

# ERD



# Class Diagram



# Use Case

# **Use Case Specifications**

|  |  |
| --- | --- |
| Name of Use Case | Login |
| Brief Description | Login to access certain data in respective portal |
| Actor | Administrator, Teacher, Member |
| Pre-condition | No user has login in the system |
| Post-condition | User logins into portal accessing data they can see |
| Special requirement | User has to be registered in the database |
| Flow of events | User enters the homepage and decides to login into his/her account |
| Expected flow | 1. User entered into landing page 2. User enters the login page 3. User enters particulars into the two textbox, one for User ID and the other, password   A1. User ID not found in database  A2. Password does not match   1. User enters particular’s page displaying particulars. |
| Alternative flow | A1: User ID not found in database   1. System prompts user regarding inexistence of user 2. Systems goes back to 3 3. However said website will display advice to either call or go to school to the register   A2: Password does not match   1. System prompts user regarding Password Error 2. Systems goes back to 3 3. However said website will display options to reset password |

|  |  |
| --- | --- |
| Name of Use Case | View Report |
| Brief Description | Viewing of Student’s progress report |
| Actor | Member |
| Pre-condition | Member wants to view his/child’s progress report |
| Post-condition | Student viewed his/child’s progress report |
| Special requirement | Member must be logged in |
| Flow of events | Member access the progress report after login into system |
| Expected flow | 1. After the student has login, he will be in the page displaying his particulars 2. User clicks on the button “View Progress Report”   A1. Member has children.   1. The progress report page loads and member view said report |
| Alternative flow | A1: Member has children   1. Instead of loading the report page, it loads the page displaying names of his/her children 2. Member clicks on the name of the progress reports belonging to the child he/she wants to see. 3. Systems goes to 3 |

|  |  |
| --- | --- |
| Name of Use Case | Update Particulars |
| Brief Description | Updating of User’s Particulars if any change is necessary |
| Actor | Member, Teacher, Admin |
| Pre-condition | User wants to update their particulars |
| Post-condition | User updates their particulars |
| Special requirement | User must be logged in.  Must be Admin to edit all Info. |
| Flow of events | User updates their particulars if any of the required information is being changed |
| Expected flow | 1. Student enters page after login 2. Click the ‘Edit Particulars Button’ 3. System loads all available Info that can be changed 4. User edits the info and submit changes   A1. Wrong Information type  A2. No changes made   1. Edits are updated into database 2. Particulars now show updated Information |
| Alternative flow | A1: Wrong Information Type   1. Error message pops out showing which values are wrong   A2: No changes made   1. Confirmation Message display confirming whether User wants to keep data as of last changes 2. If yes, system goes back to particular page 3. If no, system goes back to 4. |

|  |  |
| --- | --- |
| Name of Use Case | Change Password |
| Brief Description | Changing of password |
| Actor | Student , Admin , Teacher |
| Pre-condition | User wants to change password |
| Post-condition | Password of User is successfully changed |
| Special requirement | User must be logged in |
| Flow of events | Student logins to system with old password and changed the old password with the new one |
| Expected flow | 1. User enter the change password page after logging in 2. System loads the empty textbox for 3 values. 3. User enters old password and new password accordingly 4. User then have to enter their new password again in the final textbox to confirm their new password before clicking the submitted button   A1: Old Password does not match system  A2: New Password does not match   1. New Password is updated into System in place of Old password 2. An email is sent to User for verification confirmation |
| Alternative flow | A1: Old Password does not match   1. System display message mentioning mismatch of Old Password 2. Systems goes back to 2   A2: New passwords doesn’t match   1. System display message mentioning mismatch of the New Passwords 2. System goes back to 2 |

|  |  |
| --- | --- |
| Name of Use Case | Register new member |
| Brief Description | Registering of a New Member |
| Actor | Administrator |
| Pre-condition | A new member signs up for lesson |
| Post-condition | Administrator successfully registers a new member |
| Special requirement | Administrator must be logged in |
| Flow of events | Administrator needs to click on the register member tab in order to enter new member’s particulars into the registration form. |
| Expected flow | 1. Administrator enters the registration page after logging in 2. Administrator fills in the required information   A1: The information filled in contain error   1. Data is inserted into database 2. New member is registered into System 3. Email sent to member to notify registration and password. |
| Alternative flow | A1 : Information filled in contain errors   1. Administrator corrects the details and resubmit to the system |

