

**GROUP ASSIGNMENT**

**TECHNOLOGY PARK MALAYSIA**

**AAPP006-4-2-SDP**

**SOFTWARE DEVELOPMENT PROJECT**

**DOCUMENTATION**

**HAND OUT DATE: 7 DECEMBER 2020**

**HAND IN DATE: 22 MARCH 2021**

**WEIGHTAGE: 100%**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**INSTRUCTIONS TO CANDIDATES:**

**1.Submit your assignment at the administrative counter.**

**2. Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing).**

**3. Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.**

**4. Cases of plagiarism will be penalized.**

**5. The assignment should be bound in an appropriate style (comb bound or stapled).**

**6. Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate should be on a CD in an envelope/ CD cover and attached to the hardcopy.**

**7. You must obtain 50% overall to pass this module.**



**Software Development Project – Cover Sheet**

ASSIGNMENT

**Instruction:**

* Marks will be awarded for good presentation and thoroughness in your approach.
* Referencing Code: If you use some code, or ideas for code, which are taken or adapted from another source (book, magazine, internet, discussion forum, etc.), then this **must** be cited and referenced using the Harvard Name convention within your source code. Failure to reference code properly is considered as plagiarism.
* Complete this cover sheet and attach it to your project first page.
* This project is to be attempted by a group of 3-4 students.

**Student declaration:**

|  |  |
| --- | --- |
| I declare that: | * We understand what is meant by plagiarism * The implication of plagiarism have been explained to us by our lecture * This project is all our work and we have acknowledged any use of the published or unpublished works of other people |

Group Leader’s Signature: \_\_\_\_\_\_\_\_   Date: 10 December 2020

|  |  |  |
| --- | --- | --- |
| Project Title: | Online Attendance System | Intake: UCDF1905ICT(SE) |
| Role | Name | Signature |
| Project Leader | Leong Wei Loon |  |
| Project Assistant | Simpson Chong Sheng An |  |
| Programmer | Li Jia Chong |  |
| Programmer | Loo Yi Xiang |  |

Table of Contents

[**Acknowledgement** 6](#_heading=h.gjdgxs)

[**Abstract** 6](#_heading=h.30j0zll)

[**Workload Matrix** 6](#_heading=h.1fob9te)

[**Introduction** 7](#_heading=h.3znysh7)

[**Project Background** 7](#_heading=h.2et92p0)

[**Project Context** 7](#_heading=h.tyjcwt)

[**Project Solution** 8](#_heading=h.3dy6vkm)

[**Project Objective** 8](#_heading=h.1t3h5sf)

[**Project Scope** 9](#_heading=h.4d34og8)

[**Project Plan** 9](#_heading=h.2s8eyo1)

[**System Development Methodology** 9](#_heading=h.17dp8vu)

[**Project Gantt Chart** 11](#_heading=h.3rdcrjn)

[**System Hierarchy Chart** 13](#_heading=h.26in1rg)

[**Context Diagram** 14](#_heading=h.lnxbz9)

[**Data Flow Diagram Level 0** 15](#_heading=h.35nkun2)

[**Data Flow Diagram Level 1** 16](#_heading=h.1ksv4uv)

[**1.0 Manage Login** 16](#_heading=h.44sinio)

[**2.0 Register Account** 17](#_heading=h.2jxsxqh)

[**3.0 Timetable Process** 17](#_heading=h.z337ya)

[**4.0 Generate Code** 18](#_heading=h.3j2qqm3)

[**5.0 Manage Feedback** 19](#_heading=h.1y810tw)

[**6.0 Manage News** 20](#_heading=h.4i7ojhp)

[**7.0 Consultation Process** 21](#_heading=h.2xcytpi)

[**8.0 Manage Class List** 22](#_heading=h.1ci93xb)

[**9.0 Manage User** 23](#_heading=h.3whwml4)

[**Data Dictionary** 23](#_heading=h.2bn6wsx)

[**Entity Relationship Diagram (ERD)** 49](#_heading=h.qsh70q)

[**Flowcharts** 50](#_heading=h.3as4poj)

[**Screen Design and User Manual** 75](#_heading=h.1pxezwc)

[**Report Designs and User Manual** 97](#_heading=h.49x2ik5)

[**Test Plan** 107](#_heading=h.2p2csry)

[**User Acceptance Testing** 118](#_heading=h.147n2zr)

[**Significant Source Codes** 123](#_heading=h.23ckvvd)

[**Conclusion** 128](#_heading=h.ihv636)

[**Assumptions** 128](#_heading=h.32hioqz)

[**Limitation** 129](#_heading=h.1hmsyys)

[**Enhancements** 129](#_heading=h.41mghml)

[**Reference** 129](#_heading=h.2grqrue)

[**Appendix** 129](#_heading=h.vx1227)

[**Meeting Details** 129](#_heading=h.3fwokq0)

[**Installation Guide** 130](#_heading=h.1v1yuxt)

# **Acknowledgement**

After exposing and finishing this assignment. We would like to give special thanks to our lecture Ms Salasiah Sulaiman for giving us this golden opportunity to create this Online Attendance System. We all have learnt much information about the system and about creating the system.

Secondly, we would like to thank our web development lecture from last semester who have brought us into this course which is Ts. Dr. Syed Mohd Zahid Zainal Arrifin. He had taught us many information which we have implemented in this assignment.

# **Abstract**

This particular system is to create an Online Attendance System for a university, it mainly provides simple process like taking attendance, looking at announcements, looking at announcements and more. We have used few programming languages during this system was created. The languages include HTML, PHP, and CSS. This system is currently successful after created by me and my group. In conclusion, this simple system is created to run an attendance system for a university.

# **Workload Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Group Component | Leong Wei Loon | Li Jia Chong | Simpson Chong Sheng An | Loo Yi Xiang | Total % |
| **Implementation Part** |  |  |  |  |  |
| Implementing the system | 30% | 30% | 30% | 10% | 100% |
| **Documentation Part** |  |  |  |  |  |
| Writing the document | 25% | 25% | 25% | 25% | 100% |

# **Introduction**

## **Project Background**

Attendance is as important as result of your subject. By having a good result in every subject, it is said to lighten then weight for university students to find jobs. But as easy as it goes, normally, students cannot take their exam when their attendance is too low and even if they get excellent results with low attendance rate, some company will think that he/she took things lightly and not good a handling time when working.

With that being said, we took this opportunity to make an ideal system mainly for universities students to take attendance. By building an attendance system, lecturer do not have to worry about losing the attendance list after the whole class filed up the list. Other than that, it is easier for students to keep track with their attendance.

We hope that our system will make the process of taking attendance with ease for both students and lecturer. Other than that, we also hope that university students will be more aware of the importance of their attendance though the system and not just some numbers for taking their exams.

## **Project Context**

The current system being used sometimes brings inconvenient towards both students and lecture. For example, when the lecture is calling name one by one after the class ends, it delays time by a lot. If there are more than 100 students, it will be very tiring and time-consuming for the students to get their attendance.

On the other hand, the current system might be slow sometimes, for example, if there are few classes that are going to take their attendance at the same time, the system might be slow and even server going down.

Moreover, during online classes, there might have some students who did not attend for the class, but when the lecture takes attendance, the student will still get his/her attendance marked with the help of their friends.

## **Project Solution**

The solution to overcome the problems mentioned above. In the system that we are going to generate few methods. It is mainly about fixing the current bugs. Improving the system, creating a better version from the old one.

The solution is generating a code. From creating codes, students need to be online or on the spot to take their own attendance. This makes it hard to impossible for a student to take multiple attendance for their friends in a short period of time.

Overall, this solution can improve the whole system, creating a better system than the old one. Lectures can take the attendance for the whole class at the same time, reducing the time needed and not wasting any time.

## **Project Objective**

The main objective of our website is We wanted to provide our users with a simple and reliable tool with high productivity for our project target. Through this framework, we strive to provide schools with the same amount of assistance, particularly in terms of attendance management. We also strive to incorporate a system that is increasingly reliable and does its job without any mistakes or exploits. By doing so, Students and lecturers can easily take attendance through our website without any time consuming.

Other than that, increase the efficient when using our website is one of our objectives. Because lecturer having a problem like take attendance through handwriting or calling names would make students wait or causing time-consuming if no one is answering. Of cause student and lecturer, satisfaction is also one of our website main objectives but to make that happen, we need to make our work more approachable for lecturer and student to make their experience on taking attendance more smooth and easier for them. The latecomers cannot escape from the background of low attendance. With more precision, users will also know how to better manage their time.

Furthermore, Convenient is also one of our objectives. We got added some function that may can help user solve their problem. For example, lecturer, student and admin can edit their profile, after they login into our website. Student can search for lecturer detail and student able book consultant time. This function can let student book consultant time when they still any question that does not understand in class. Admins can manage the website, which mean admin can view student detail, and delete student detail.

Besides, improve website environment is also one of our objectives. In our website, every user can send feedback when user having some issues. For example, student and lecturer face taking attendance problem (cannot record student name or cannot generate code) then make user confusing, those can be report in give feedback function, and admin will fix those problem.

## **Project Scope**

The main scope for our website is for the education area. It is because students and lecturers can be using our website to successfully take attendance. So, the lecturer and student can be focusing on education without wasting time calling the student to take attendance. For example, our website can be using code to let the student take attendance.

Furthermore, our framework is the most appropriate for the user of the browser. It will not be limited to users of phones and PCs. Through the connection given, everyone can access and use our system. It is a method of attendance that develops an easy and high efficiency system for any university that needs it. We are not restricted to any university, either, or they can register as our users.

 Moreover, we will keep improvement with the website to become a best attendance system in Malaysia, but we never stop improve our system until the date we prepare to do one newest website. Although it is hard for us to create a best attendance system in Malaysia but At least we will make it as a best choice for people to choose as their school attendance system.

# **Project Plan**

## **System Development Methodology**

**Methodology**

Boehm’s Spiral model is a combination of a waterfall model and iterative model. Each phase in the spiral model begins with a design goal and ends with the client reviewing the progress. The development team in the Spiral-SDLC model starts with a small set of requirements and goes through each development phase for those sets of requirements. There are no fixed phases such as specifications or design. The loops in the spiral are chosen depending on what is required. The risks are explicitly assessed and resolved throughout the process. Boehm’s Spiral model is also very useful when it comes to uncertainty due to, we can add, delete, or change functionality at a later stage.

**Identification Phase**

There are a total of 4 different phases in Boehm’s Spiral model. Those phases are identification, risk analysis, development, and planning. It all begins with identification. In this phase, project objective and project scope for the phase is identified. It begins by collecting the feature requirements in the baseline spiral. As the product evolves, system requirements, component info and unit-level inputs occur in this phase. It also involves learning the system requirements through constant communication between the client and the system analyst. Alternative solutions possible for each phase are also proposed in this phase.

**Risk Analysis Phase**

The second phase being risk analysis. In this phase, a detailed analysis is carried out. All possible solutions are evaluated to select the best possible solution to minimize risk. For example, if there is a risk in the requirements is inappropriate, a prototype system may be developed for testing. Not only that, but risk analysis also comprises classifying, measuring, and observing technical feasibility and business risks such as schedule impact and cost escalations.

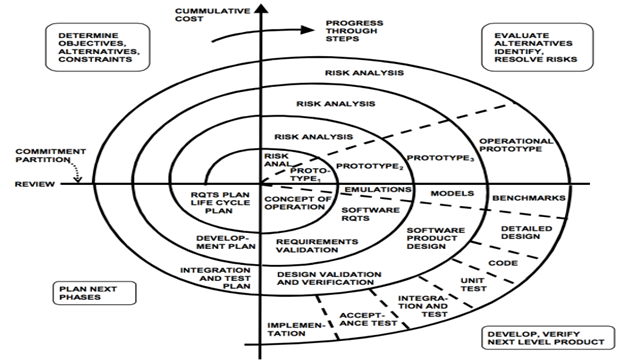
**Development Phase**

Development is the third phase. In this phase, the programmer will need to start coding based on the objectives and requirements. A unit testing, integration testing and acceptance testing will be carried out after the coding part is done to ensure that it meets all the requirements stated and also to avoid system defects. Developers are required to create a test plan that will include all the testing details and testing results. A test summary report and defect report will be generated after each test to help with the planning of next phase for the project.

**Planning Phase**

The fourth phase being the planning phase. In this phase, the project and the reports will be reviewed.  Developers will then be required to decide about whether to continue with the next phases or not. If it is decided to continue, a plan will be drawn up for the next phase of the project. New objectives and requirements will also be identified in the next phase. A test plan will also be generated during the planning phase.

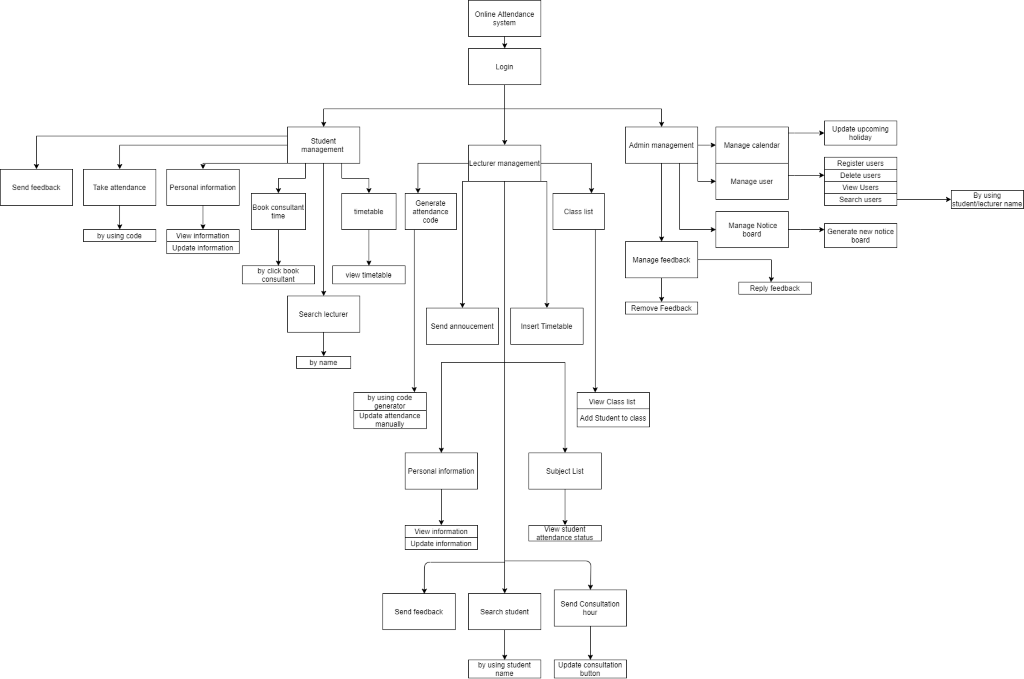
**Spiral model EXP**

****

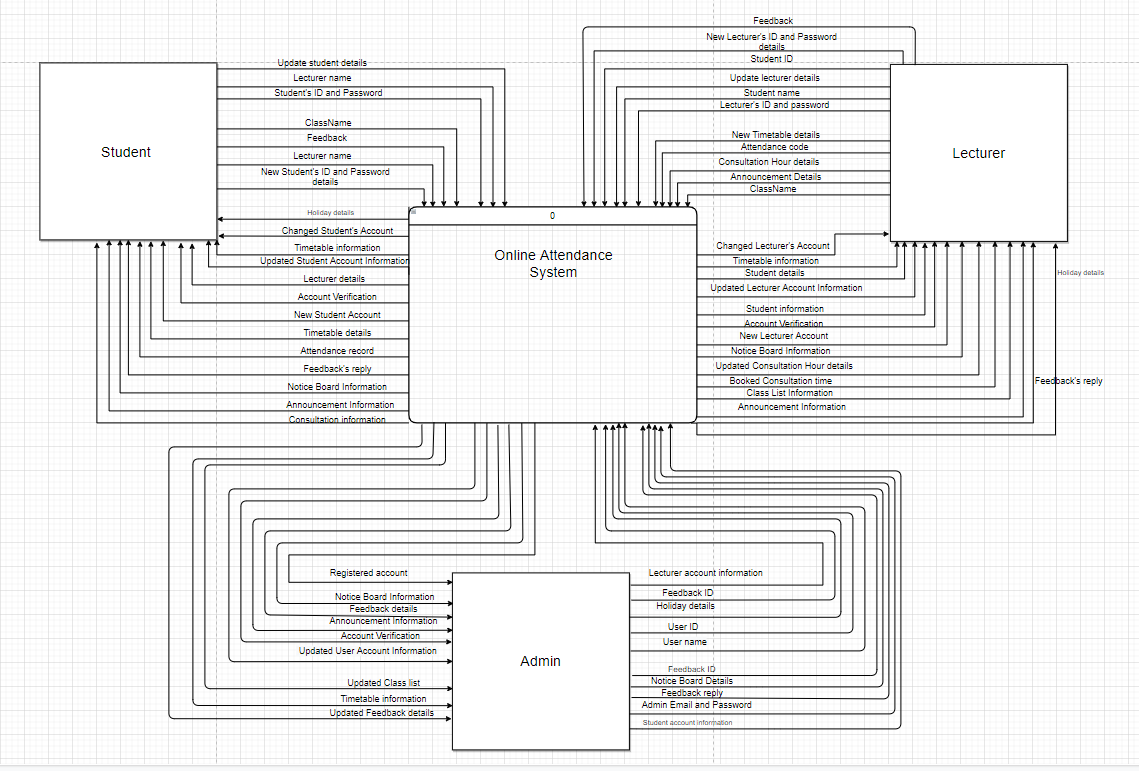
## **Project Gantt Chart**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task               \             Week | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** |
| Introduction |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Group deviation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Task allocation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Determine objectives phase** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Project Background |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Problem Context |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Proposed Solution |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Project Objectives |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Project Scope |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Identify and resolve risks phase** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| risk analysis 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| risk analysis 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| prototype of DFD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| risk analysis 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| prototype of ERD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| risk analysis 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| prototype of design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Development and testing phase** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| System Hierarchy Chart |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Context Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Data Flow Diagrams Level 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| - Data Flow Diagrams Level 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Entity Relationship Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Screens Design and User Manual |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| data dictionary |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Flowcharts |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Code Program |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| unit testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| integration testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| acceptance testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **planning phase** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| operation planning |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| system design discussion |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| test plan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

