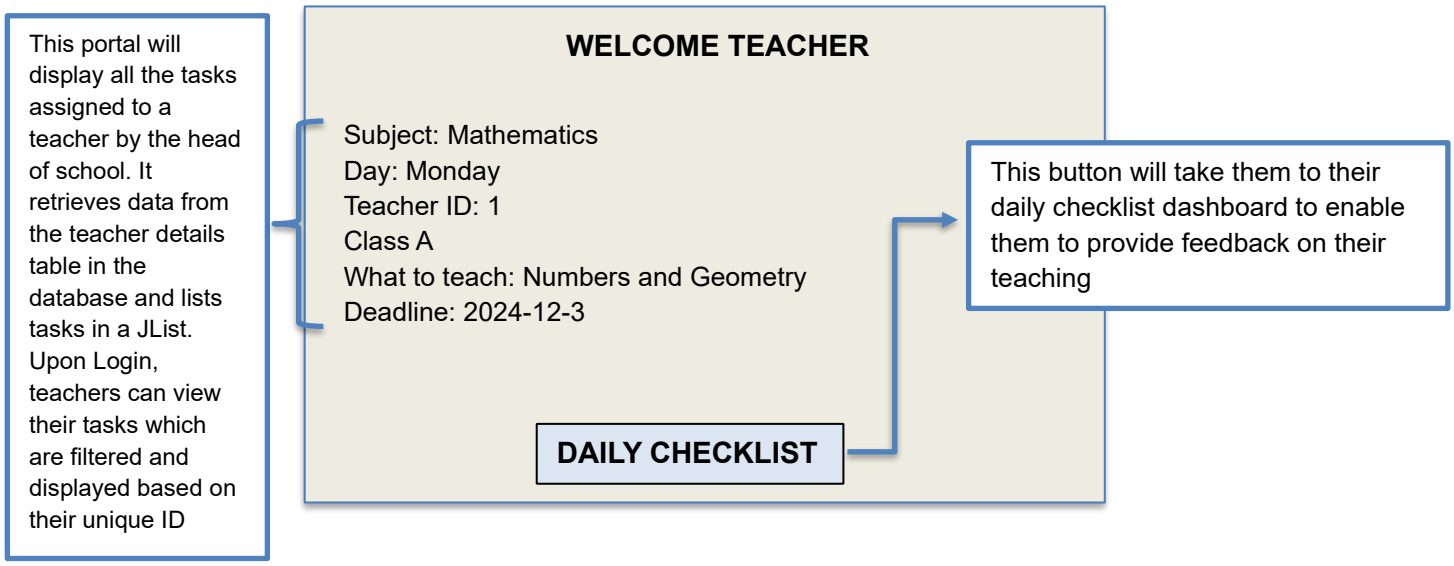


Figure 5; Teacher Dashboard