|  |  |
| --- | --- |
| Name of use case | Edit Progress Report |
| Brief description | Administrator and teacher can edit student progress report |
| Actor | Administrator, Teacher |
| Pre-condition | A student finishes a lesson and the teacher needs to update the progress report. |
| Post-Condition | Actors updated the progress report |
| Special requirement | Teaching Staff Users must be logged in. |
| Flow of events | User logs into system and goes to the Courses tab and choose a student in it. Progress report page loads and user then updates the report. |
| Expected flow | 1. User logs into the system and reaches the landing page. 2. User then clicks the Course Tab 3. System loads the page filled with students taking chosen course and have latest lesson taught by User. 4. User chooses student name 5. Report page will load for the targeted student. 6. Staff edits the report and submits. 7. Report is updated into database 8. Notification to student/’s parent’s email. |
| Alternative flow |  |

|  |  |
| --- | --- |
| Name of use case | Take Attendance |
| Brief description | Teachers have to take attendance before the class begins |
| Actor | Teacher |
| Pre-condition |  |
| Post-Condition |  |
| Special requirement | Using mobile web app |
| Flow of events | Teachers need to login into the system through their smartphones before they can access the attendance system |
| Expected flow | 1. Teachers take attendance with their smartphones   A1: Not all teachers have the right to do so   1. The use case ends here |
| Alternative flow | A1: Not all teachers are tech-savvy   1. The teachers need to mark the attendance before submitting to the admin 2. The admin needs to manually key the attendance into the system 3. The use case ends here |

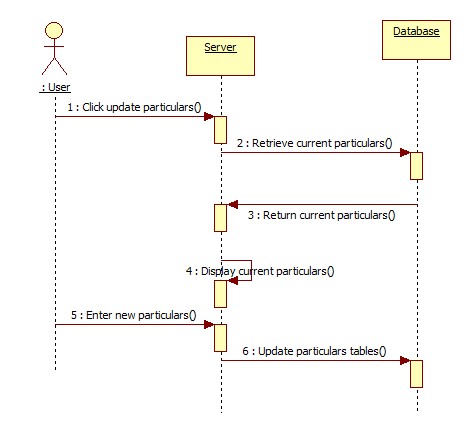
|  |  |
| --- | --- |
| Name of use case | Handle Payment |
| Brief description  (Describe the activity, not saying what actors can do) | Administrator handle the payment whenever he receives the school fees |
| Actor  (Actors involved) | Administrator |
| Pre-condition | Administrator receives the school fees from the parent |
| Post-condition | Administrator generates the receipt for the payment |
| Special requirement  (Must happen before Use case) | Administrator needs to login to the system |
| Flow of events  (what will happen) | Administrator generates the receipt for the parent after he receive the payment |
| Expected flow | 1. Administrator sends SMS reminder to remind parents to pay school fees for the month   A1: Parents is paying presently   1. Administrator receives the payment from the parent   A2: Parent had not make payment to the school for the month lesson  A3: Payment has been made to AboutMusic in advance   1. Administrator generates the receipt for the parent |
| Alternative flow | A1: Parent is paying presently   1. Skip to 2   A2: Parent has not make payment to the administrator   1. Administrator will indicate payment is not make in the system 2. Administrator will send more SMS reminders if payment is still not received after a period of time   A3: Payment has been made in advance to AboutMusic   1. Skip to 3 |

|  |  |
| --- | --- |
| Name of use case | Calculate teachers’ pay |
| Brief description | Administrator calculates teacher’s pay base on the generated teacher schedule |
| Actor | Administrator |
| Pre-condition | Administrator has not calculated teachers’ pay for the month |
| Post-Condition | Administrator generate teachers’ pay after calculation |
| Special requirement | Administrator access the system to generate teachers’ pay |
| Flow of events | Administrator needs to access the system and retrieve the teacher schedule before calculating the pay |
| Expected flow | 1. Administrator access the payroll system 2. He needs to retrieve the teachers’ teaching schedule 3. Pay is generated after matching the teaching schedule and lesson type 4. Payslip is generated once pay is confirmed |
| Alternative flow |  |