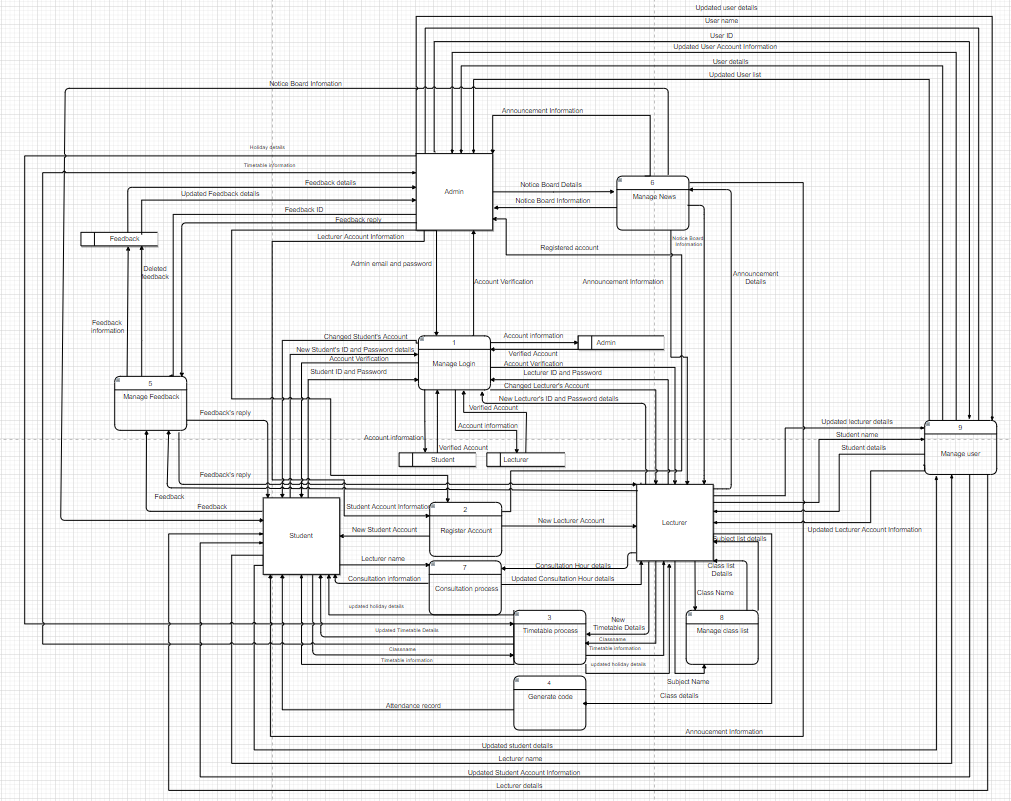
# **System Hierarchy Chart**



# **Context Diagram**

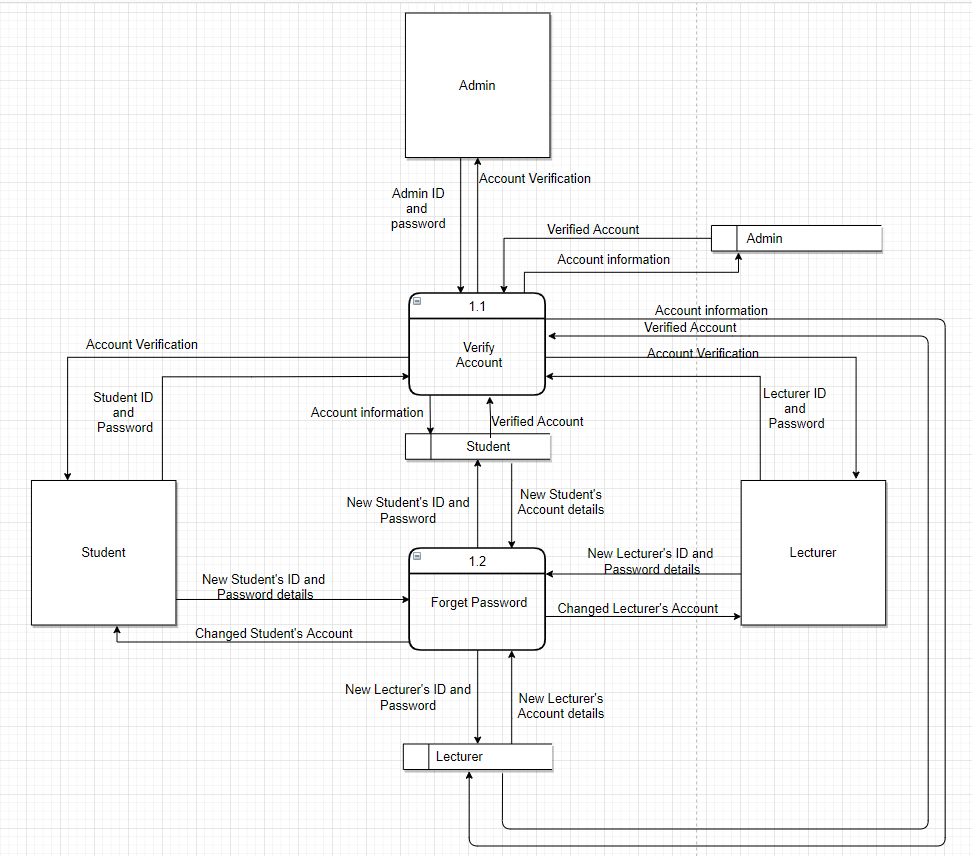


# **Data Flow Diagram Level 0**

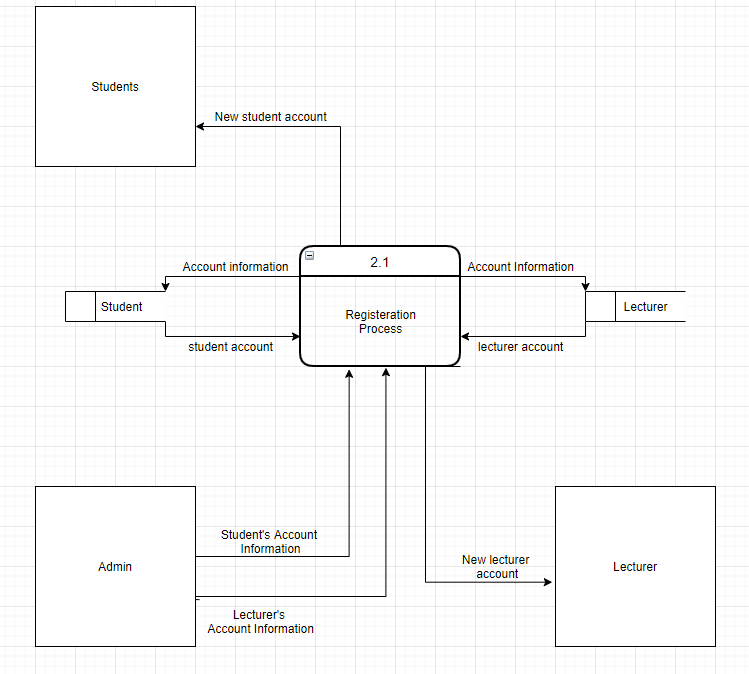


# **Data Flow Diagram Level 1**

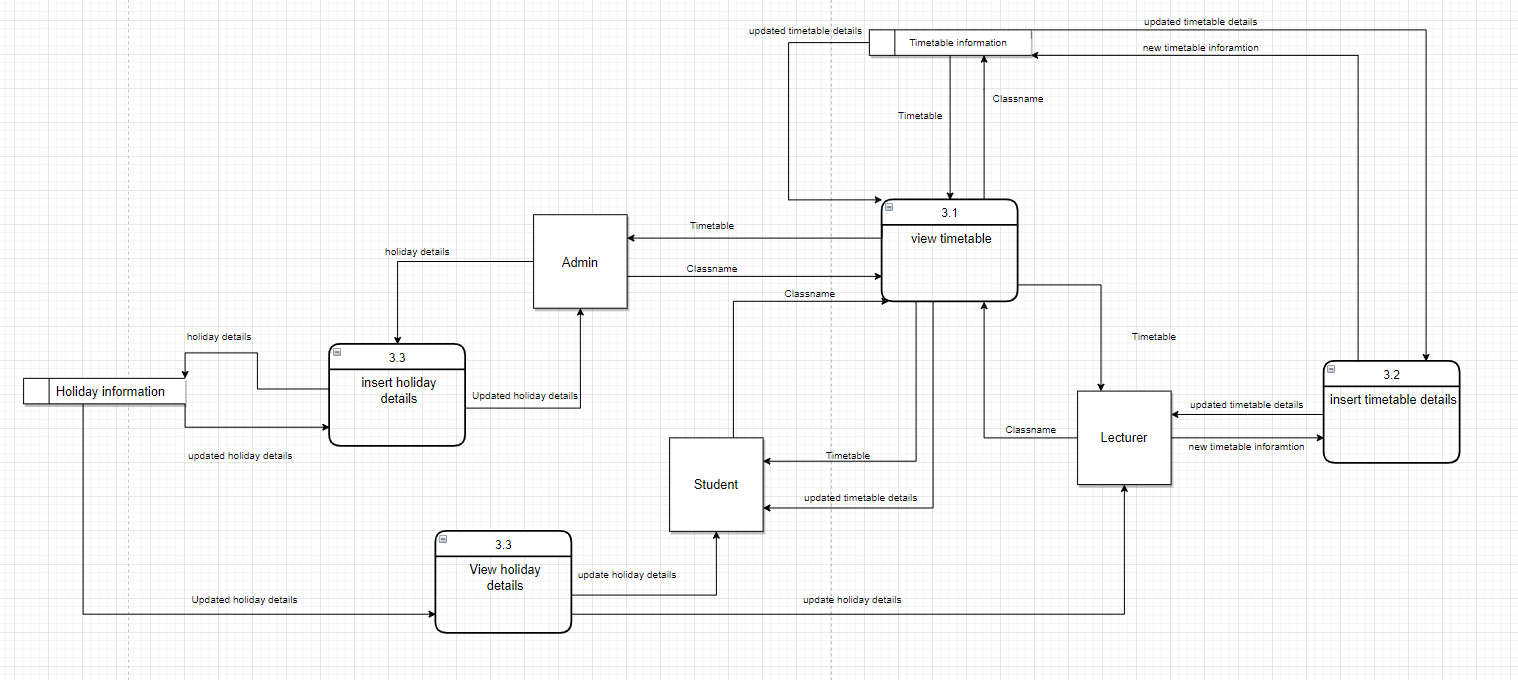
## **1.0 Manage Login**



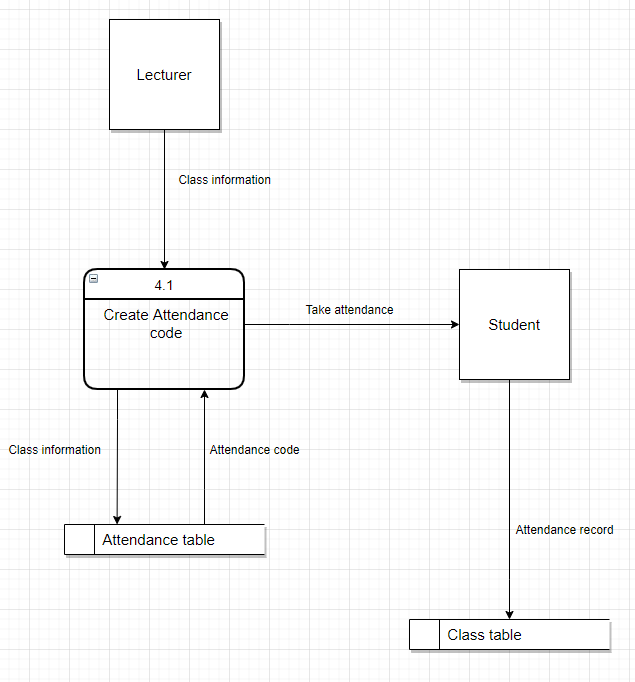
## **2.0 Register Account**



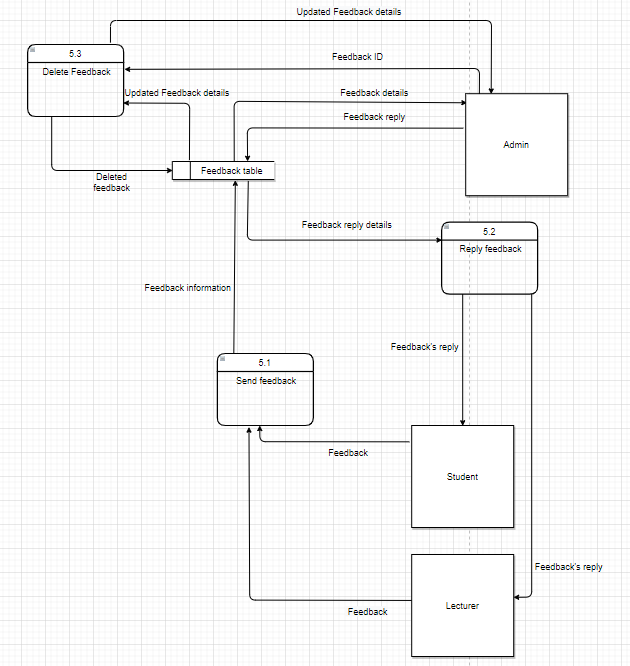
## **3.0 Timetable Process**



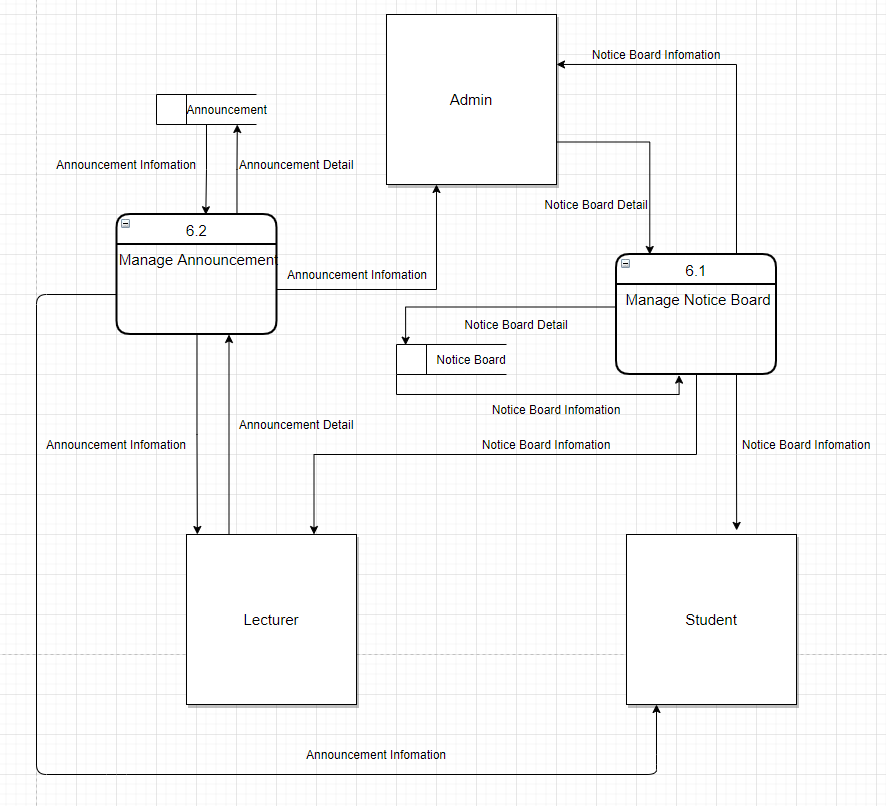
## **4.0 Generate Code**



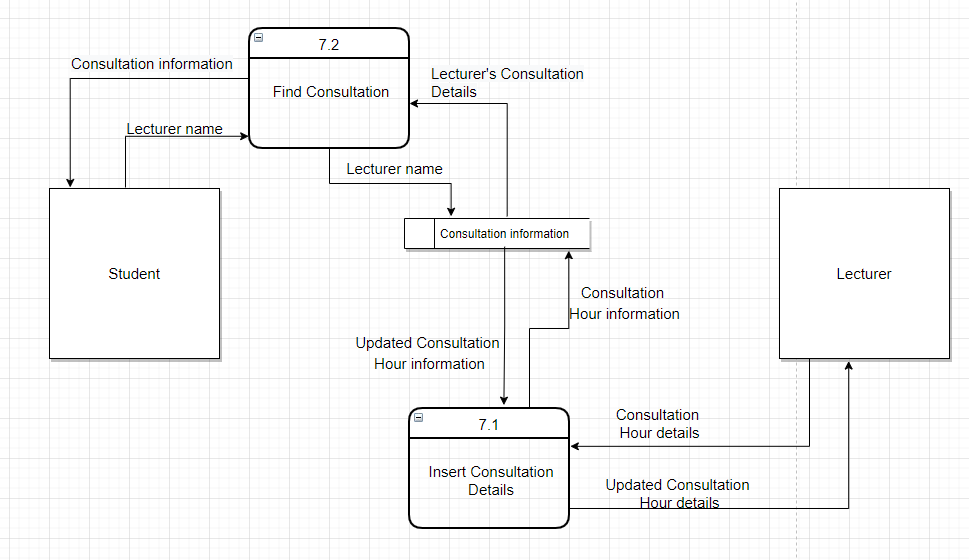
## **5.0 Manage Feedback**



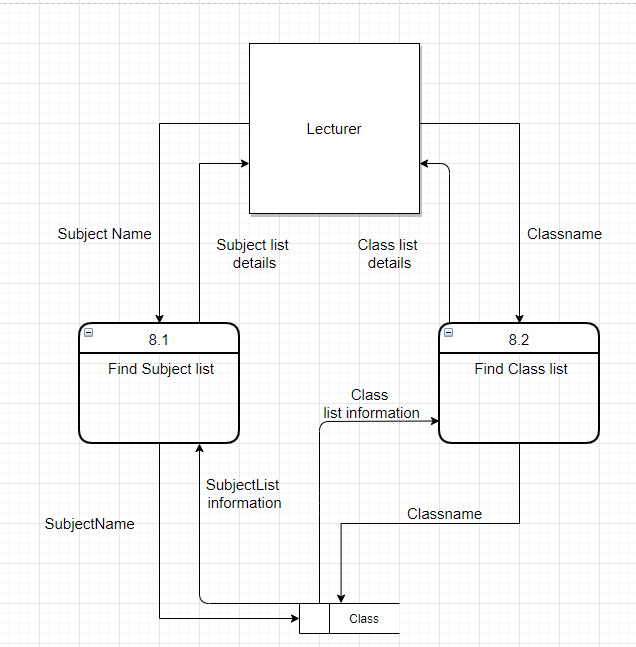
## **6.0 Manage News**



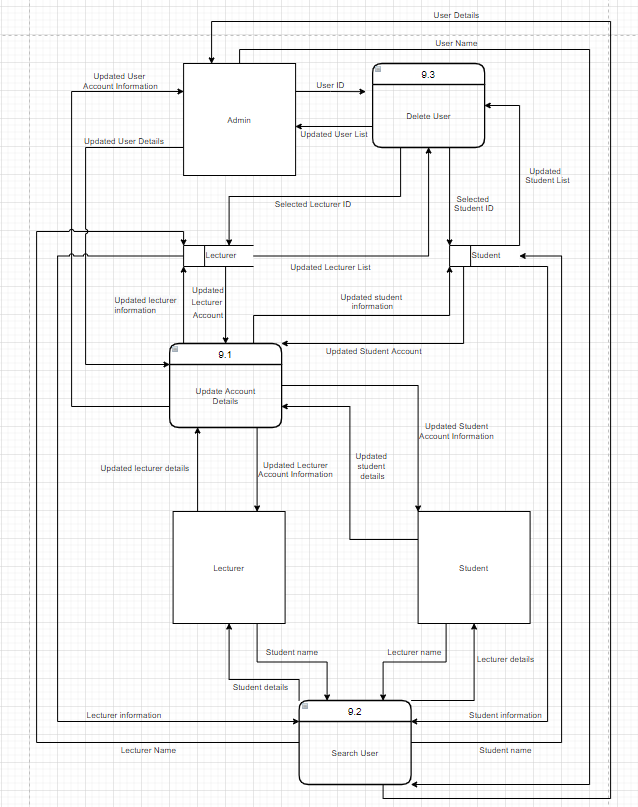
## **7.0 Consultation Process**



## **8.0 Manage Class List**



## **9.0 Manage User**



# **Data Dictionary**

**External Entity: Student**

|  |  |
| --- | --- |
| **Name** | **Student** |
| **Description** | Student can do many activities included login to their account using their **Student ID and Password**, if they do not have an account yet, they can register an account by sending us their email and our system will provide the **registration link** to them. Students can also do other activities such as **find timetable**, **find consultation**, **give feedback, manage their account** and many more. |
| **Input data flows** | Account Verification, Changed Student’s Account, Feedback’s reply, Notice Board Information, Lecturer details, Updated Student Account Information, Announcement detail, Attendance record, Calendar Information, Updated Timetable details, Consultation information, Registration Link. |
| **Output data flows** | Student ID and Password, New Student ID and Password details, Feedback, Lecturer name, Updated Student details, Month, Student’s email, |
|  |  |

**External Entity: Lecturer**

|  |  |
| --- | --- |
| **Name** | **Lecturer** |
| **Description** | Lecturer can do many activities such as login with **Lecturer ID and Password**, register an account if they do not have an account, **updated their available consultation hour**, **update timetable details**, **generate attendance code**,**manage their account** and many more. |
| **Input data flows** | Account verification, Changed Lecturer’s Account, Notice Board Information, Student details, Updated Lecturer Account Information, Student ID, Class ID, Calendar Information, Updated Consultation details, Registration link. |
| **Output data flows** | Lecturer ID and Password, New Lecturer ID and Password details, Announcement detail, Updated Lecturer details, Student name, Attendance code, Student List, Student details, Month, New Timetable details, Consultation Hour details, Lecturer’s email. |

**External Entity: Admin**

|  |  |
| --- | --- |
| **Name** | **Admin** |
| **Description** | Admin is in charge of the system, activities that admin can do while using our system included, login using their **email and password**, **manage news, manage user, manage** the **feedback** section and many more. |
| **Input data flows** | Notice Board Information, Announcement Detail, Updated User list, User details, Updated User Account Information, Calendar information, Feedback details, Updated Feedback details, Account Verification. |
| **Output data flows** | Notice Board Details, User ID, Username, Update user details, Feedback ID, Feedback reply, Admin email and password. |

**Data Store: Student**

|  |  |
| --- | --- |
| **Name** | **Student information** |
| **Description** | Stores Student’s account information and verify their information according to the input given by student. |
| **Input data flows** | Account information |
| **Output data flows** | Verified Account |
| **Data Structure** | StudentID, Password, StudentName, Address, Email, PhoneNumber, SchoolName, Gender |