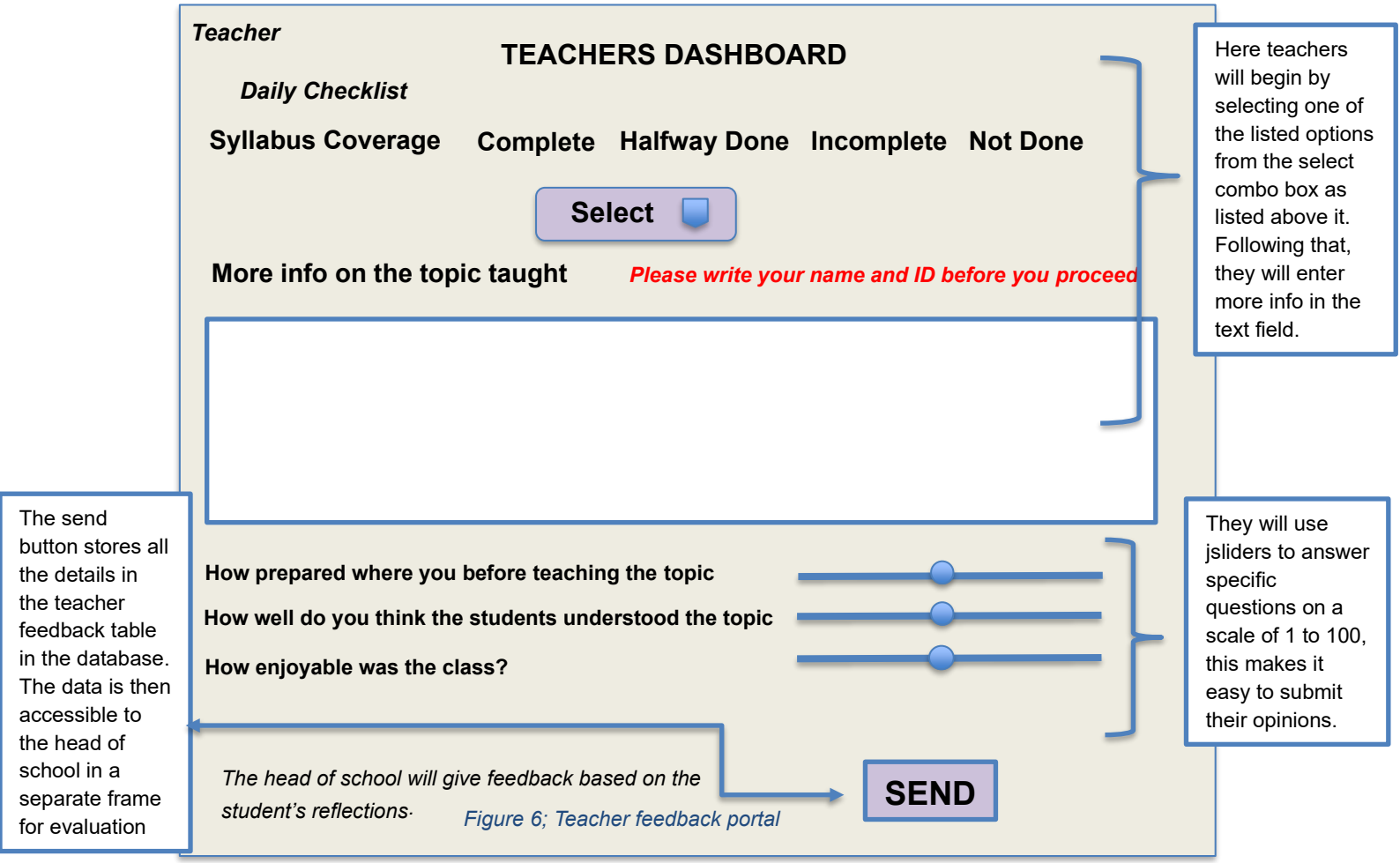


Figure 6; Teacher feedback portal