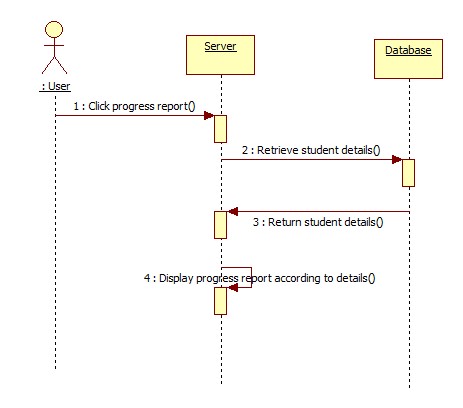
|  |  |
| --- | --- |
| Name of use case | Handle schedule |
| Brief description | Administrator handles the schedule including lesson change and notify the changes made. |
| Actor | Administrator |
| Pre-condition | Administrator has already planned the schedule |
| Post-Condition | Administrator makes new changes to the schedule if required |
| Special requirement | Administrator needs to login to the system |
| Flow of events | Administrator |
| Expected flow | 1. Administrator has already planned out the schedule 2. Administrator makes changes to meet teachers’ request 3. System is updated and changes will be notified to students and teachers |
| Alternative flow |  |

# C:\Users\Fujitsu\Desktop\SequenceDiagram1.jpg

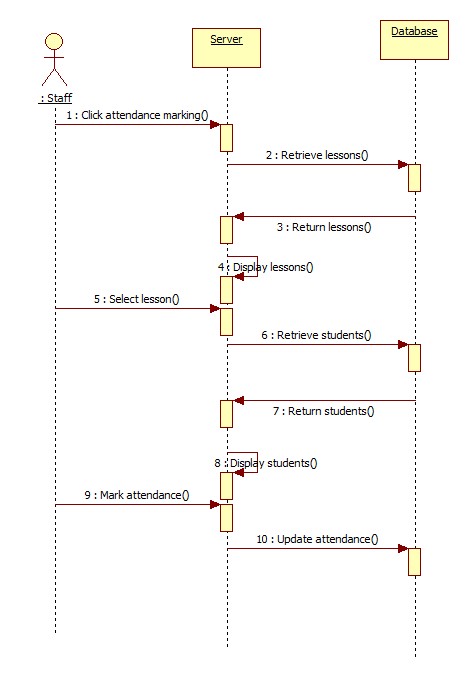
# Login



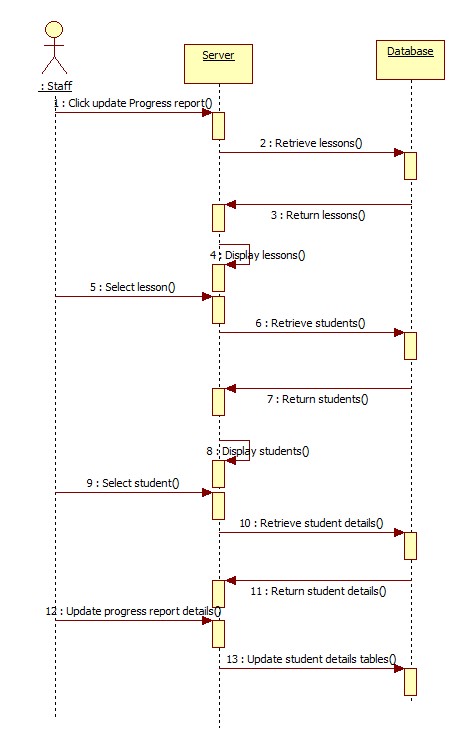
# Update particulars (ALL)



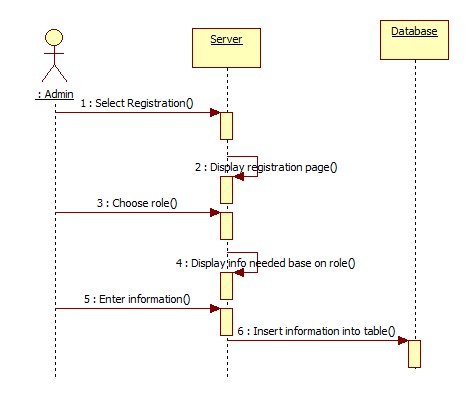
# View progress report (USER)