**Data Store: Lecturer**

|  |  |
| --- | --- |
| **Name** | **Lecturer information** |
| **Description** | Stores Lecturer’s account information and verify their information according to the input given by Lecturer. |
| **Input data flows** | Account information |
| **Output data flows** | Verified Account |
| **Data Structure** | LecturerID, Password, LecturerName, Address, Email, PhoneNumber, SchoolName, Gender |

**Data Store: Admin**

|  |  |
| --- | --- |
| **Name** | **Admin information** |
| **Description** | Stores Admin’s account information and verify their information according to the input given by Admin. |
| **Input data flows** | Account information |
| **Output data flows** | Verified Account |
| **Data Structure** | AdminID, Password, AdminName, PhoneNumber, Email, Address, Gender |

**Data Store: Feedback**

|  |  |
| --- | --- |
| **Name** | **Feedback table** |
| **Description** | Stores the Feedback that student and admin had sent and provide the Feedback message to student / admin. |
| **Input data flows** | Feedback information, Deleted feedback |
| **Output data flows** | Feedback details, Updated Feedback details |
| **Data Structure** | StudentID, FeedbackID, AdminID, Feedback\_info, Feedback\_date, StudentName |

**Process: Manage Login**

|  |  |
| --- | --- |
| **Name** | **Manage Login** |
| **Description** | Manage Login is for the user to Login to their account and check users account with database. |
| **Input data flows** | Admin email and password, New Student's ID and Password details, Student ID and Password, Verified Account, New Lecturer's ID and Password details, Lecturer ID and Password |
| **Output data flows** | Account Verification, Account information, Changed Lecturer's Account, Changed Student's Account |
| **Process Description** | User first input their ID/email and password, after that, our system will use their input to verify with the database which is Student Information, Lecturer Information and Admin Information. If their input match with the data in the database, the user then get access to our system. If user forgot their ID or password, they could input their new ID and password and our system will updated with database. |

**Process: Manage Feedback**

|  |  |
| --- | --- |
| **Name** | **Manage Feedback** |
| **Description** | Manage Feedback allowed Students to send feedback and get replies from Admin. Other than that, it allowed Admin to Delete and Reply to Students feedback. |
| **Input data flows** | Feedback ID, Feedback reply, Feedback |
| **Output data flows** | Feedback's reply, Deleted  feedback, Feedback  information |
| **Process Description** | Once Student enter his/her feedback, the system will save it to the Feedback table in database then display it for the Admin to view. Besides that, the Admin can reply to Student’s feedback witch also saved it into the database and display it out. Admin also can delete the feedback by selecting the feedback ID and according to the ID, the system can delete it from the database table. |

**Data Flow: Admin email and Password**

|  |  |
| --- | --- |
| **Name** | **Admin email and Password** |
| **Description** | Input of Admin to verify them according to Admin Information for Admin to login their account. |
| **Origin / Source** | Admin External Entity |
| **Destination / Sink** | Manage Login process |
| **Data Structure** | Admin email, Password |

**Data Flow: Account Verification**

|  |  |
| --- | --- |
| **Name** | **Account Verification** |
| **Description** | Allowed user to login to our system. |
| **Origin / Source** | Manage Login process |
| **Destination / Sink** | Admin External Entity |
| **Data Structure** | Granted access to login |

**Data Flow: Account Information**

|  |  |
| --- | --- |
| **Name** | **Account Information** |
| **Description** | To verify Users input with database. |
| **Origin / Source** | Manage Login process |
| **Destination / Sink** | Admin Information, Student Information, Lecturer Information |
| **Data Structure** | Admin email, Student ID, Lecturer ID, Password |

**Data Flow: Verified Account**

|  |  |
| --- | --- |
| **Name** | **Verified Account** |
| **Description** | To allowed user login to our system. |
| **Origin / Source** | Admin Information, Student Information, Lecturer Information |
| **Destination / Sink** | Manage Login process |
| **Data Structure** | Granted access to login |

**Data Flow: Lecturer ID and Password**

|  |  |
| --- | --- |
| **Name** | **Lecturer ID and Password** |
| **Description** | Input of Lecturer to verify them according to Lecturer Information for Lecturer to login their account. |
| **Origin / Source** | Lecturer External Entity |
| **Destination / Sink** | Manage Login process |
| **Data Structure** | Lecturer ID, Password |

**Data Flow: Changed Lecturer’s Account**

|  |  |
| --- | --- |
| **Name** | **Changed Lecturer’s Account** |
| **Description** | New details of Lecturer’s Account after he/she had forgotten their account information and changed the login details of their account. |
| **Origin / Source** | Manage Login process |
| **Destination / Sink** | Lecturer External Entity |
| **Data Structure** | Lecturer ID, Password |

**Data Flow: New Lecturer’s ID and Password details**

|  |  |
| --- | --- |
| **Name** | **New Lecturer’s ID and Password details** |
| **Description** | Input of Lecturer’s new ID and password after clicked forget password. |
| **Origin / Source** | Lecturer External Entity |
| **Destination / Sink** | Manage Login process |
| **Data Structure** | Lecturer ID, Password |

**Data Flow: Student ID and Password**

|  |  |
| --- | --- |
| **Name** | **Student ID and Password** |
| **Description** | Input of Student to verify them according to Student Information for Student to login their account. |
| **Origin / Source** | Student External Entity |
| **Destination / Sink** | Manage Login process |
| **Data Structure** | Student ID, Password |

**Data Flow: New Student’s ID and Password details**

|  |  |
| --- | --- |
| **Name** | **New Student’s ID and Password details** |
| **Description** | Input of Student’s new ID and password after clicked forget password. |
| **Origin / Source** | Student External Entity |
| **Destination / Sink** | Manage Login process |
| **Data Structure** | Student ID, Password |

**Data Flow: Changed Student’s Account**

|  |  |
| --- | --- |
| **Name** | **Changed Student’s Account** |
| **Description** | New details of Student’s Account after he/she had forgotten their account information and changed the login details of their account. |
| **Origin / Source** | Manage Login process |
| **Destination / Sink** | Student External Entity |
| **Data Structure** | Student ID, Password |

**Data Flow: Feedback**

|  |  |
| --- | --- |
| **Name** | **Feedback** |
| **Description** | Input of student’s or lecturer’s feedback towards the system. |
| **Origin / Source** | Student External Entity |
| **Destination / Sink** | Manage Feedback process |
| **Data Structure** | StudentID, StudentName, FeedbackID, Feedback\_info, Feedback\_date, |

**Data Flow: Feedback’s reply**

|  |  |
| --- | --- |
| **Name** | **Feedback’s reply** |
| **Description** | Reply of Admin given to answer Student's or lecturer’s feedback. |
| **Origin / Source** | Manage Feedback process |
| **Destination / Sink** | Student External Entity |
| **Data Structure** | AdminID, FeedbackID, Feedback\_info, |

**Data Flow: Feedback information**

|  |  |
| --- | --- |
| **Name** | **Feedback information** |
| **Description** | Information of Student’s feedback that need to save to Feedback table. |
| **Origin / Source** | Manage Feedback process |
| **Destination / Sink** | Feedback table Data Store |
| **Data Structure** | StudentID, StudentName, FeedbackID, Feedback\_info, Feedback\_date, |

**Data Flow: Feedback details**

|  |  |
| --- | --- |
| **Name** | **Feedback details** |
| **Description** | Information of Student’s feedback that displayed from the database for Admin to view and reply. |
| **Origin / Source** | Feedback table Data Store |
| **Destination / Sink** | Admin External Entity |
| **Data Structure** | StudentName, FeedbackID, Feedback\_info, Feedback\_date, |

**Data Flow: Feedback reply**

|  |  |
| --- | --- |
| **Name** | **Feedback reply** |
| **Description** | Admin’s answer towards Student’s Feedback |
| **Origin / Source** | Admin External Entity |
| **Destination / Sink** | Manage Feedback process |
| **Data Structure** | AdminID, FeedbackID, Feedback\_info, |

**Data Flow: Feedback ID**

|  |  |
| --- | --- |
| **Name** | **Feedback ID** |
| **Description** | Selected Feedback ID for admin to find and delete from the database. |
| **Origin / Source** | Admin External Entity |
| **Destination / Sink** | Manage Feedback process |
| **Data Structure** | FeedbackID |

**Data Flow: Deleted feedback**

|  |  |
| --- | --- |
| **Name** | **Deleted feedback** |
| **Description** | Selected Feedback ID that Admin had chosen to delete from the database. |
| **Origin / Source** | Manage Feedback process |
| **Destination / Sink** | Feedback table Data Store |
| **Data Structure** | FeedbackID |

**Data Flow: Updated Feedback detail**

|  |  |
| --- | --- |
| **Name** | **Updated Feedback details** |
| **Description** | Updated feedback list after Admin deletes some feedback from the database. |
| **Origin / Source** | Feedback table Data Store |
| **Destination / Sink** | Admin External Entity |
| **Data Structure** | StudentName, FeedbackID, Feedback\_info, Feedback\_date, |

**Process: Manage News**

|  |  |
| --- | --- |
| **Name** | **Manage News** |
| **Description** | Manage News is for the Admin to key in notice board details and Lecture to key in announcement news. |
| **Input data flows** | Notice Board Details, Announcement Detail |
| **Output data flows** | Notice Board Information, Announcement Details |
| **Process Description** | Admin can first enter the notice board details to the system, then the system will send the notice bard information to admin, lecture, and student. Lecture can enter the announcement details into the system, then the system will post the announcement information at the home page of the system. |

**Data Flow: Notice Board Details**

|  |  |
| --- | --- |
| **Name** | **Notice Board Details** |
| **Description** | Allow the admin to enter the notice board details into the system |
| **Origin / Source** | Admin External Entity |
| **Destination / Sink** | Manage News Process |
| **Data Structure** | Notice Board Details |

**Data Flow: Notice Board Information**

|  |  |
| --- | --- |
| **Name** | **Notice Board Information** |
| **Description** | Send notice board information to admin, student and lecture. |
| **Origin / Source** | Manage News Process |
| **Destination / Sink** | Admin External Entity, Student External Entity, Lecture External Entity |
| **Data Structure** | Notice Board Information |

**Data Flow: Announcement Details**

|  |  |
| --- | --- |
| **Name** | **Announcement Details** |
| **Description** | Allow the lecture to enter the announcement details into the system. |
| **Origin / Source** | Lecture External Entity |
| **Destination / Sink** | Manage News Process |
| **Data Structure** | Announcement Details |

**Data Flow: Announcement Information**

|  |  |
| --- | --- |
| **Name** | **Announcement Information** |
| **Description** | Send announcement information to admin, student and lecture. |
| **Origin / Source** | Manage News Process |
| **Destination / Sink** | Admin External Entity, Student External Entity, Lecture External Entity |
| **Data Structure** | Announcement Information |

**Process: Consultation Process**

|  |  |
| --- | --- |
| **Name** | **Consultation Process** |
| **Description** | Consultation Process is for the student to book consultation with the lecture. |
| **Input data flows** | Lecture Name, Consultation Hour Details |
| **Output data flows** | Consultation Information, Updated Consultation Hour Details |
| **Process Description** | The lecture will first enter their available consultation hour details into the system, then the student can view the consultation details by searching the lecture name, when the student selected the consultation time, the system will update the lecture with the consultation hour details. |

**Data Flow: Lecture Name**

|  |  |
| --- | --- |
| **Name** | **Lecture Name** |
| **Description** | Allow the student to key in the lecture name. |
| **Origin / Source** | Student External Entity |
| **Destination / Sink** | Consultation Process |
| **Data Structure** | Lecture Name |

**Data Flow: Consultation Information**

|  |  |
| --- | --- |
| **Name** | **Consultation Information** |
| **Description** | Details of lecture’s consultation information showed to the student. |
| **Origin / Source** | Consultation Process |
| **Destination / Sink** | Student External Entity |
| **Data Structure** | Consultation Information |

**Data Flow: Consultation Hour Details**

|  |  |
| --- | --- |
| **Name** | **Consultation Hour Details** |
| **Description** | Allow the lecture to key in their available consultation hour details. |
| **Origin / Source** | Lecture External Entity |
| **Destination / Sink** | Consultation Process |
| **Data Structure** | Consultation Hour Details |

**Data Flow: Updated Consultation Hour Details**

|  |  |
| --- | --- |
| **Name** | **Updated Consultation Hour Details** |
| **Description** | Update the lecture consultation hour details after student chose consultation hour. |
| **Origin / Source** | Consultation Process |
| **Destination / Sink** | Lecture External Entity |
| **Data Structure** | Consultation Hour Details |

**Process: Manage Class List**

|  |  |
| --- | --- |
| **Name** | **Manage Class List** |
| **Description** | Manage Class List is for the lecture to check class details and Subject list information. |
| **Input data flows** | Subject name, Class name |
| **Output data flows** | Class list Details, Subject list information |
| **Process Description** | The lecture can key in the Class name to search the class details. If the lecture wishes to view the whole Subject list, the lecture can key in the subject name to search the subject list list information from the database. |

**Data Flow: Subject Name**

|  |  |
| --- | --- |
| **Name** | **Subject Name** |
| **Description** | Lecture input Subject name into the system to search the Subject details. |
| **Origin / Source** | Lecture External Entity |
| **Destination / Sink** | Manage Subject list Process |
| **Data Structure** | Subject Name |

**Data Flow: Class Name**

|  |  |
| --- | --- |
| **Name** | **Class Name** |
| **Description** | Lecture input class name into the system to search the class list information. |
| **Origin / Source** | Lecture External Entity |
| **Destination / Sink** | Manage Class List Process |
| **Data Structure** | Class name |

**Data Flow: Subject List Information**

|  |  |
| --- | --- |
| **Name** | **Subject List Information** |
| **Description** | Allow the lecture to view the Subject list. |
| **Origin / Source** | Manage Subject List Process |
| **Destination / Sink** | Lecture External Entity |
| **Data Structure** | Subject List |

**Data Flow: Class list details**

|  |  |
| --- | --- |
| **Name** | **Class list Details** |
| **Description** | Allow the lecture to view the class details. |
| **Origin / Source** | Manage class List Process |
| **Destination / Sink** | Lecture External Entity |
| **Data Structure** | Class Details |

**Process: Generate code**

|  |  |
| --- | --- |
| **Name** | **4.0 Generate code** |
| **Description** | Generate code is allow lecturer to generate a new code, the student can able take attendance using the code that generated. |
| **Input data flows** | Attendance code |
| **Output data flows** | Attendance record |
| **Process Description** | When lecturer enter a class name to generate a code for student to take attendance. Once student take the attendance using code, the attendance record will be saved into database. |

**Data Flow: Attendance code**

|  |  |
| --- | --- |
| **Name** | **Attendance code** |
| **Description** | Allow lecture generate code randomly |
| **Origin / Source** | Lecture External Entity |
| **Destination / Sink** | Generate code |
| **Data Structure** | Attendance code |

**Data Flow: Attendance record**

|  |  |
| --- | --- |
| **Name** | **Attendance record** |
| **Description** | Allow saved the attendance record after student take attendance with code. |
| **Origin / Source** | Student External Entity |
| **Destination / Sink** | Generate Code |
| **Data Structure** | Attendance ID |

**Process: Timetable process**

|  |  |
| --- | --- |
| **Name** | **Timetable Process** |
| **Description** | Timetable process is allowed student, admin to view the timetable details and lecturer are able to update timetable details. |
| **Input data flows** | New Timetable Details, Class name, holiday details |
| **Output data flows** | Timetable information, Updated Timetable details |
| **Process Description** | When student and admin insert classname, they are allowed to view the timetable details. Lecturer can be able to add timetable to certain class and update the new timetable details. |

**Data Flow:  New Timetable Details**

|  |  |
| --- | --- |
| **Name** | **New Timetable Details** |
| **Description** | Allow lecturer insert new timetable details. |
| **Origin / Source** | Lecturer external entity |
| **Destination / Sink** | Timetable process |
| **Data Structure** | Timetable ID |

**Data Flow:  Class name**

|  |  |
| --- | --- |
| **Name** | **Classname** |
| **Description** | Student, lecturer and admin need insert certain class name to view the suitable timetable |
| **Origin / Source** | Student external entity, lecturer external entity, admin external entity |
| **Destination / Sink** | Timetable process |
| **Data Structure** | Timetable ID |

**Data Flow: Timetable information**

|  |  |
| --- | --- |
| **Name** | **Timetable information** |
| **Description** | Student, lecturer, and admin able to view Timetable information which following class name. |
| **Origin / Source** | Timetable process |
| **Destination / Sink** | Student external entity, lecturer external entity, admin external entity |
| **Data Structure** | Timetable ID |

**Data Flow:  Updated Timetable Details**

|  |  |
| --- | --- |
| **Name** | **Updated timetable details** |
| **Description** | Student and admin able to view the updated timetable, which added by lecturer. |
| **Origin / Source** | Timetable process |
| **Destination / Sink** | Student external entity |
| **Data Structure** | Timetable ID |

**Process: Holiday Details process**

|  |  |
| --- | --- |
| **Name** | **Holiday Details Process** |
| **Description** | Holiday Details process is allowed admin to insert the holiday start date, and end date, if student and lecturer need to know, which day is public holiday, they can click to see the holiday details. |
| **Input data flows** | Holiday details |
| **Output data flows** | Updated Holiday details |
| **Process Description** | When student and lecturer click view holiday details button, they are able to view upcoming holiday. Admin able to update the holiday details into the system |

**Data Flow: Holiday Details**

|  |  |
| --- | --- |
| **Name** | **Holiday Details** |
| **Description** | Student, lecturer and admin able to view Timetable information which click view holiday details button |
| **Origin / Source** | Holiday Details process |
| **Destination / Sink** | Admin external entity |
| **Data Structure** | Calendar ID |

**Data Flow:  Updated Holiday Details**

|  |  |
| --- | --- |
| **Name** | **Updated Holiday details** |
| **Description** | Student, lecturer, admin able to view the updated holiday details, which added by admin. |
| **Origin / Source** | Timetable process |
| **Destination / Sink** | Student external entity, lecturer external entity, admin external entity |
| **Data Structure** | Calendar ID |

**Process: Register Account**

|  |  |
| --- | --- |
| **Name** | **Register Account** |
| **Description** | Register Account is function for admin insert lecturer and student information into database. |
| **Input Data Flow** | Student Account Information, Lecturer Account Information |
| **Output Data Flow** | New Student Account, New Lecturer Account |
| **Process Description** | Admin will insert all student information and lecturer information into system which follow the list that school giving to us. |

**Data Flow: Student Account Information**

|  |  |
| --- | --- |
| **Name** | **Student Account Information** |
| **Description** | Allow admin insert student account information for registration |
| **Origin / Source** | Student External Entity |
| **Destination / Sink** | Register Account Process |
| **Data Structure** | Student ID, Student Name, Student information |

**Data Flow: New Student Account**

|  |  |
| --- | --- |
| **Name** | **New Student Account** |
| **Description** | A new student account is given towards the student by the system |
| **Origin / Source** | Register Account Process |
| **Destination / Sink** | Student External Entity |
| **Data Structure** | New Student Account |

**Data Flow: Lecturer Account Information**

|  |  |
| --- | --- |
| **Name** | **Lecturer Account Information** |
| **Description** | Allow admin insert student account information for registration |
| **Origin / Source** | Lecturer External Entity |
| **Destination / Sink** | Register Account Process |
| **Data Structure** | Lecturer ID, Lecturer Name, Lecturer information |

**Data Flow: New Lecturer Account**

|  |  |
| --- | --- |
| **Name** | **New Lecturer Account** |
| **Description** | A new lecturer account is given towards the lecturer by the system |
| **Origin / Source** | Register Account Process |
| **Destination / Sink** | Lecturer External Entity |
| **Data Structure** | New Lecturer Account |

**Data Flow: Registered Account**

|  |  |
| --- | --- |
| **Name** | **Registered Account** |
| **Description** | User accounts get an account for using the attendance system. |
| **Origin / Source** | Register Account Process |
| **Destination / Sink** | Admin External Entity |
| **Data Structure** | Registered Account |

**Process: Manage User**

|  |  |
| --- | --- |
| **Name** | **Manage User** |
| **Description** | Allow user to update, search and delete account. |
| **Input Data Flow** | Updated Student Details, Lecturer Name, Updated Lecturer Details, Student Name, Updated User Details, Username, User ID |
| **Output Data Flow** | Updated Student Account Information, Lecturer Details, Updated Lecturer Account Information, Student Details, Updated User Account Information, User Details, Updated User List |
| **Process Description** | User first enter their new account information; our system will then send their updated information towards admin to update their account information. |