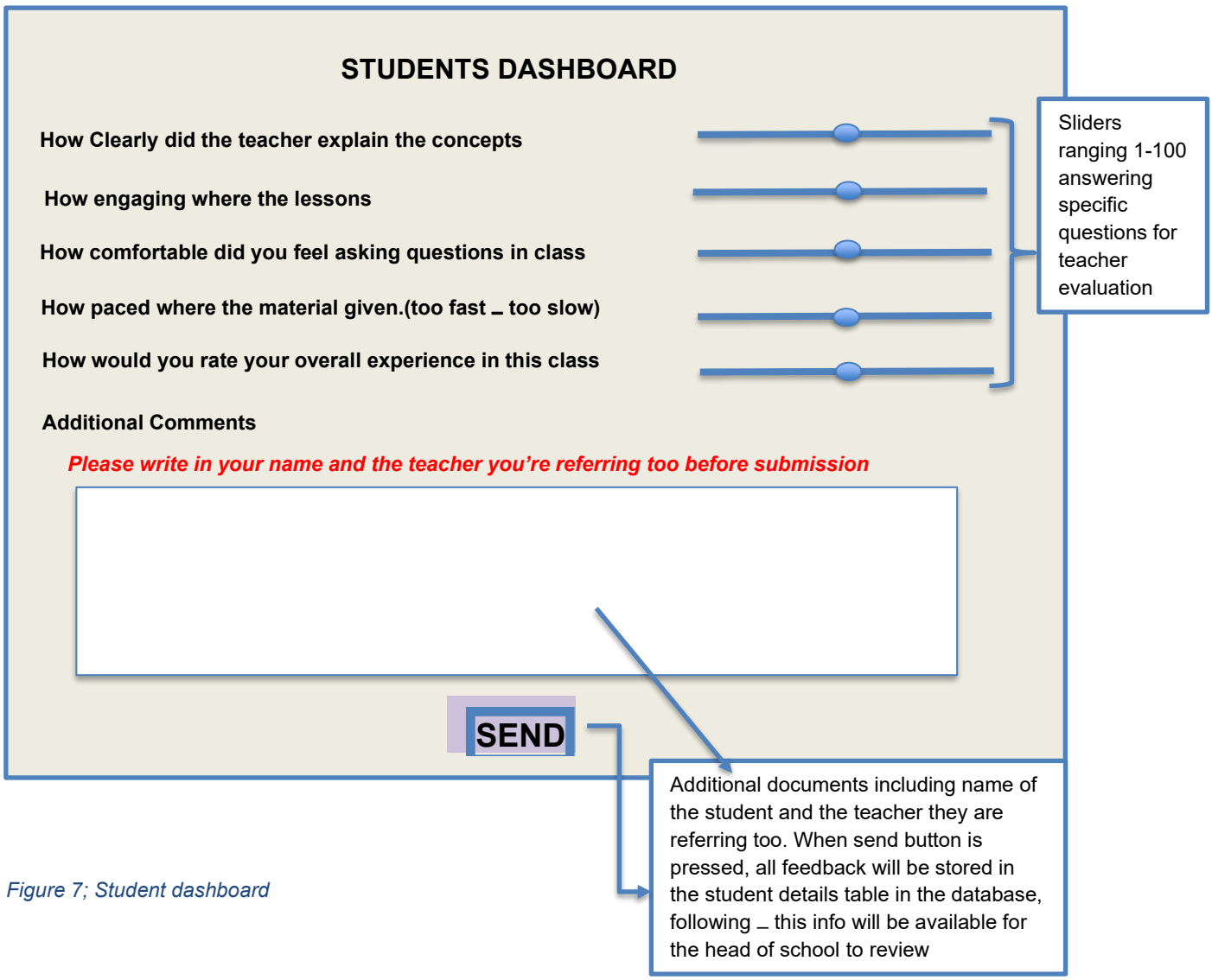
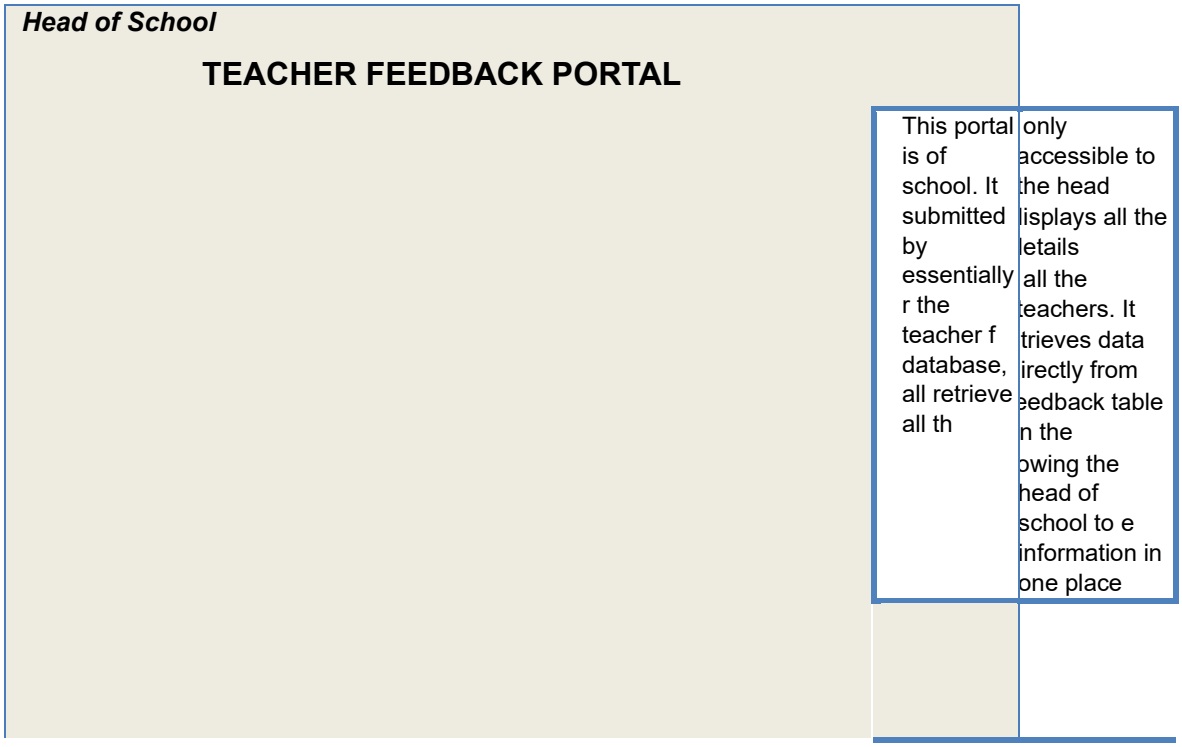