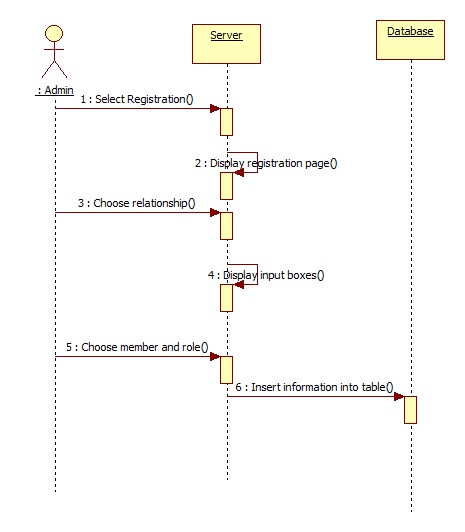
# Mark attendance (Staff, if given rights)



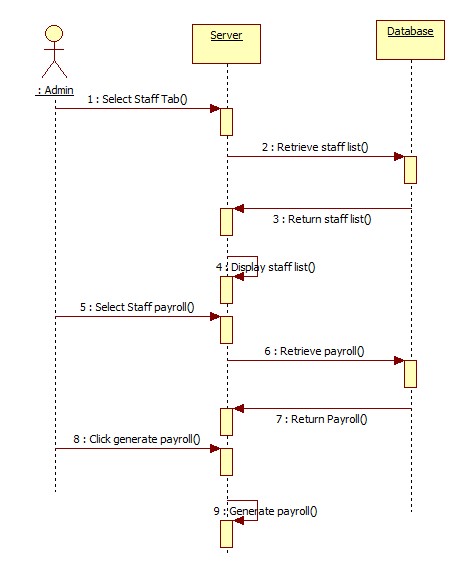
# Update progress report (Staff, Admin)



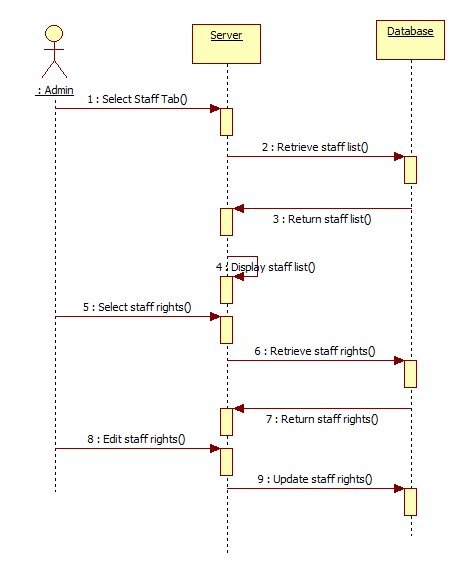
# Registration (ADMIN)



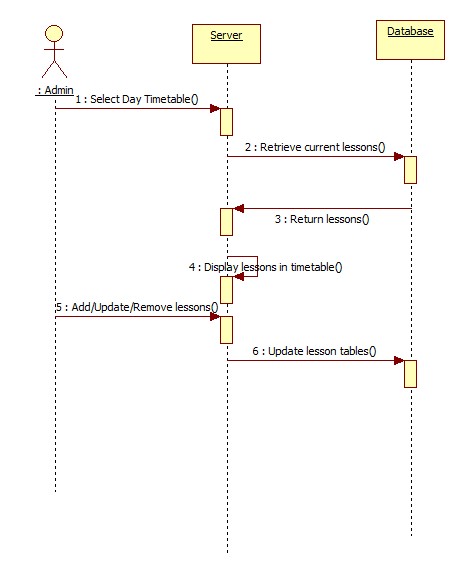
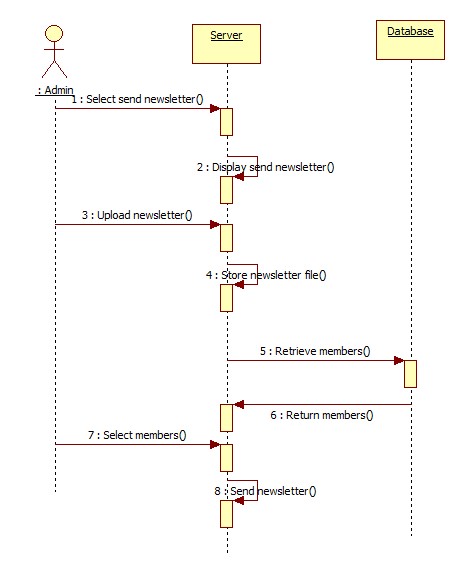
# Add relationship



# Generate payroll (ADMIN)



# Edit staff rights (ADMIN)



# Send newsletter (ADMIN)

# Timetable (ADMIN)