**Data Flow: Updated Student Details**

|  |  |
| --- | --- |
| **Name** | **Updated Student Details** |
| **Description** | Allow student to update their account details |
| **Origin / Source** | Student External Entity |
| **Destination / Sink** | Manage User Process |
| **Data Structure** | Address, Email, Phone Number |

**Data Flow: Lecturer Name**

|  |  |
| --- | --- |
| **Name** | **Lecturer Name** |
| **Description** | Allow student to search for lecturer details |
| **Origin / Source** | Student External Entity |
| **Destination / Sink** | Manage User Process |
| **Data Structure** | Lecturer Name |

**Data Flow: Updated Student Account Details**

|  |  |
| --- | --- |
| **Name** | **Updated Student Account Details** |
| **Description** | Update student details after admin updated student account details in database |
| **Origin / Source** | Manage User Process |
| **Destination / Sink** | Student External Entity |
| **Data Structure** | Address, Email, Phone Number |

**Data Flow: Lecturer Details**

|  |  |
| --- | --- |
| **Name** | **Lecturer Details** |
| **Description** | Details of lecturer shown to student |
| **Origin / Source** | Manage User Process |
| **Destination / Sink** | Student External Entity |
| **Data Structure** | Lecturer ID, Lecturer Name, Address, Email, Phone Number, School Name, Gender |

**Data Flow: Updated Lecturer Details**

|  |  |
| --- | --- |
| **Name** | **Updated Lecturer Details** |
| **Description** | Allow lecturer to update their account details |
| **Origin / Source** | Lecturer External Entity |
| **Destination / Sink** | Manage User Process |
| **Data Structure** | Address, Email, Phone Number |

**Data Flow: Student Name**

|  |  |
| --- | --- |
| **Name** | **Student Name** |
| **Description** | Allow lecturer to search for student’s details |
| **Origin / Source** | Lecturer External Entity |
| **Destination / Sink** | Manage User Process |
| **Data Structure** | Student Name |

**Data Flow: Updated Lecturer Account Information**

|  |  |
| --- | --- |
| **Name** | **Updated Lecturer Account Information** |
| **Description** | Update lecturer details after admin updated lecturer details in database |
| **Origin / Source** | Manage User Process |
| **Destination / Sink** | Lecturer External Entity |
| **Data Structure** | Address, Email, Phone Number |

**Data Flow: Student Details**

|  |  |
| --- | --- |
| **Name** | **Student Details** |
| **Description** | Details of student shown to lecturer |
| **Origin / Source** | Manage User Process |
| **Destination / Sink** | Lecturer External Entity |
| **Data Structure** | Student ID, Student Name, Address, Email, Phone Number, School Name, Gender |

**Data Flow: Updated User Details**

|  |  |
| --- | --- |
| **Name** | **Updated User Details** |
| **Description** | Allow admin to update user’s details |
| **Origin / Source** | Admin External Entity |
| **Destination / Sink** | Manage User Process |
| **Data Structure** | Address, Email, Phone Number |

**Data Flow: Username**

|  |  |
| --- | --- |
| **Name** | **Username** |
| **Description** | Allow admin to search for user’ details |
| **Origin / Source** | Admin External Entity |
| **Destination / Sink** | Manage User Process |
| **Data Structure** | Username |

**Data Flow: User ID**

|  |  |
| --- | --- |
| **Name** | **User ID** |
| **Description** | Allow admin to delete user’s account |
| **Origin / Source** | Admin External Entity |
| **Destination / Sink** | Manage User Process |
| **Data Structure** | User ID |

**Data Flow: Updated User Account Information**

|  |  |
| --- | --- |
| **Name** | **Updated User Account Information** |
| **Description** | Update user’s details after admin update user’s details in database |
| **Origin / Source** | Manage User Process |
| **Destination / Sink** | Admin External Entity |
| **Data Structure** | Address, Email, Phone Number |

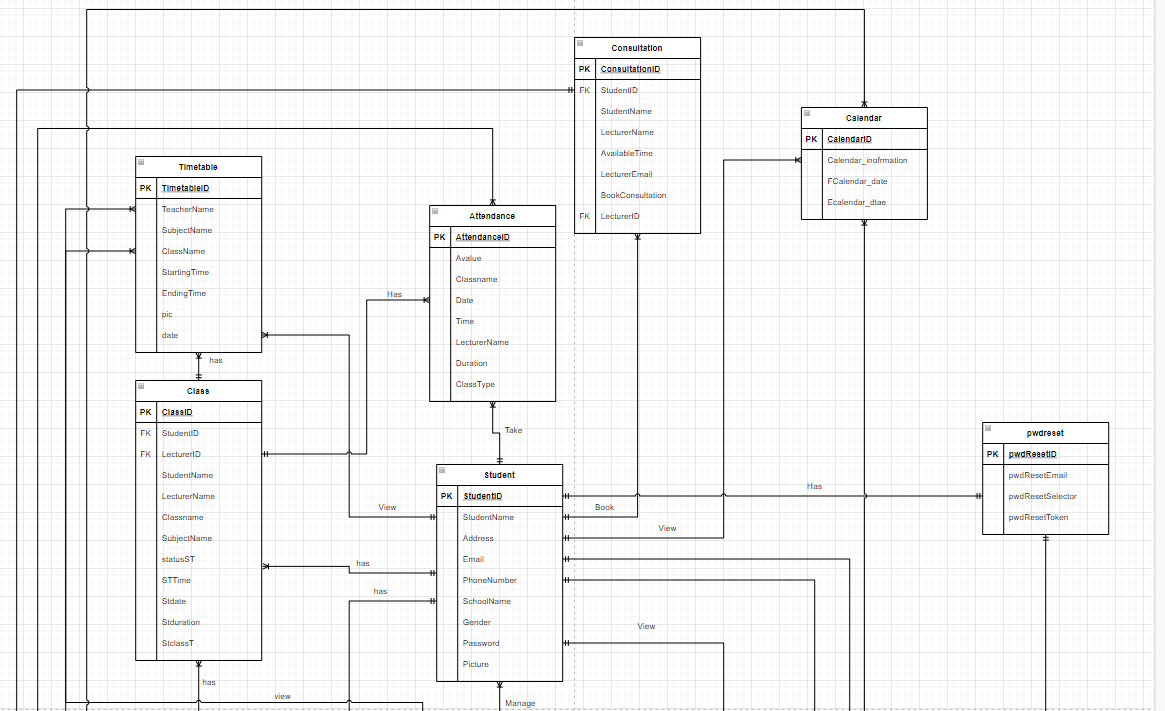
**Data Flow: User Details**

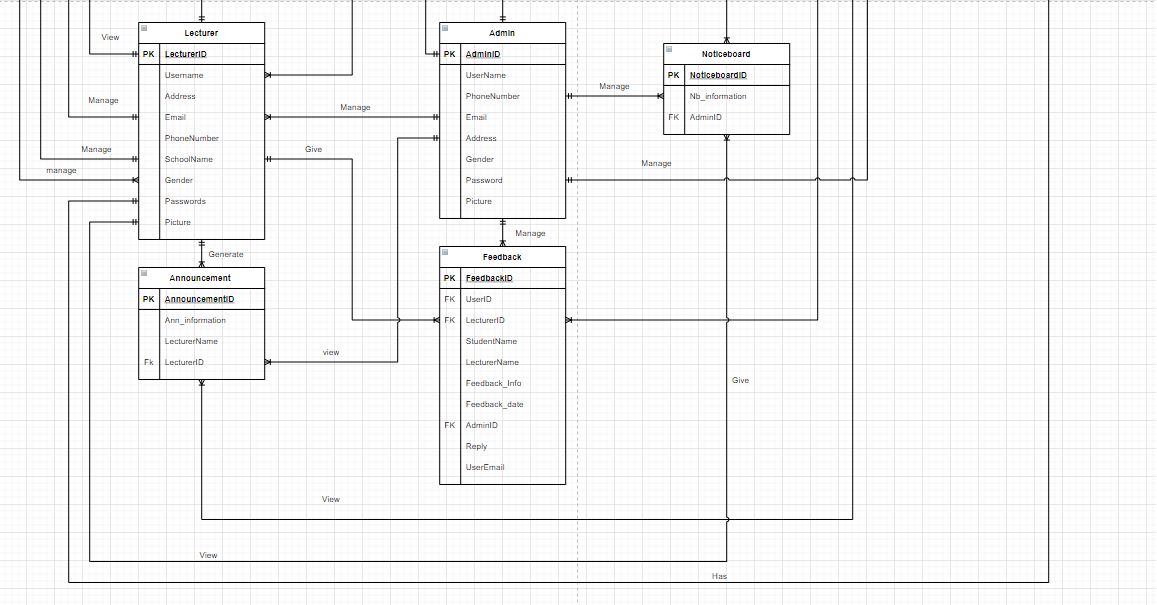
|  |  |
| --- | --- |
| **Name** | **User Details** |
| **Description** | Details of user shown to admin |
| **Origin / Source** | Manage User Process |
| **Destination / Sink** | Admin External Entity |
| **Data Structure** | User ID, Username, Address, Email, Phone Number, School Name, Gender |

**Data Flow: Updated User List**

|  |  |
| --- | --- |
| **Name** | **Updated User List** |
| **Description** | Update user list after user’s account has been deleted from database |
| **Origin / Source** | Manage User Process |
| **Destination / Sink** | Admin External Entity |
| **Data Structure** | User ID |