Figure 7; Student dashboard



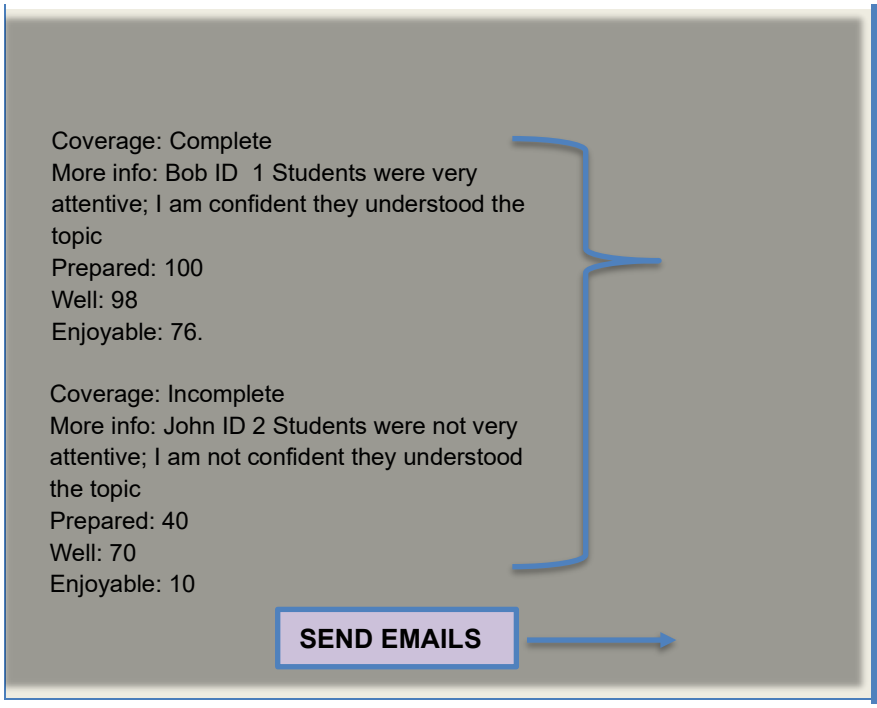


Figure 8; teacher feeback portal - only seen by the head of school

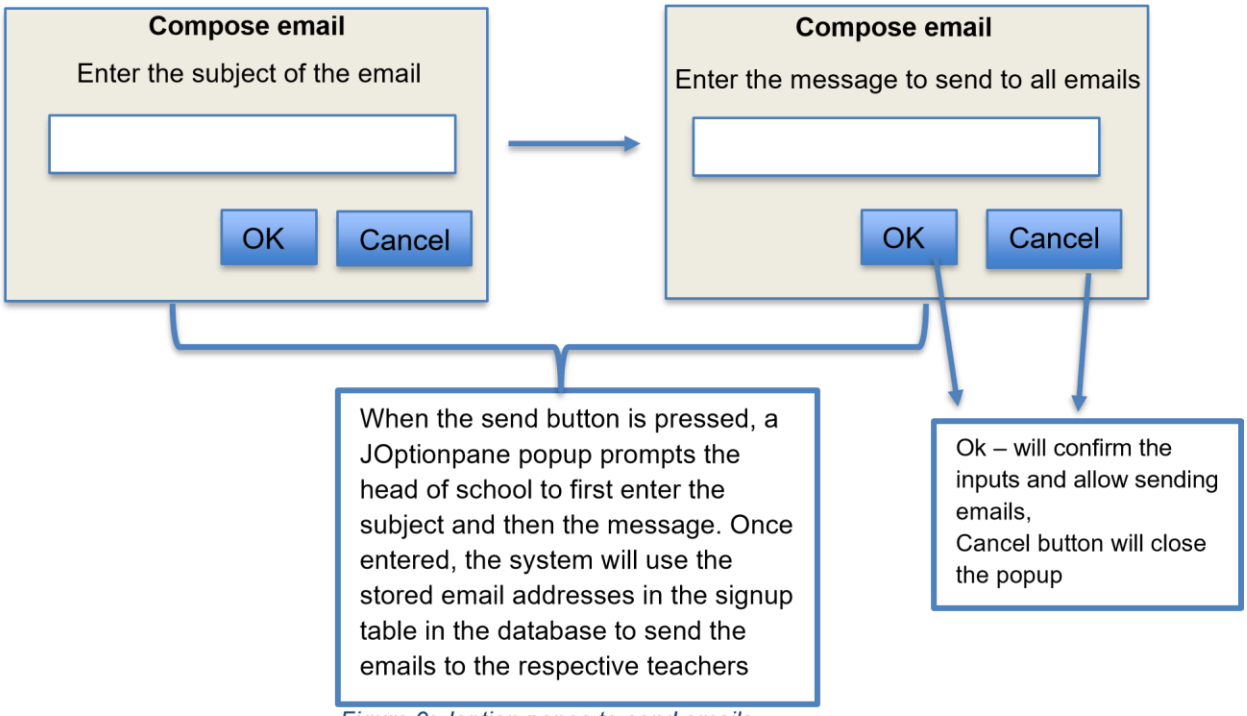
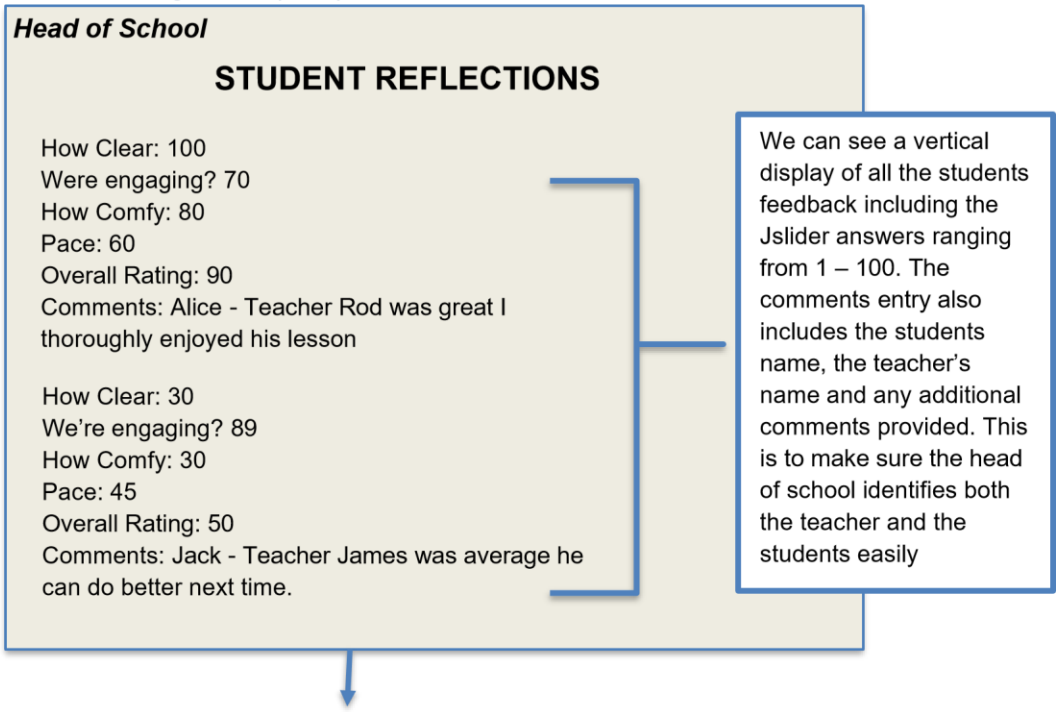


Figure 9; Joption panes to send emails



This portal displays all the feedback given by the students, which is stored in the student feedback table in the database. Once a student submits their reflections, the details the details are

saved and retrieved as a final step before the head of school can make any decisions about what to communicate with the teachers.

Figure 10; Students reflections page – only seen by the head of school

2. Database

Storage Systems	My SQLite Databases

Number of databases used	1
Number of Tables used	4
Format Database	db.
Database Facilitator	JDBC

Table 1; Table of summary

Table	Name
1	Signup
2	Teacher details
3	Student Feedback
4	Teacher feedback

Table 2; Tables available in the database

Field	Type	Null	Key	Notes
First Name	Text	No	Unique	
Last Name	Text	No		
ID No	Integer	No	Primary	
Email	Text	No	Unique	For receiving emails, communication between teacher and head of school
Password	Text	No		Enhances security in the program
Confirm Password	text	No		
Role	Enum	No		Used to direct the user to the correct frame based on their role.

Table 3; Sign up table

Field	Type	Null	Key	Autoincrement
id	integer	No	Primary key	Autoincrement
Subject	text	No		
day	text	No		

Teacher id	integer	No		
class	Text	No		
Additional info	Text	No		
Deadline	Date	No		

Table 4; Teacher details table

Field	type	null	key	Autoincrement
Id	Integer	no	Primary key	Autoincrement
How clearly	Integer	no		
How engaging	integer	no		
How comfy	integer	no		
How paced	integer	no		
Rating	integer	no		
Add comments	Text	no		