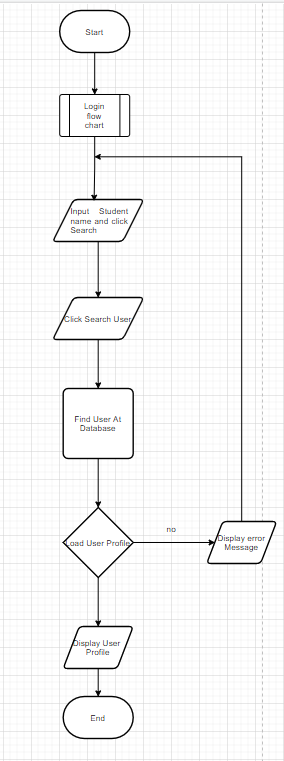
# **Entity Relationship Diagram (ERD)**



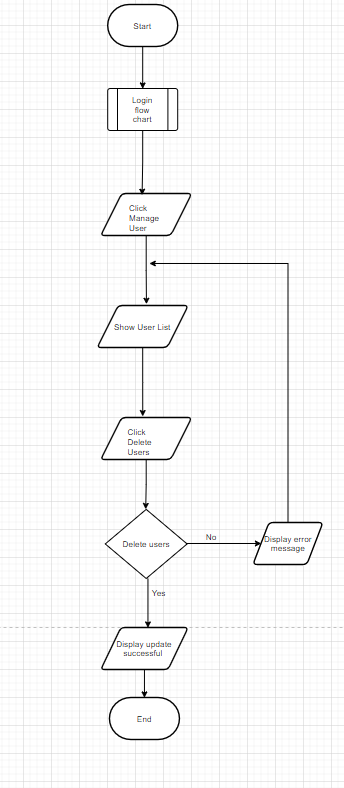


# **Flowcharts**

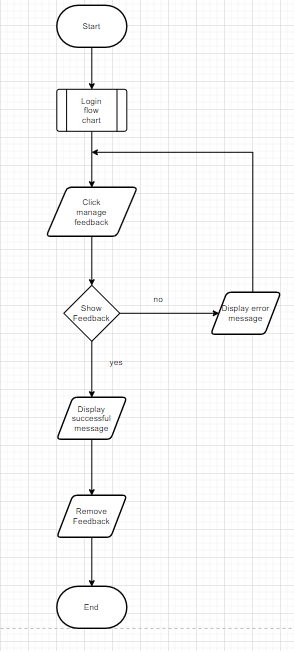
**Search User**



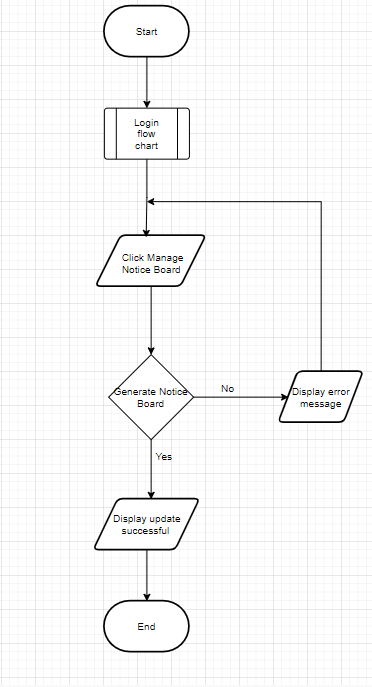
**Delete User**



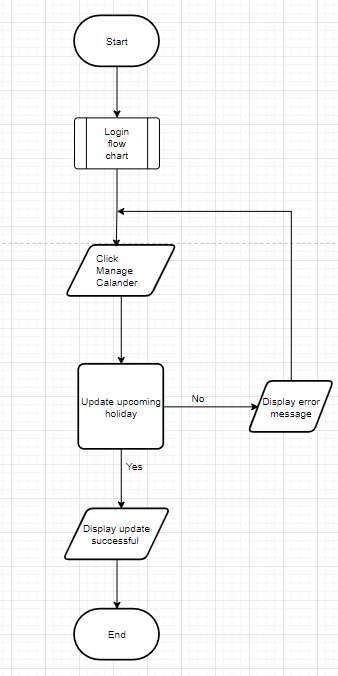
**Show Feedback**



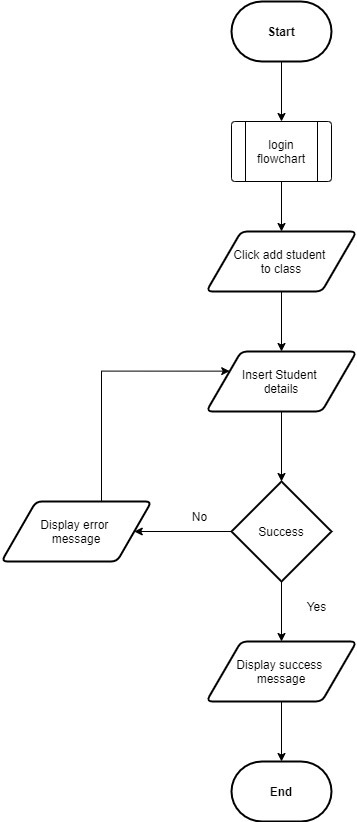
**Manage Notice Board**



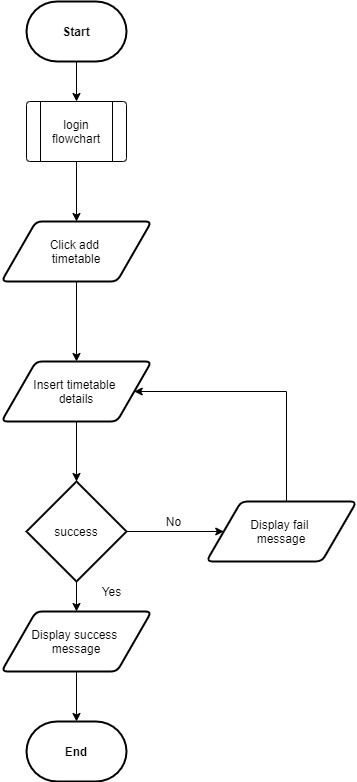
**Manage Calendar**



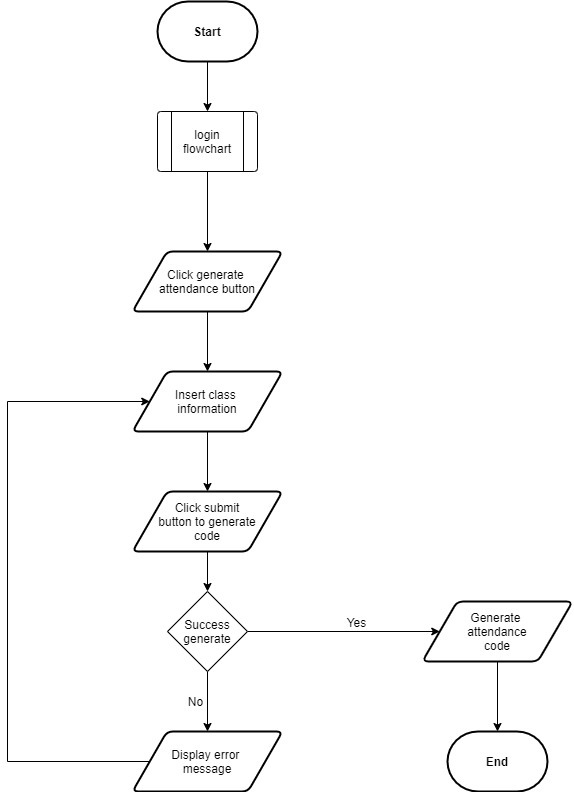
**Add Student into Class**



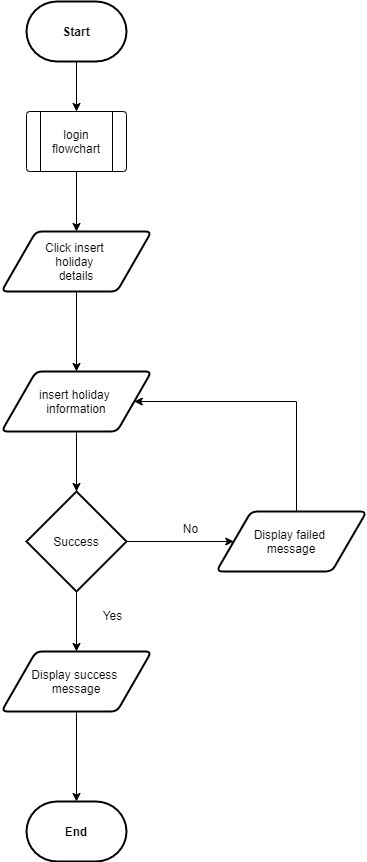
**Add Timetable Process**



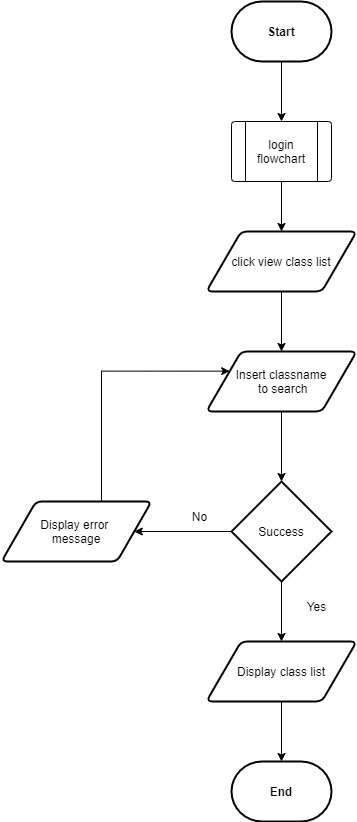
**Generate Attendance Code**



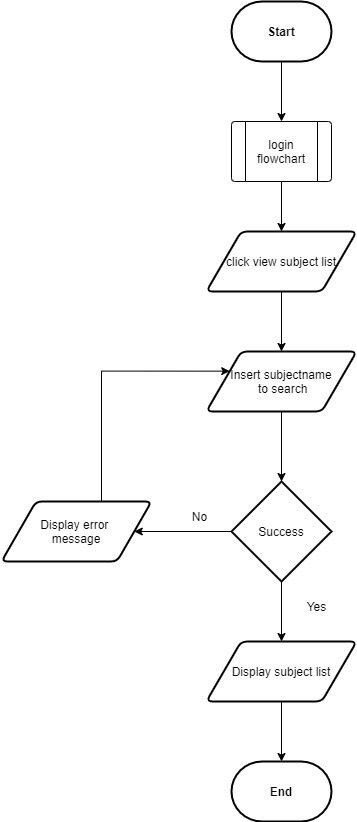
**Insert Holiday Details**



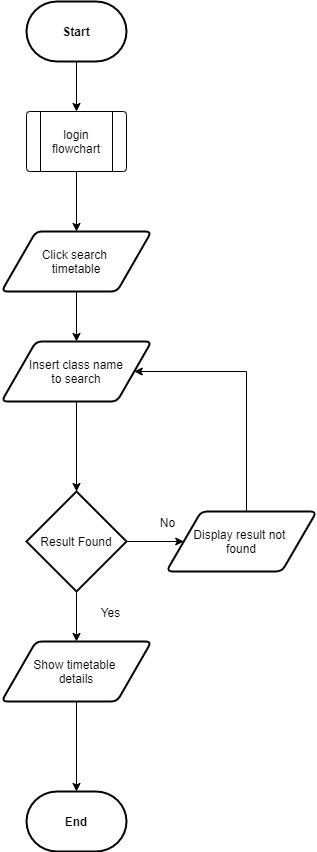
**Search Class List**



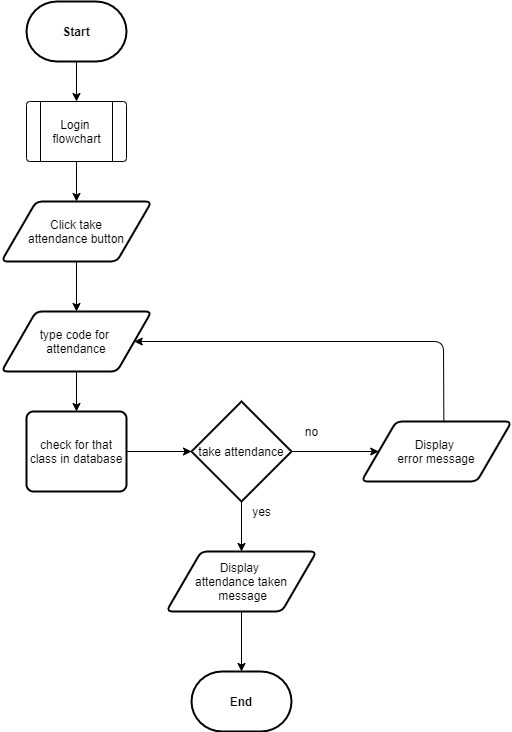
**Search Subject List**



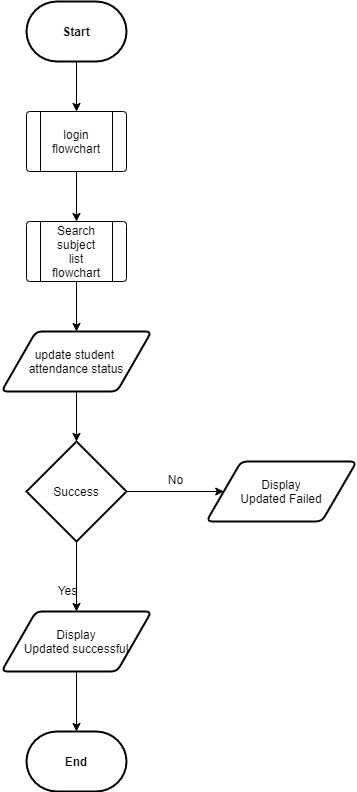
**Search Timetable**



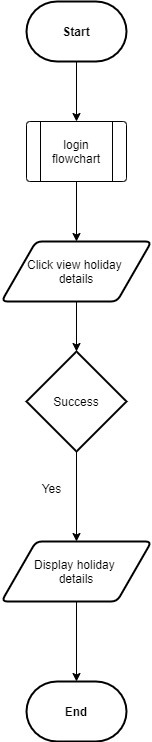
**Student Take Attendance**



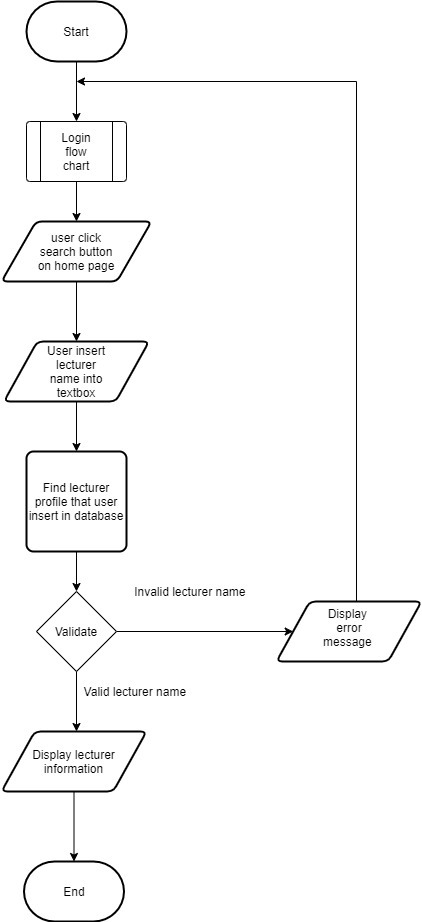
**Update Attendance Information**



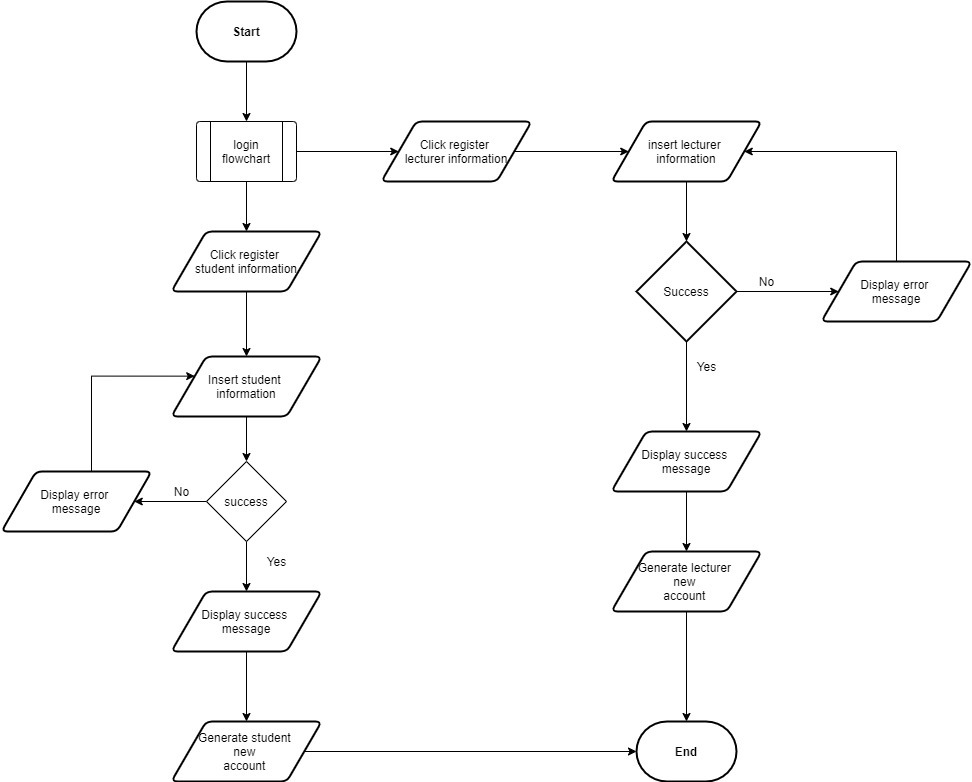
**View Holiday Details**



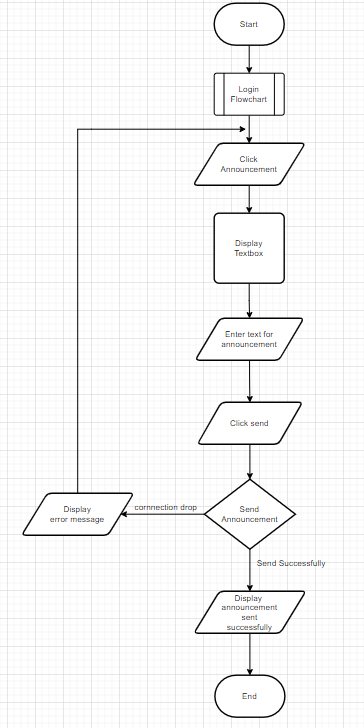
**Search Lecture**



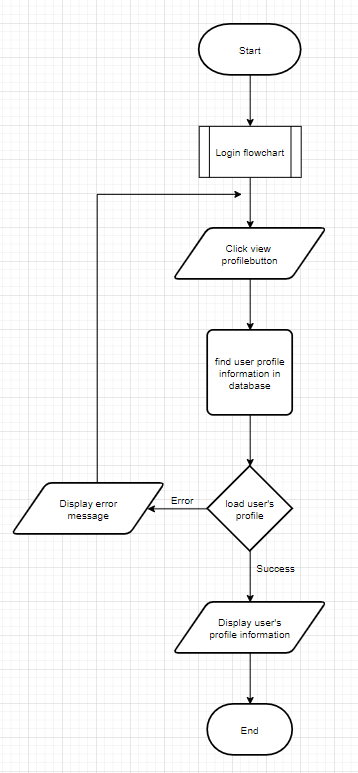
**Registration Function**



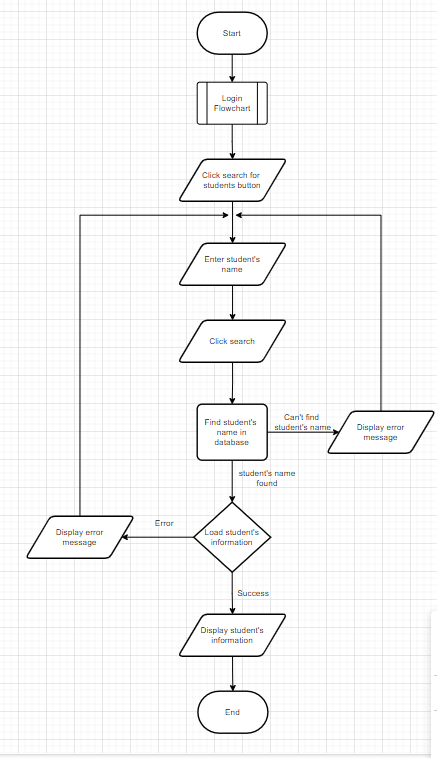
**Lecture Send Announcement**



**Lecture Personal Information**



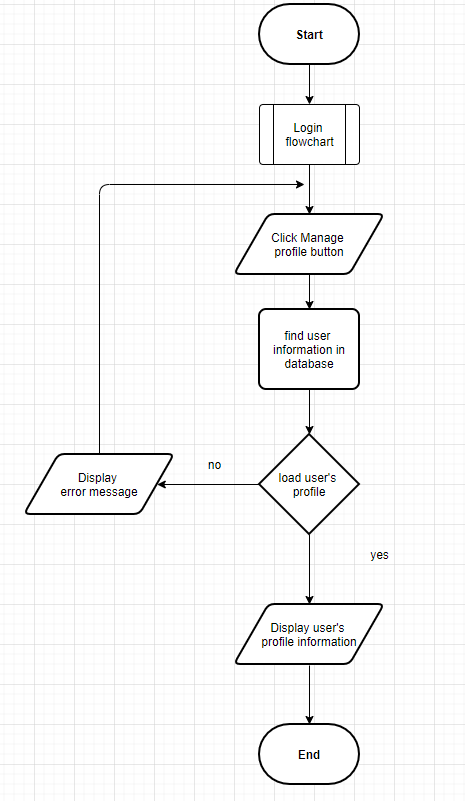
**Search Student**



**Login**



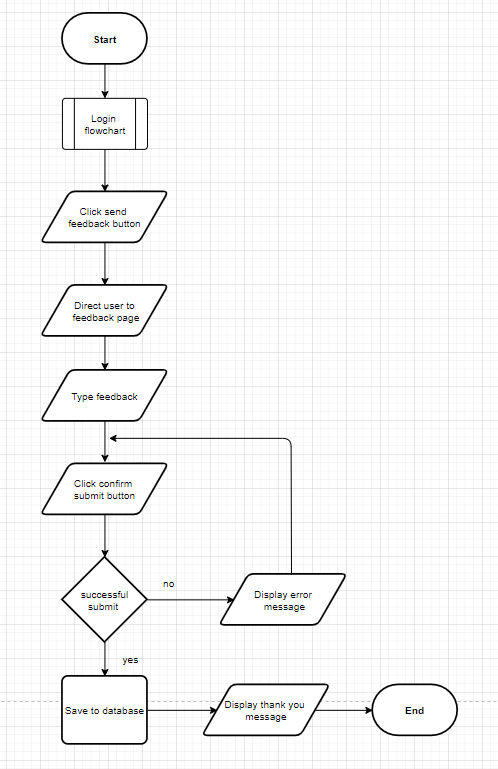
**Student view profile**



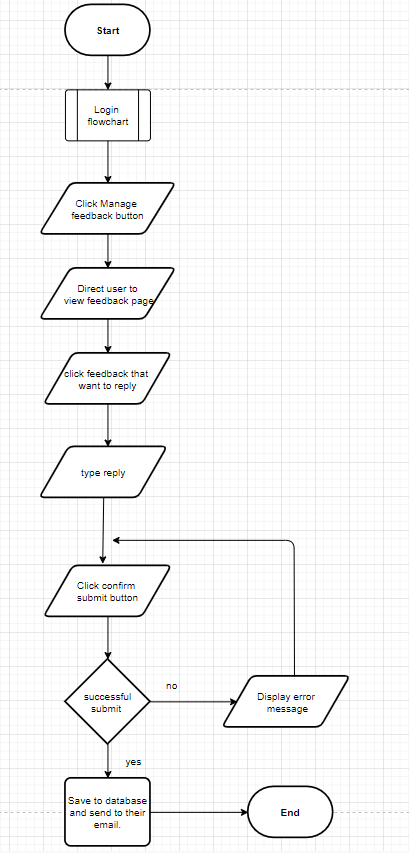
**Student update profile**



**Student send feedback**



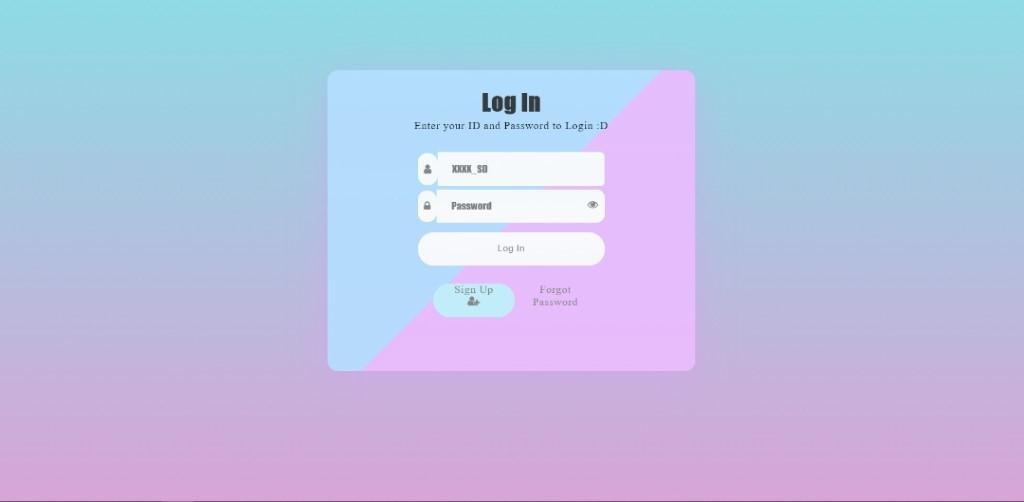
**Admin reply feedback**



# **Screen Design and User Manual**



This is the homepage of our website. User are required to login to access any function of the website has to offer. The search bar will also redirect user to login page if clicked. New user to the website can click on “sign up” at the top left to go to register page to create a new account to login.



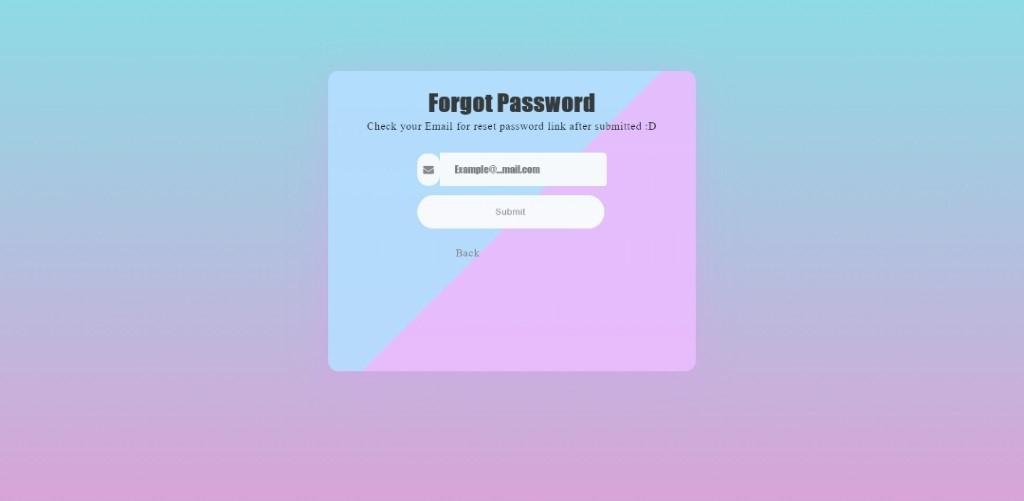
This is the login page of our website. User are required to enter their ID and password in order to login and have access to the website main feature. The “eye” icon at the password will display user’s password after they type it in as their password will be covered as they type in safety purposes. User can also use the “sign up” to go to the register account page to register new account. If user forget their password, they can click on the “Forgot Password” to head to forget password page to reset their password.



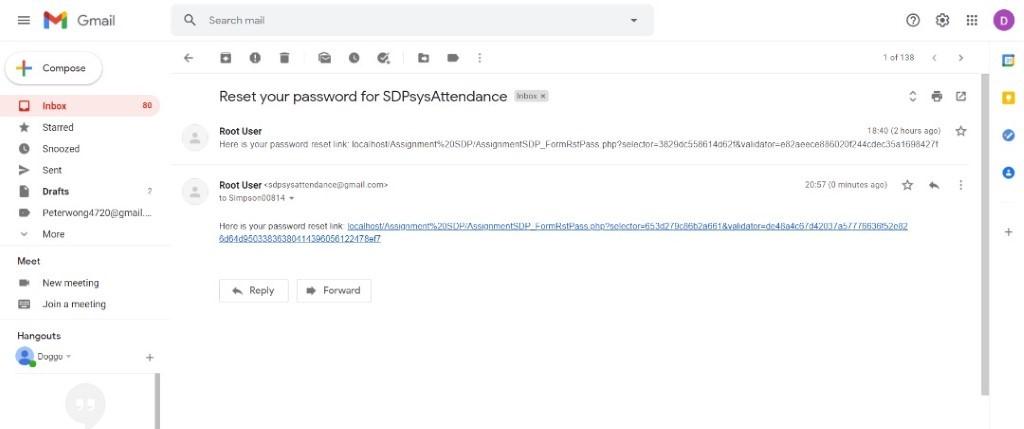
This error message showed up if the user tried to sign in using wrong ID or password.



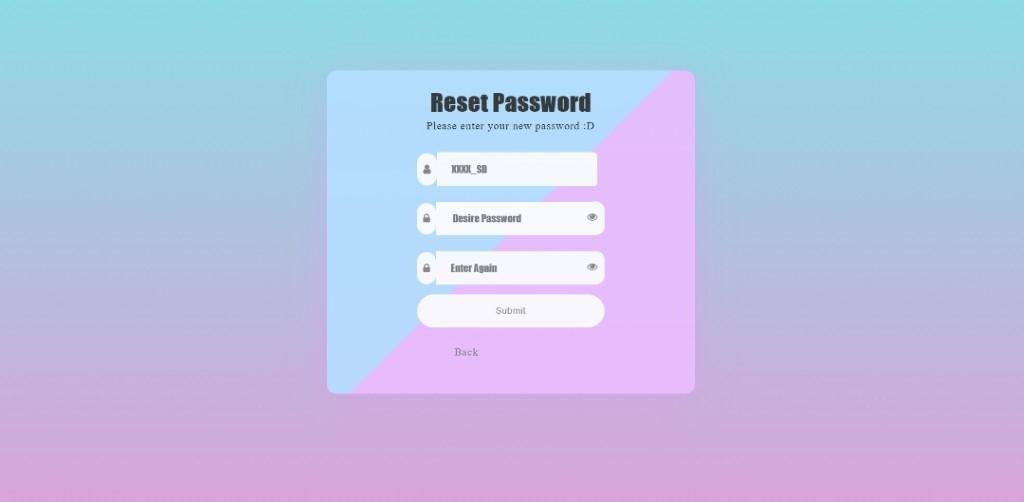
This message will show up if the user login successfully.



This is the forgot password page. User are required to type in their Email and then submit it. Our system will then proceed to send an email to that entered email entered by the user which will include a link that allow them to reset their password.



Above is an example of a password reset link included in the Email sent to the user. The link will redirect user to the reset password page which allow them to reset their password.



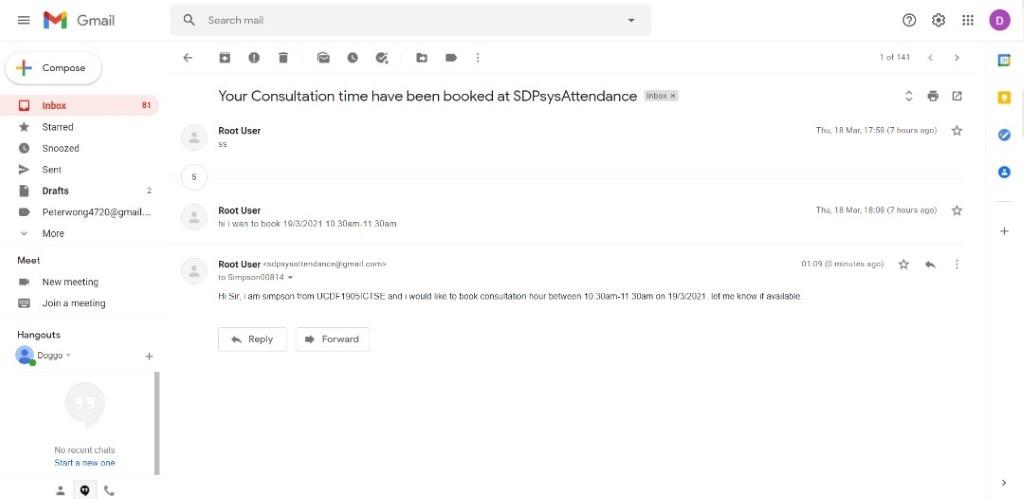
This is the reset password page. User are required to enter their ID and new password twice and submit it. Our system will proceed to check our database for the user’s ID and Email. If matching result of user’s ID and Email was found in our database, our system will then only proceed to reset their password.



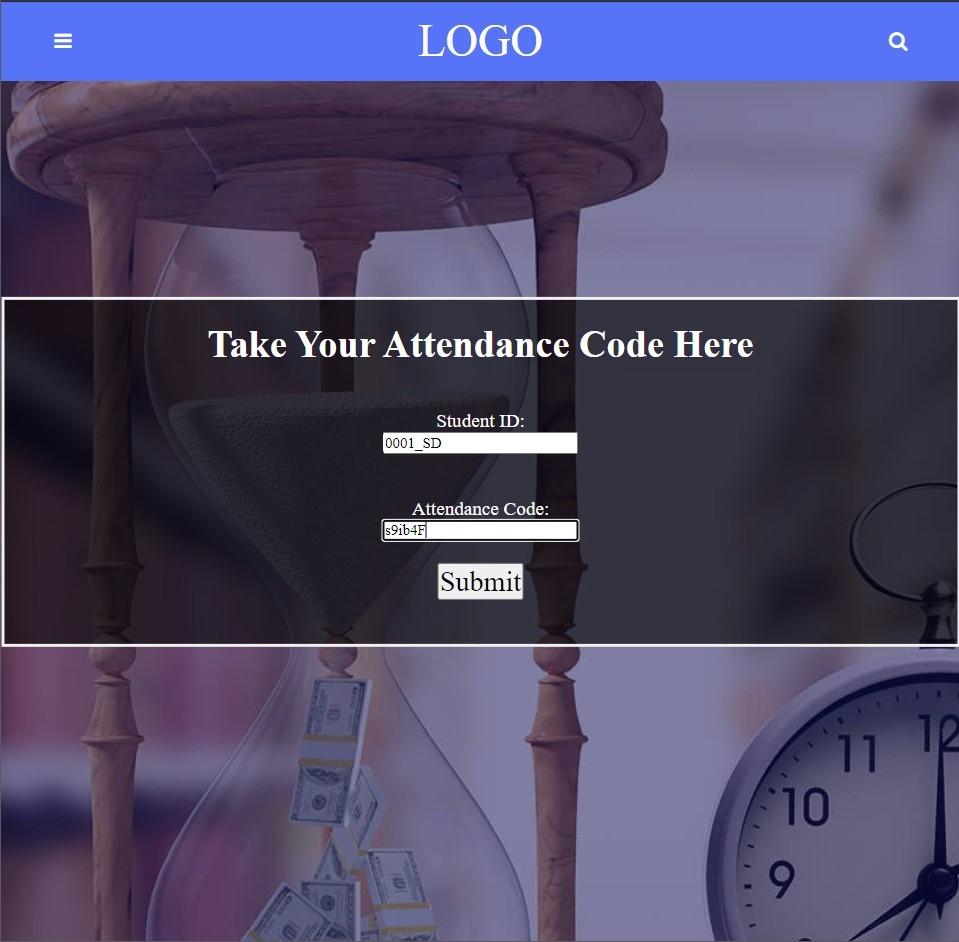
This error message would show up if the user entered two different passwords during resetting password.



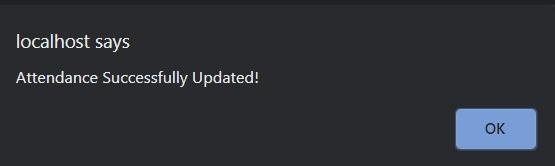
This message will show up after user password was successfully changed. If the user is a student, it will show up as “Student password updated”, lecturer will show up as “Lecturer password updated” and admin will show up as “Admin password updated”. The system will then redirect the user back to login page after password updated successfully.



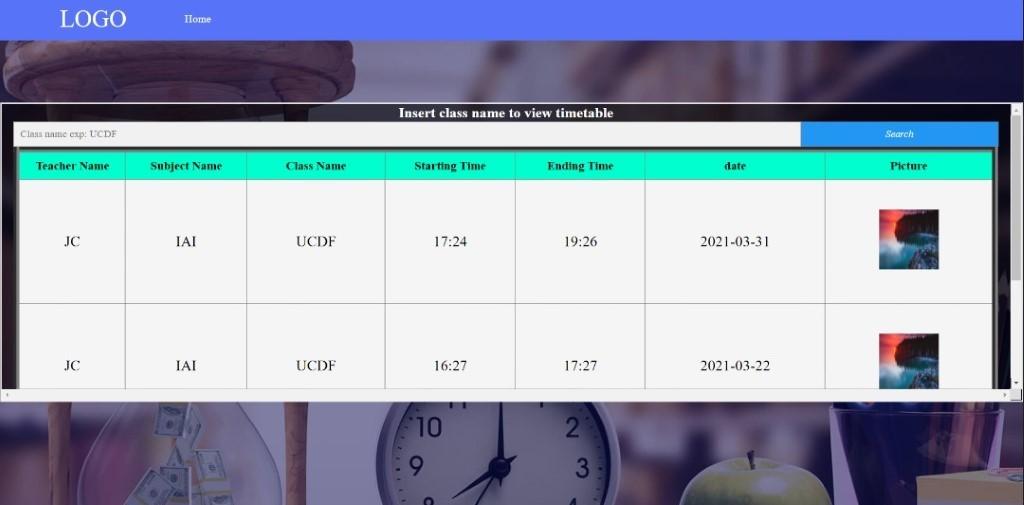
Here is an example of the email the lecturer will receive.



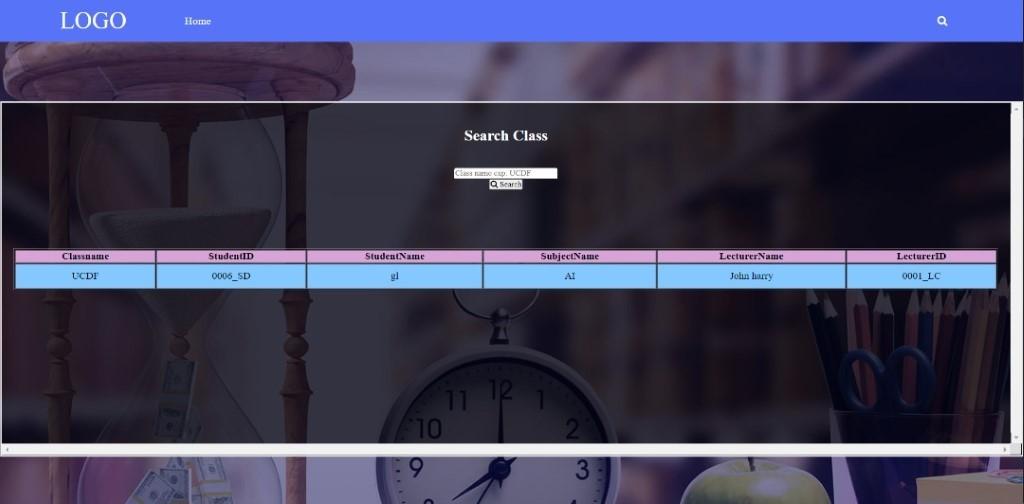
This is the take attendance page. To take the attendance, student is required to enter the required information and click on the “submit” button to submit their attendance record into the database.



This message will show up if the attendance is updated successfully at database.



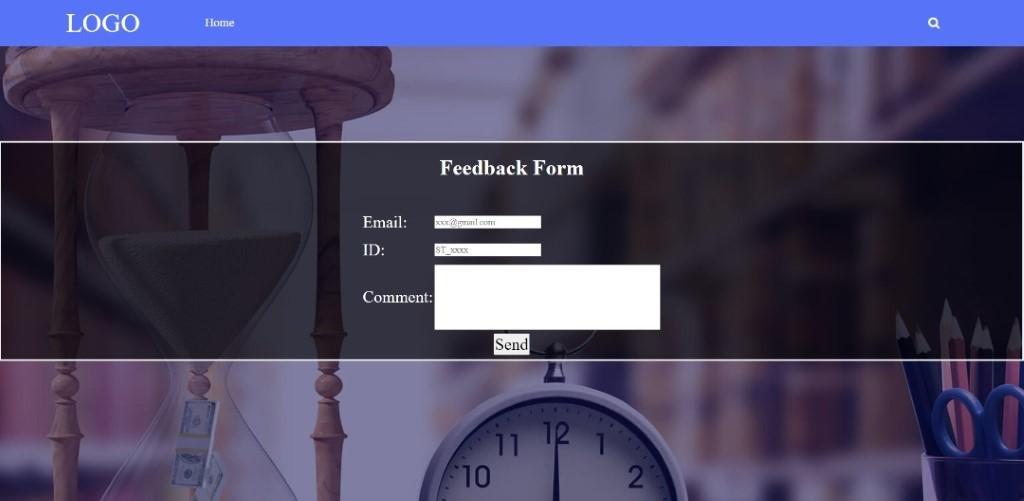
This is the view timetable page. Student will be able to view the timetable by searching for it with the class name.



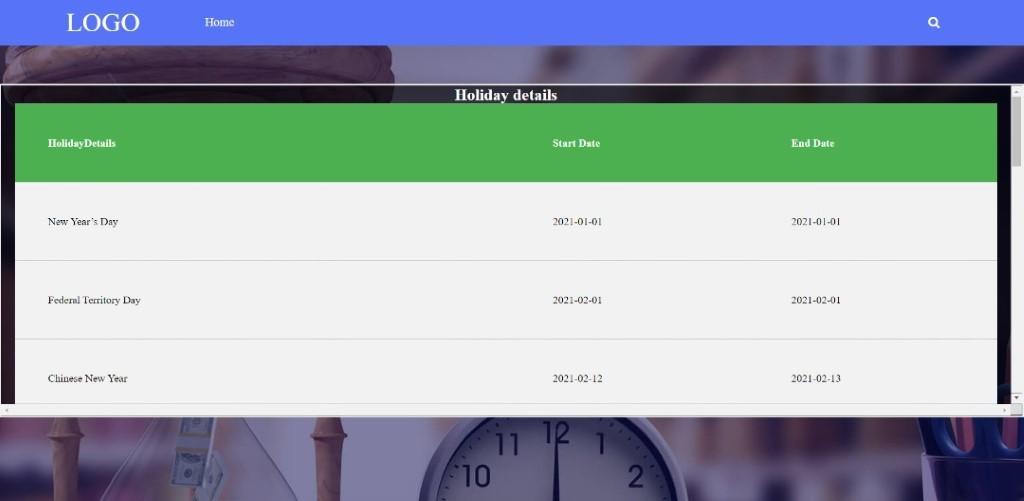
This is the view class list page. Student will be able to view information about their class by searching with the class name.



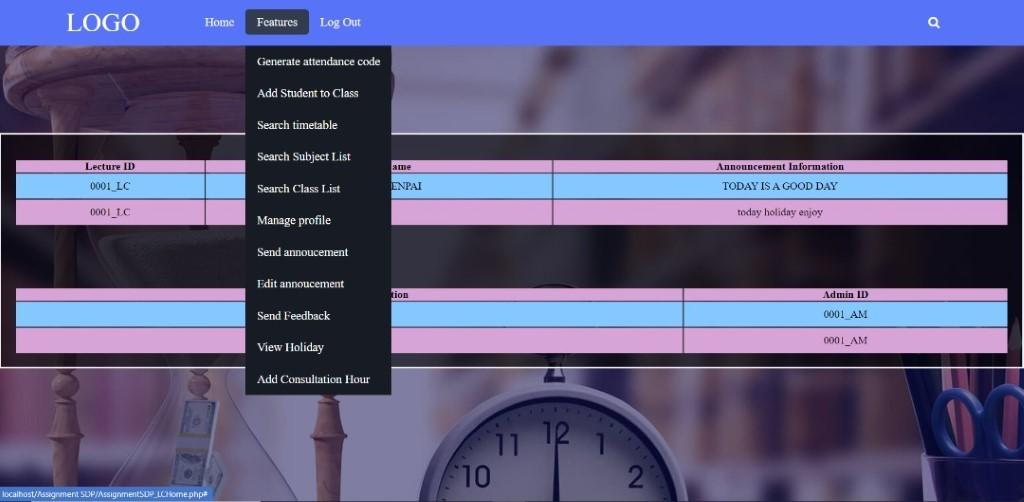
Student will be redirected to this manage profile page if they press on “Manage Profile” from features in the homepage. Students are able to upload new profile picture and change their personal information. The only thing the student are not allowed to change is the student ID as every student will have their very own student ID given to them as they register on the website and it will stay with them until they graduate. After the student has finish changing their personal information, they can click on the “Update” button to update their information.



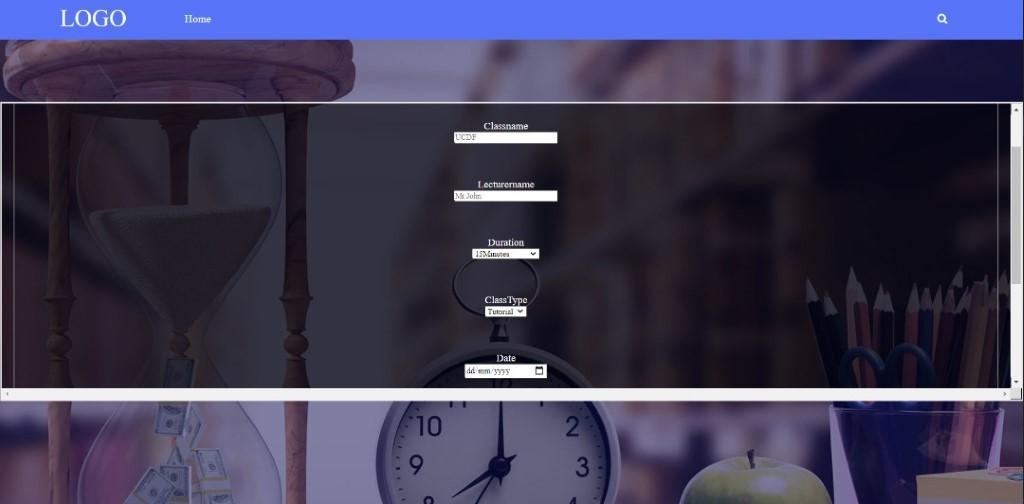
Student will be redirected to this send feedback form if they press on the “Send Feedback” on the feature at homepage. Student have to enter their Email, student ID and comment to submit the feedback.

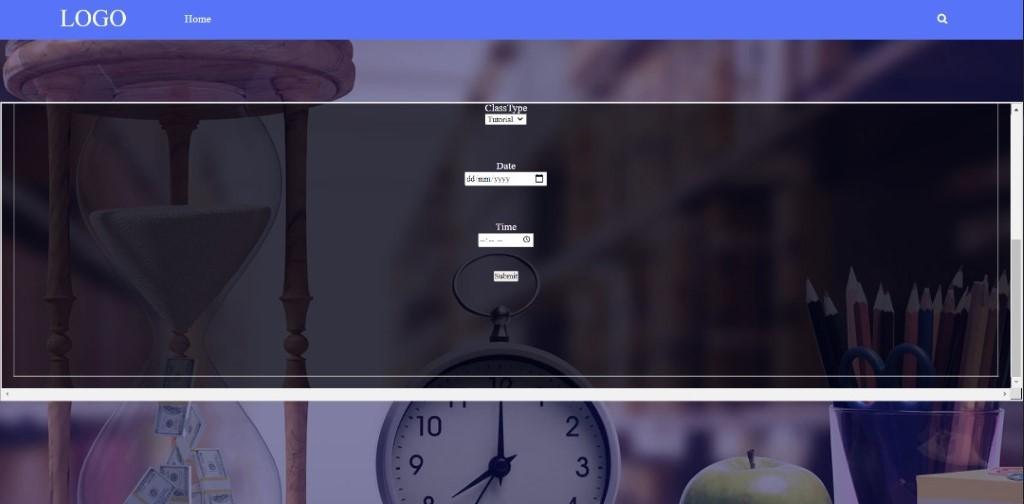


This is the check Holiday page. Student are able to view “Holiday Details” such as the start date and the end date of the Holiday if they press on the view holiday on the feature at the homepage.

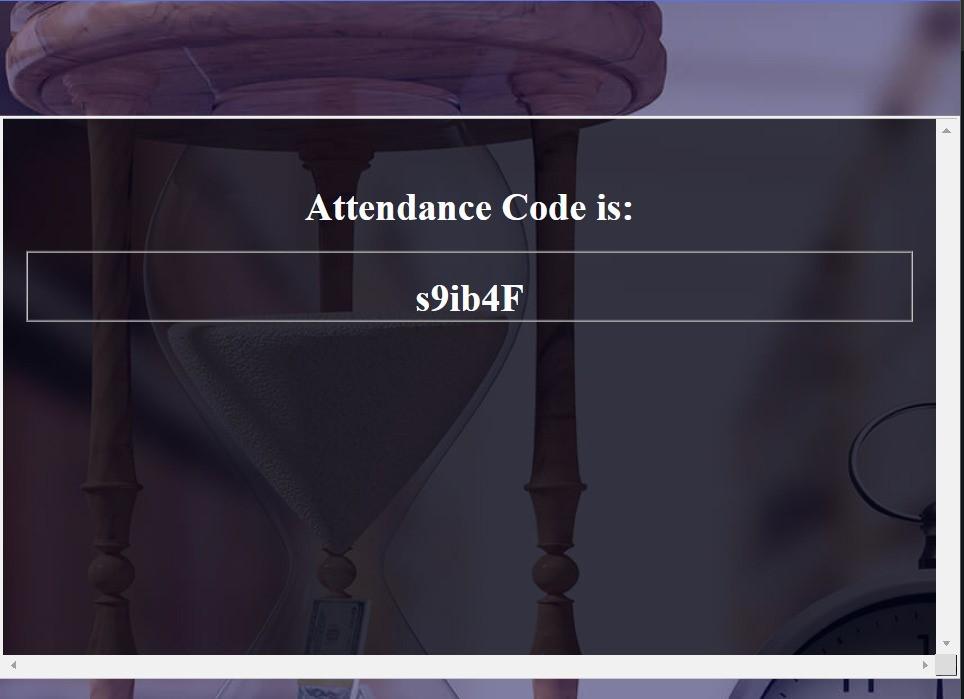


This is the homepage after the user login as a lecturer. The middle section of the website will display any announcement post by the lecturer and any notices releases by the website admin. At the top of the website, there will be a few buttons available for the lecturer. The home button will simply refresh the page. The features button when click will show all the functions available at the website for lecturer. Lecturer can click on the logout button to exit the website.