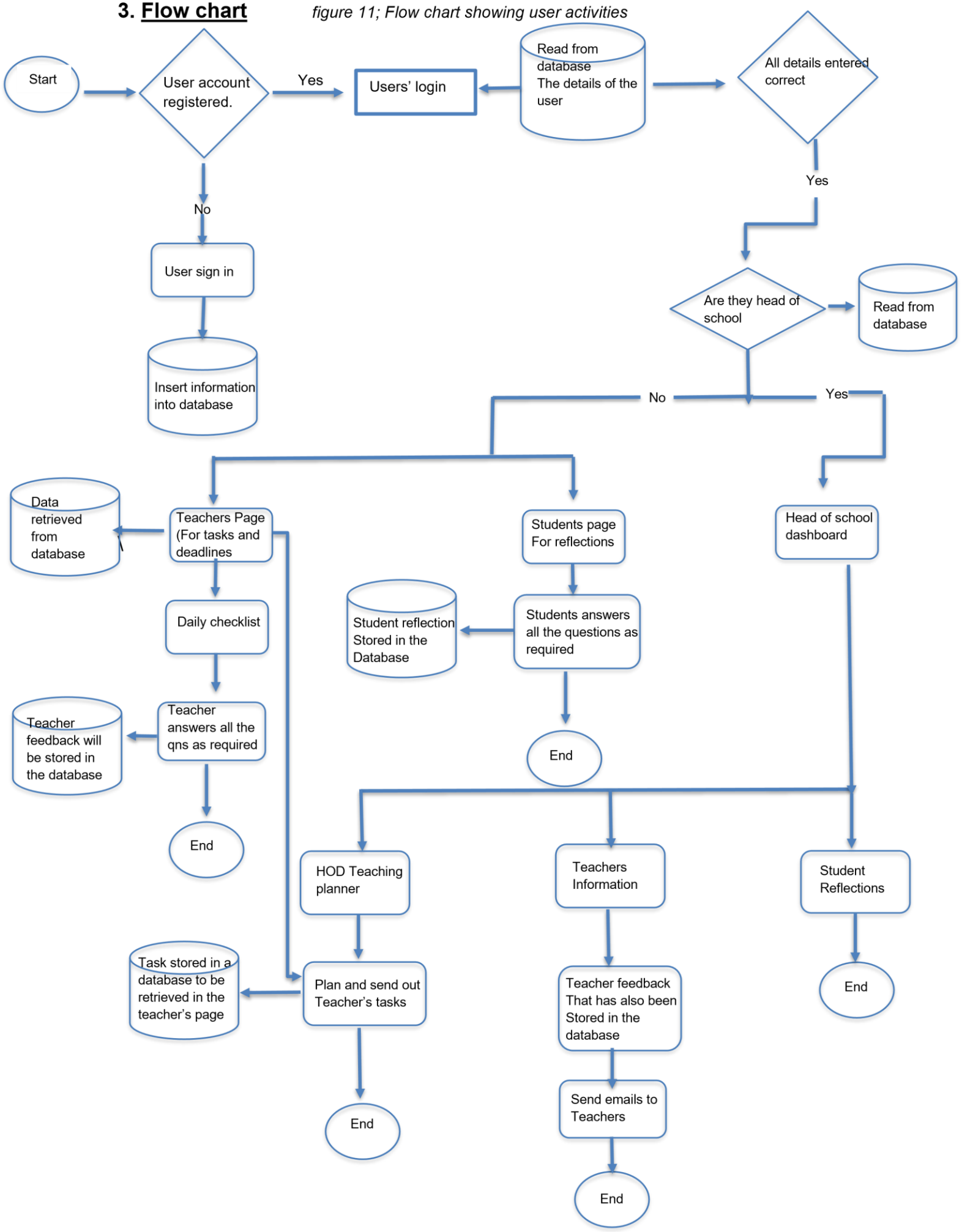
Table 5; Student feedback table

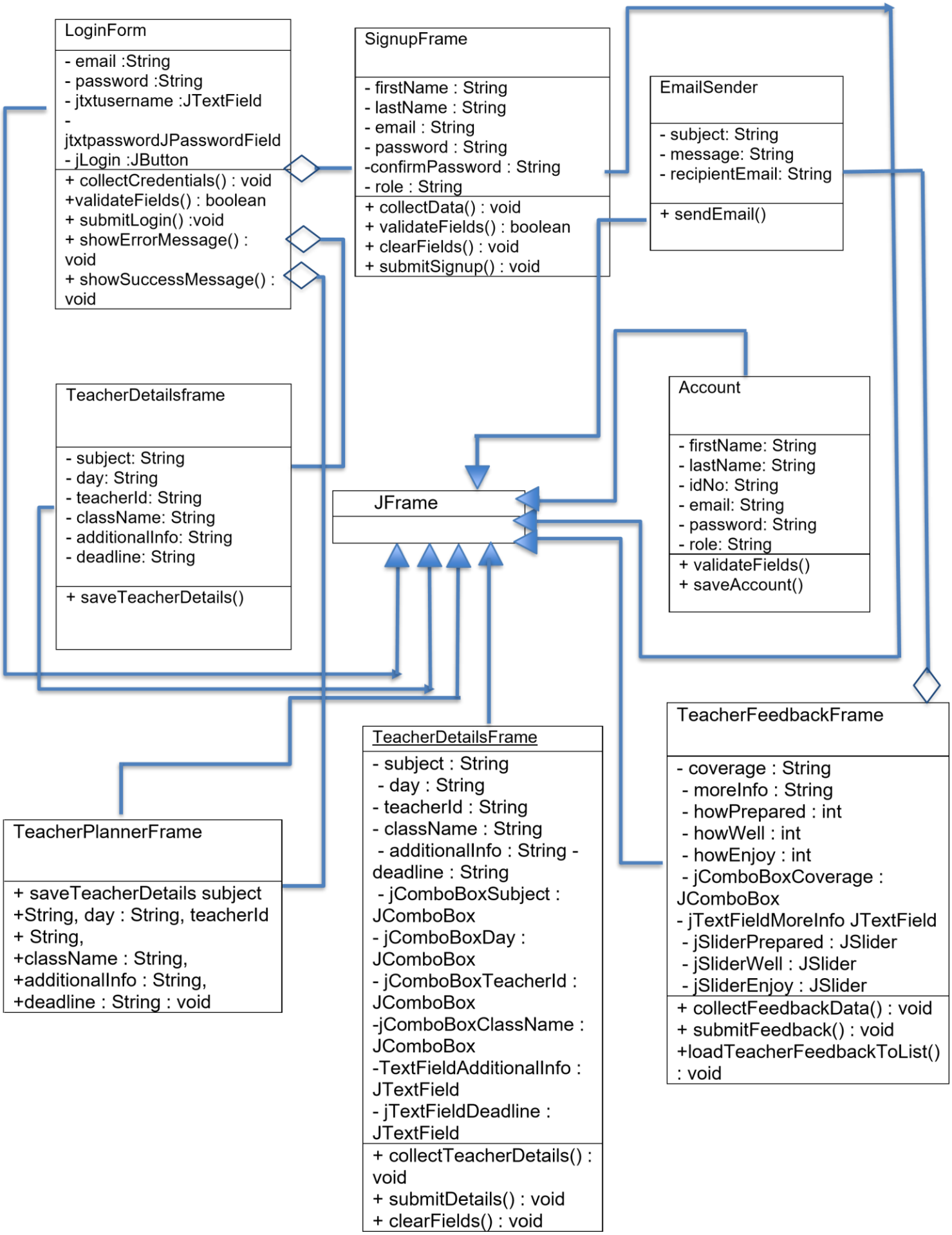
Field	type	null	key	Autoincrement
Id	Integer	no	Primary key	Autoincrement
Coverage	Integer	no		
More info	text	no		
How prepared	Integer	no		
How well	Integer	no		
How enjoy	Integer	no		

Table 6; teacher feedback table

3. Flow chart

figure 11; Flow chart showing user activities





4. UML DIAGRAMS

figure 12; UML Diagrams Showing GUI Design.

5. Pseudocode Table Structure

Class	Function	Pseudocode	Explanation
LoginPage	jLogin actionPerformed	email = GetText(jtxtusername) password = GetText(jtxtpassword) If email or password is empty ShowMessage("Please enter both email and password!") Else sql = "SELECT Role, IDNo FROM signup WHERE Email = ? AND Password = ?" result = ExecuteQuery(sql, email, password) If result is not empty role = result.get ("Role") idNo = result.get("IDNo") ShowMessage("Welcome! Role: " + role) If role == "Head of School" ShowPage("HeadOfSchoolPage") Else If role == "Teacher" ShowPage("TeacherPage") Else If role == "Student" ShowPage("StudentPage") Else ShowMessage("Unrecognized role!") Else ShowMessage("Invalid email or password")	The function fetches the username and password from the input fields and checks in the database to determine which page the user should be directed to based on their role.