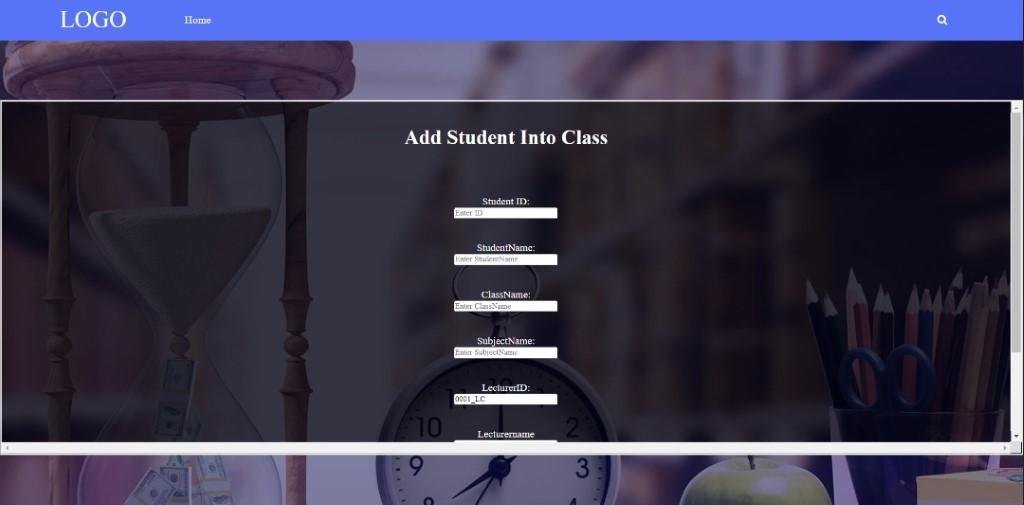


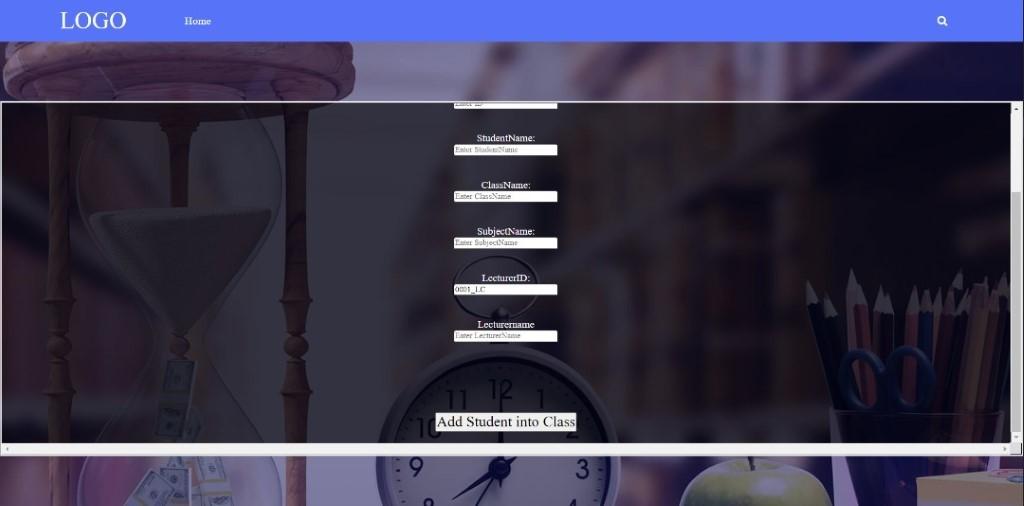


This is the generate attendance code page. Lecturer are required to fill in necessary information and submit it to generate attendance code.

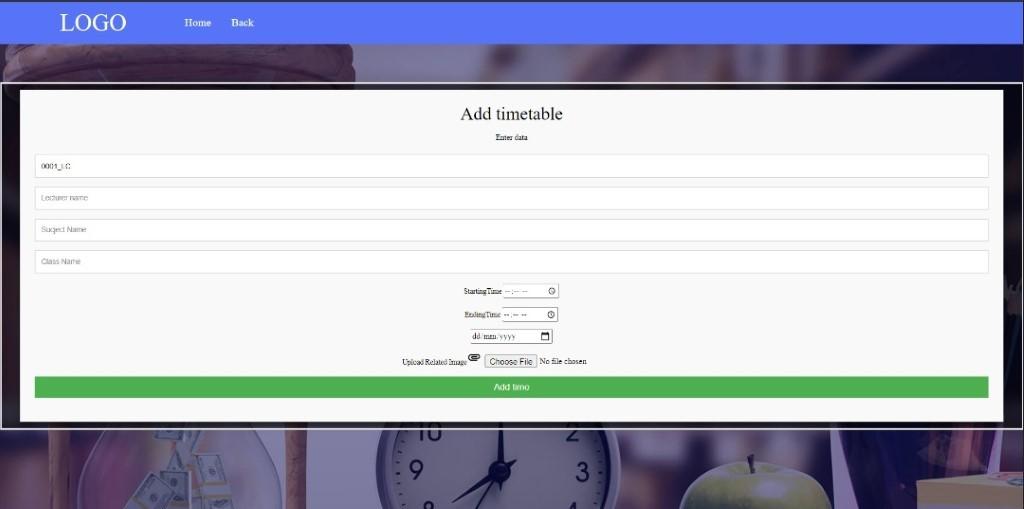


The attendance code will be generated for the student to take their class attendance.

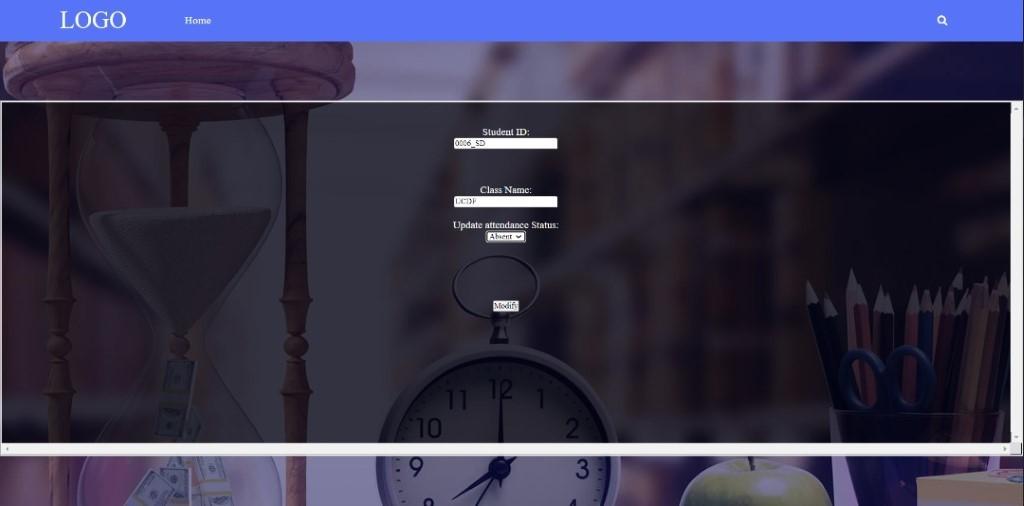


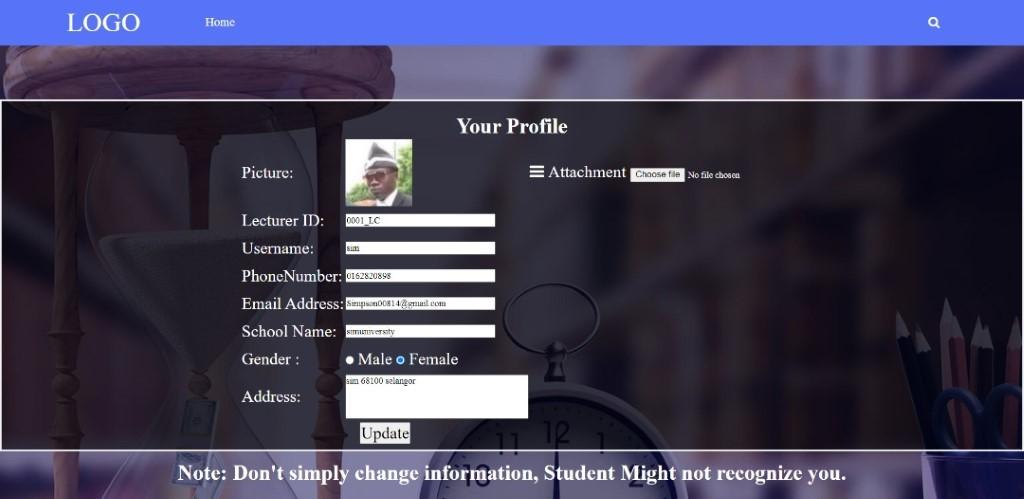


This is the page where lecturer can add student into each class. Lecturer are required to enter all the information in order to add student into class.

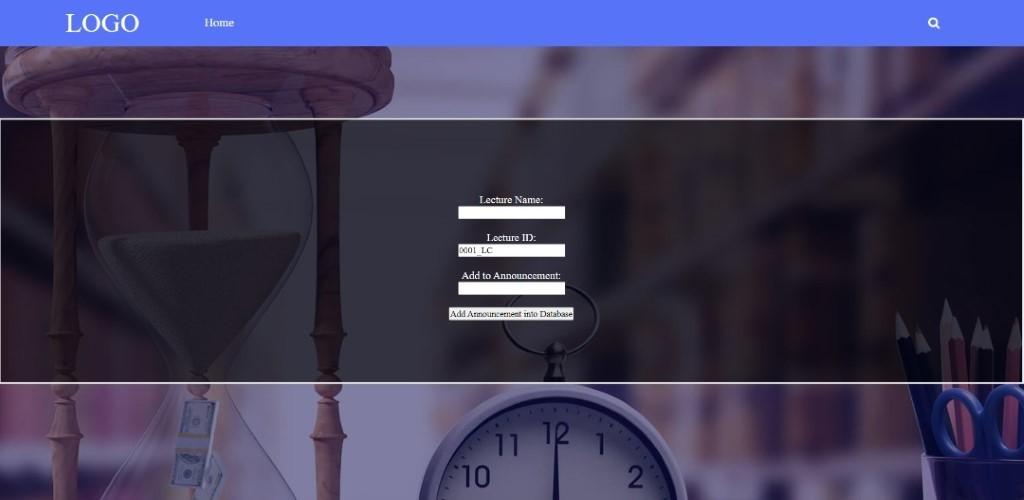


This is the add timetable page. Lecturer will use this page to add timetable for student. Lecturer are required to fill in all necessary information and click “add time” to submit it into the database.

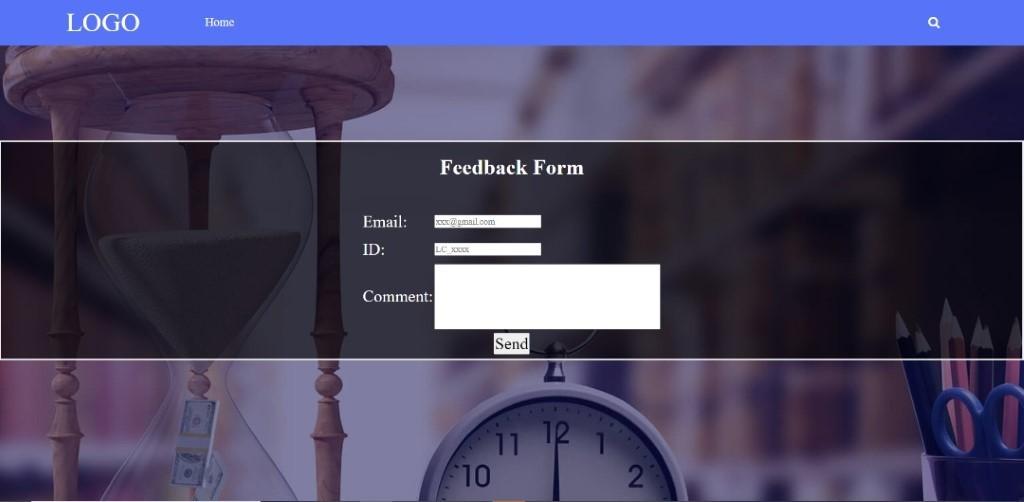




Lecturer will be redirected to this manage profile page if they press on “Manage Profile” from features in the homepage. Lecturer are able to upload new profile picture and change their personal information. The only thing the lecturer are not allowed to change is the lecturer ID as every lecturer will have their very own lecturer ID given to them as they register on the website and it will stay with them until they retired. After the lecturer has finish changing their personal information, they can click on the “Update” button to update their information.



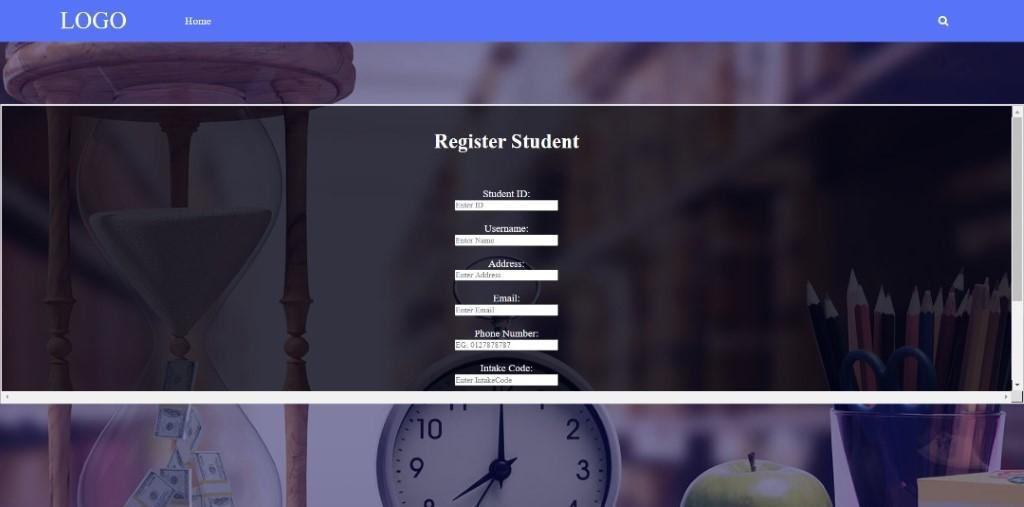
This is the lecturer add announcement function. Lecturer will be redirected to this page if they press on the “send announcement” button at the homepage. Lecturer ID will be automatically added by the system according to their ID which they use to login onto the website. After inputting the announcement, lecturer can click on the “Add Announcement onto Database” button to submit it into the database.



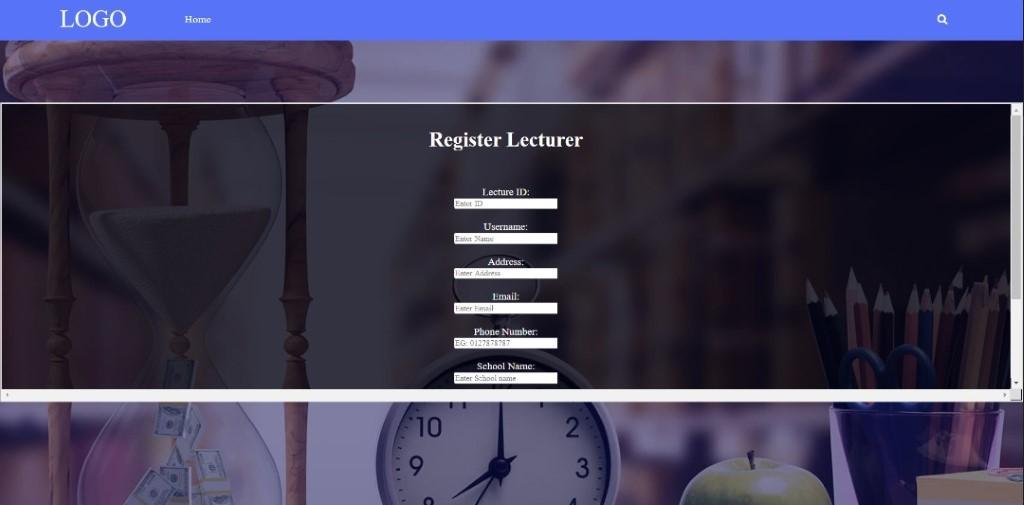
Lecturer will be redirected to this send feedback form if they press on the “Send Feedback” on the feature at homepage. Lecturer have to enter their Email, lecturer ID and comment to submit the feedback.

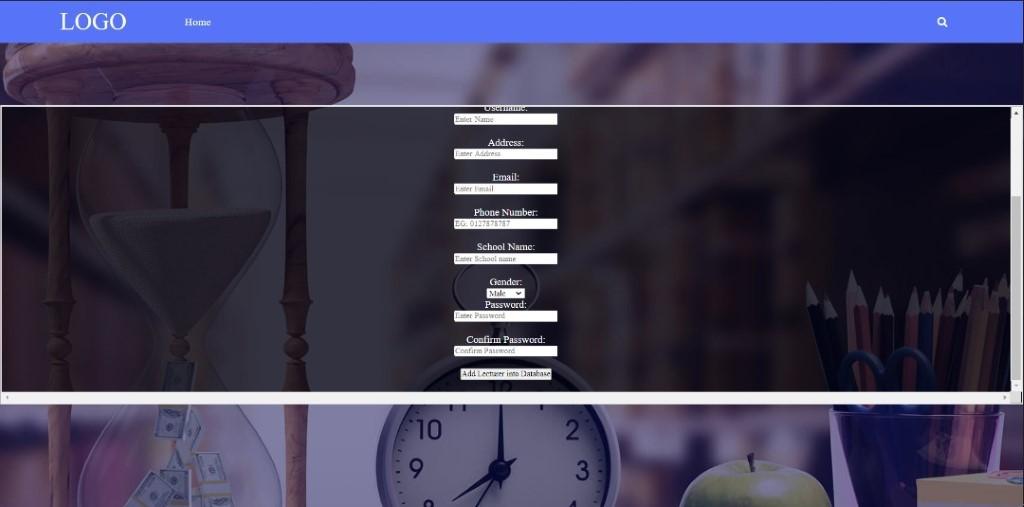


This is the lecturer add consultation hour page. Lecturer will be redirected to this page when they click on the “Add Consultation Hour” button in features at the homepage. Lecturer is required to enter the necessary information and submit into the database. Students will be able to view the available time and book consultation when they search for the particular lecturer.



This is the register student page. Admin will be redirected to this page when they click on “Student Account Registration” button in the features at the homepage. This page allow admin to register student into the system. Admin will insert the info according to a list given to them by the school.

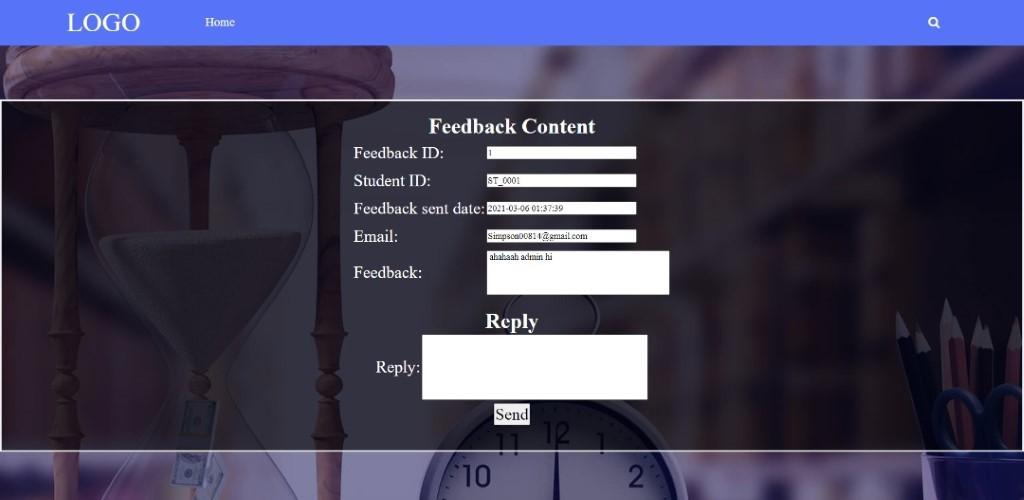




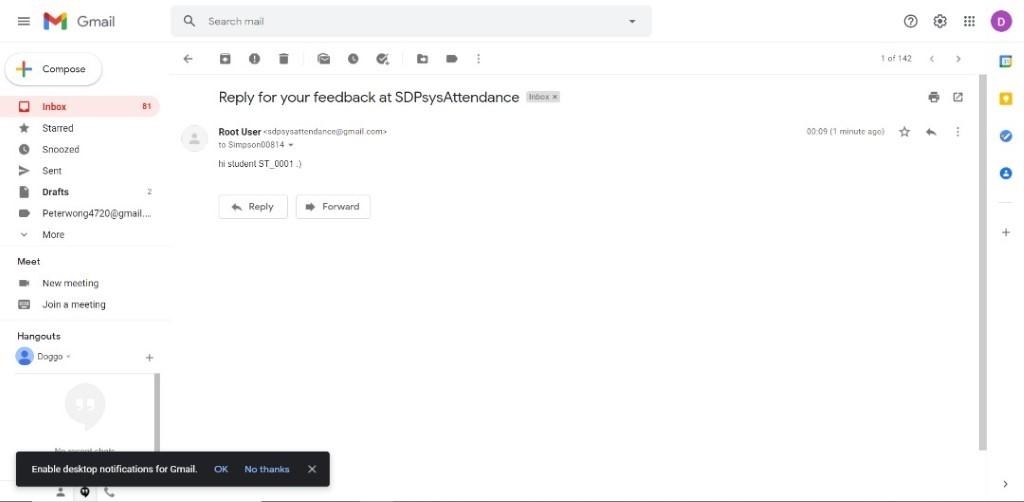
This is the register lecturer page. Admin will be redirected to this page when they click on “Lecturer Account Registration” button in the features at the homepage. This page allow admin to register lecturer into the system. Admin will insert the info according to a list given to them by the school.