SignUpPage	jSignUp actionPerformed	firstName = GetText(jTextField2) lastName = GetText(jTextField8) idNo = GetText(jTextField10) email = GetText(jTextField9) password = GetPassword(jPasswordField1) confirmPasswordGetPassword (jPasswordField2) role = GetSelectedItem(jComboBoxRole2) If any field is empty or passwords don't match ShowMessage ("All fields are required or passwords do not match!") Else sql = "INSERT INTO signup (First Name, Last Name, IDNo, Email, Password, Role) VALUES (?, ?, ?, ?, ?, ?)" ExecuteQuery(sql, firstName, lastName, idNo, email, password, role) ShowMessage("Account created successfully!") ResetFields() ShowPage("LoginPage")	The function validates the user input, checks if passwords match, and then saves the new user data into the database, creating a new account and redirecting the user to the login page.
TeacherPage	jAssignTask actionPerformed	subject GetSelectedItem(jComboBox10) day = GetSelectedItem(jComboBox12)	The function collects data from the form fields and saves it into
		teacherId = GetSelectedItem(jComboBox1) className = GetSelectedItem(jComboBox11) additionalInfo = GetText(jTextField1) deadline = GetText(jTextField3) SaveTeacherDetails(subject, day, teacherId, className, additionalInfo, deadline) ShowMessage("Details sent successfully!") ClearFields()	the database for the teacher's task, then shows a success message and clears the input fields.

TeacherFeedback	jSendFeedback actionPerformed	coverage = GetSelectedItem(jComboBox2) moreInfo = GetText(jTextField4) howPrepared = GetSliderValue(jSlider1) howWell = GetSliderValue(jSlider2) howEnjoy = GetSliderValue(jSlider3) sql = "INSERT INTO teacherfeedback (coverage, moreinfo, howprepared, howwell, how_enjoy) VALUES (?, ?, ?, ?, ?)" ExecuteQuery(sql, coverage, moreInfo, howPrepared, howWell, howEnjoy) If rowsAffected > 0 ShowMessage("Sent! Thank you for your feedback.") RefreshFeedbackList() Else ShowMessage("Failed to send feedback. Please try again.")	The function gathers feedback from the teacher's form, inserts it into the database, and provides a success or failure message based on the result.
Database	DBConnector.get Connection	Try LoadDriver("com.mysql.cj.jdbc.Driver") con = ConnectToDatabase("jdbc:mysql://127.0.0.1:3306/project", "root", "presidentofTz1!") If con is successful Return con Catch Exception ShowMessage("Connection failed")	The function establishes a connection to the database using provided credentials and returns the connection object if successful.
SendEmail	jSendEmail actionPerformed	subject = GetText(subjectField) body = GetText(bodyField) mailSession = CreateMailSession() SendEmail(mailSession, subject, body) ShowMessage("Email sent successfully") Catch Error ShowMessage("Failed to send email.")	The function collects email subject and body from the form fields and sends an email using the provided data, then shows a success or failure message.

Table 7; Main algorithm design

6. Test Plan

Test No.	Criteria	Expected Results	Method of Testing
1	Users must sign up with their information and log in using their username and password.	Successful login with the correct username and password displays the user's respective portal.	Sign up as "Jake Blake" using "jake@domain.com" and "password123". Log in to verify access to the portal based on the assigned role.
2	Passwords must be encrypted before storing them in the database.	Passwords are encrypted in the database to ensure privacy.	Check the database after Jake Blake signs up to confirm that the password is stored in an encrypted format.

3	Users can access specific portals based on their role (Teacher, Head of School, Student).	Users are redirected to their role-specific portal after login.	Log in as Jake Blake (Teacher), Head of School, and Alice K (Student) to verify that each user is redirected to their respective portal.
4	The Head of School can assign tasks to teachers.	The assigned tasks appear in the teacher's portal.	Log in as Head of School and assign a task to Jake Blake. Log in as Jake Blake to confirm the task appears in the teacher portal.
5	The Head of School can add notes, deadlines, and task details for teachers.	Task details, including notes and deadlines, are visible in the teachers' portal.	Assign a task with detailed notes and a deadline as Head of School and verify that they appear in Jake Blake's teacher portal.
6	Teachers can write and submit daily checklists and reflections.	Submitted checklists and reflections are saved in the teacher feedback database.	Log in as Jake Blake (Teacher), complete a checklist and reflection, and verify the data is saved in the database.
7	Teacher feedback is visible in the teacher feedback portal.	The teacher's feedback appears in the teacher feedback portal after submission.	Submit feedback as Jake Blake (Teacher) and verify that it is visible in the teacher feedback portal when logged in.
8	Students can send reflections visible only to the Head of School.	Student reflections are saved in the student feedback portal and accessible only by the Head of School.	Log in as Alice K (Student), submit a reflection, and verify that it is visible in the Head of School portal but not in the student portal.
9	The Head of School can send emails to teachers.	Teachers receive email notifications when tasks are assigned.	Assign a task as Head of School and verify that Jake Blake (Teacher) receives an email with the task details.
10	Users can navigate efficiently between frames.	Navigation between frames is smooth and without errors.	Test navigation by logging in as Head of School and moving through sections like "Assign Tasks" and "View Feedback".
11	The system saves all signup information securely in the database.	User signup information, including encrypted passwords,	Create accounts for multiple users and use SQL queries to confirm secure
		is securely stored in the database.	storage of usernames and encrypted passwords.
12	Teachers can verify the status of their assigned tasks.	Teachers can see all tasks with their respective status (e.g., completed, pending) on their portal.	Log in as Jake Blake (Teacher) and verify that all assigned tasks and their statuses are displayed in the teacher portal.
13	The system should allow the Head of School to check teacher availability.	Availability of teachers is displayed when assigning tasks.	Log in as Head of School and check the availability list for teachers before assigning tasks.
14	Tasks marked complete by teachers are automatically updated.	Completed tasks are automatically marked as completed in the teacher portal.	Log in as Jake Blake (Teacher), mark a task as completed, and verify that it is reflected in the teacher portal as completed.

Table 8; Test plan table

Word count - 0