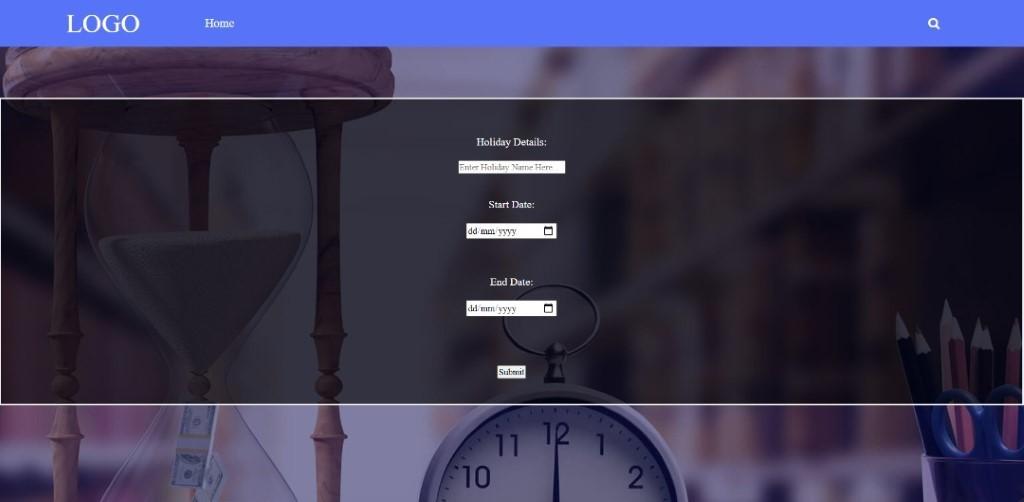
This is the admin add noticeboard function. Admin will be redirected to this page if they press on the “Add Notice Board” button at the homepage. Admin ID will be automatically added by the system according to their ID which they use to login onto the website. After inputting the notice, admin can click on the “Add Notice Board onto Database” button to submit it into the database.



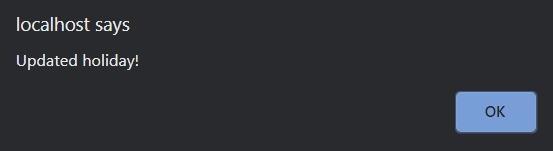
After the admin select a feedback and click reply, this page will show up. Admin will be able to reply according to the feedback details.



Above is an example of a feedback reply from admin that have been sent to student’s email.

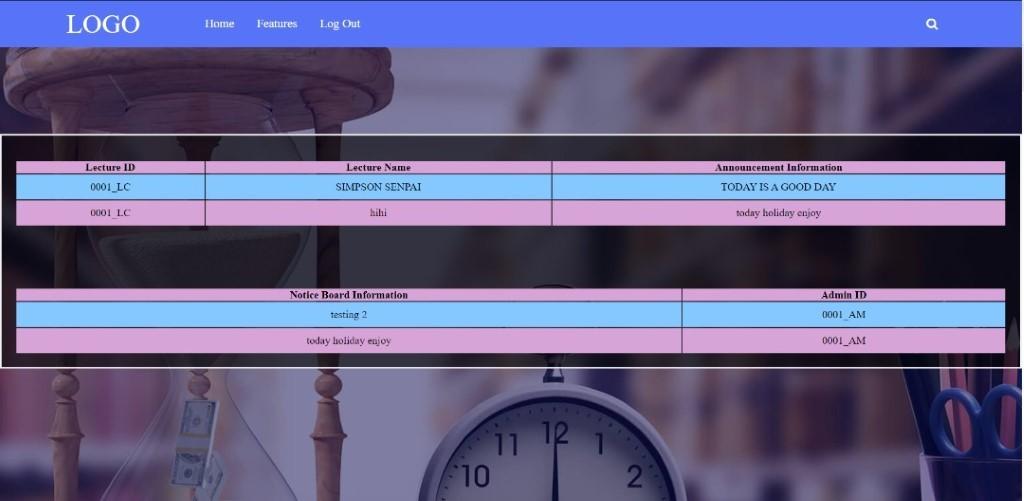


This is the add Holiday page. Admin are able to new Holiday into the database here. Admin are required to enter information needed and submit it onto the database.



After the system successfully updated the database, this message will show up.

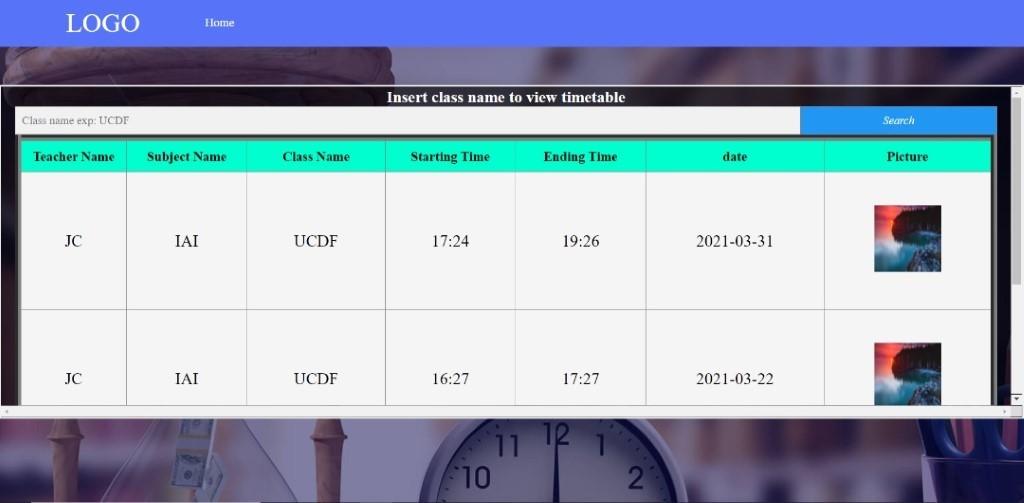
# **Report Designs and User Manual**



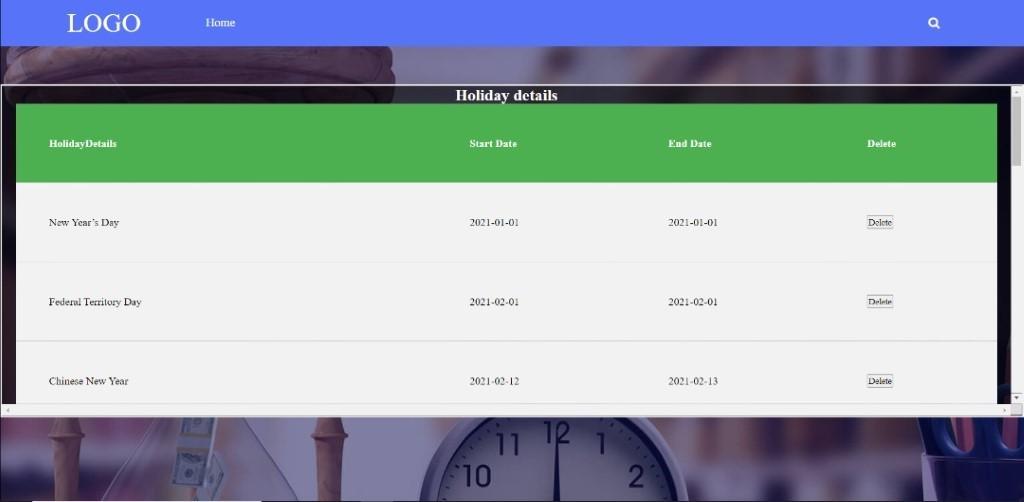
This is the homepage after the user login as a student. The middle section of the website will display any announcement post by the lecturer and any notices releases by the website admin.



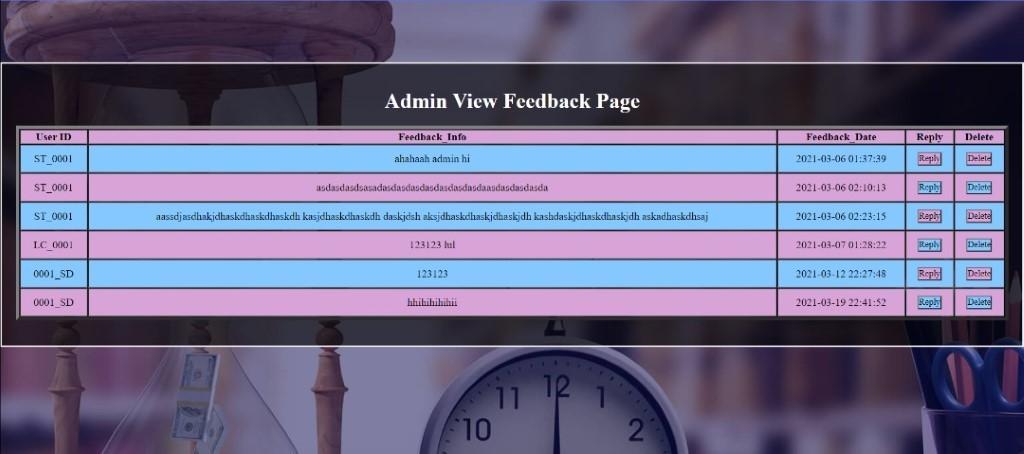
This is the homepage after the user login as a student. The middle section of the website will display any announcement post by the lecturer and any notices releases by the website admin.



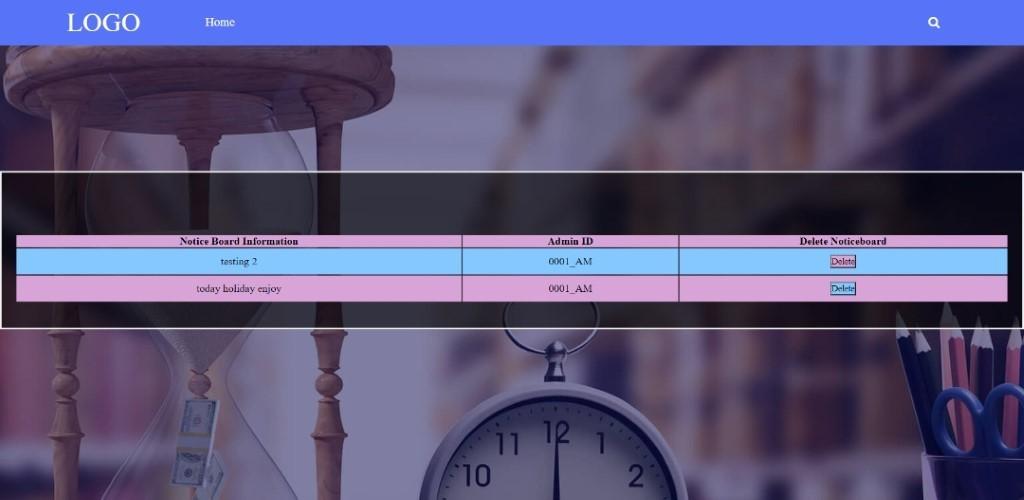
This is the view timetable page. Admin will be able to view the timetable by searching for it with the class name.



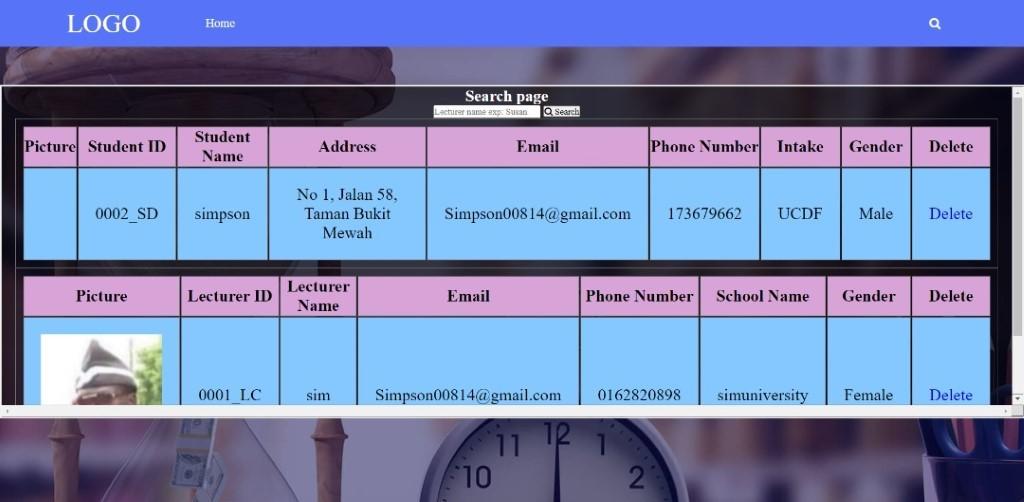
This is the view holiday page for admin. Admin will be able to view and delete holiday here.



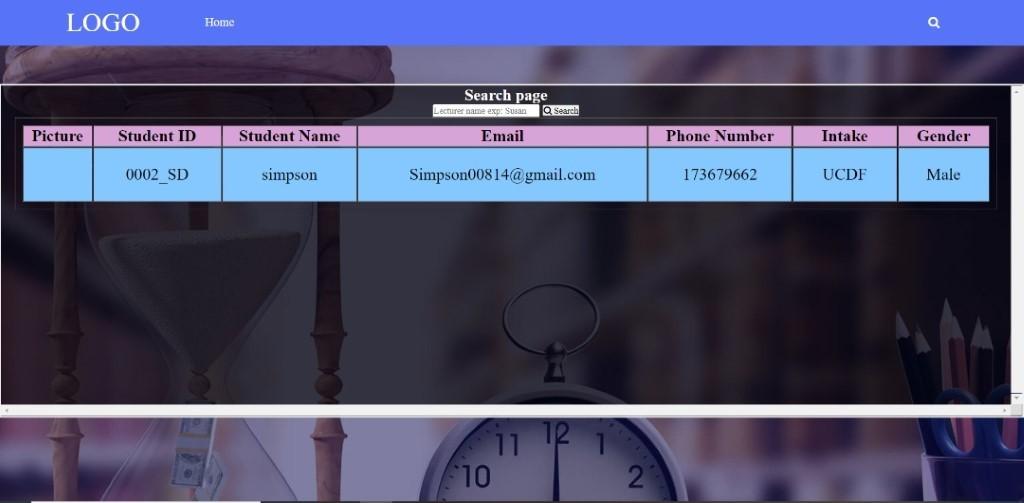
This is the admin manage feedback page. Admin will be redirected to here if they click on the “Manage Feedback” button at the homepage. Admin will be able to reply and delete the feedback.



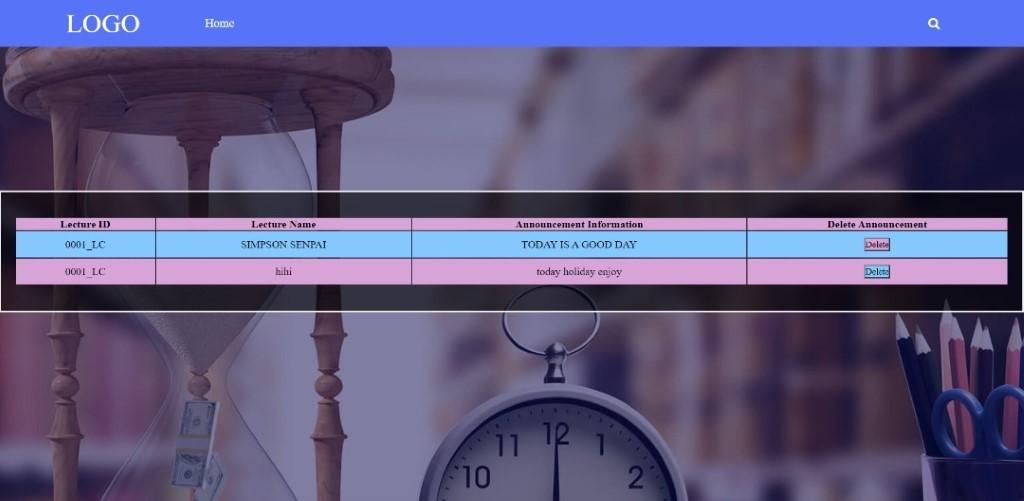
This is the edit noticeboard page. Admin are able to edit or delete the notice that they have already send after they click “Edit Notice Board” button at homepage.



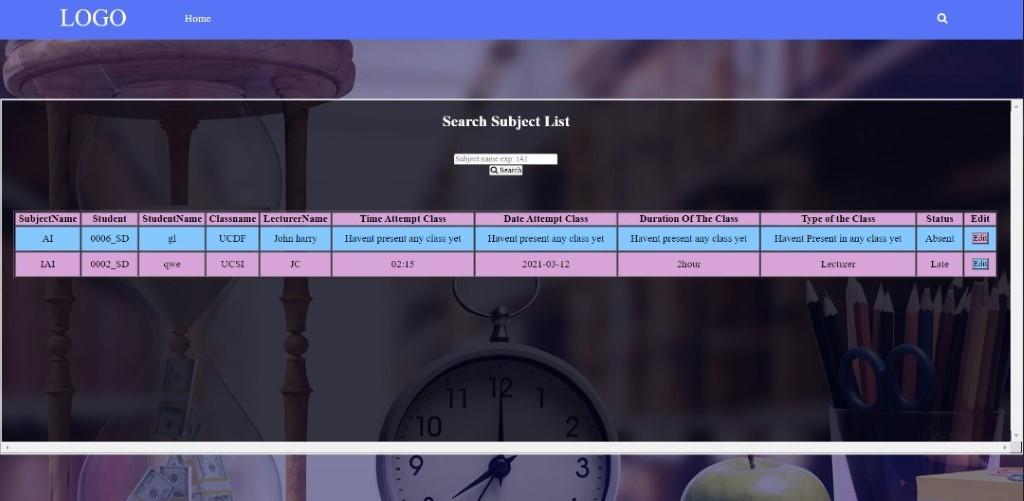
The search bar when pressed will bring the admin to the search page where the admin can search for both lecturer’s information and student’s information via lecturer’s name and student’s name. The full information about the lecturer and student will be displayed as shown.  Admin will also have the ability to delete both lecturer and student account if they retired or graduated.



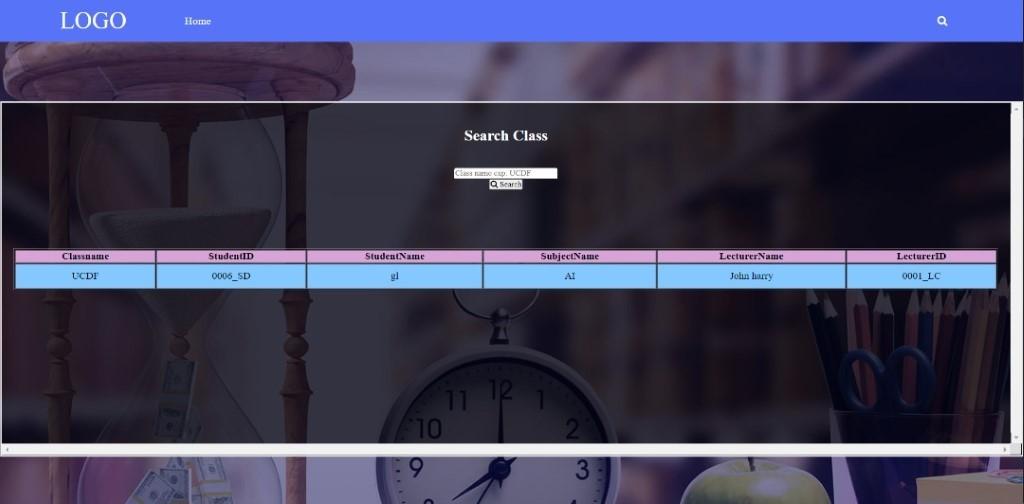
The search bar when pressed will bring the lecturer to the search page where the lecturer can search for student’s information via student’s name. The full information about the student will be displayed as shown.



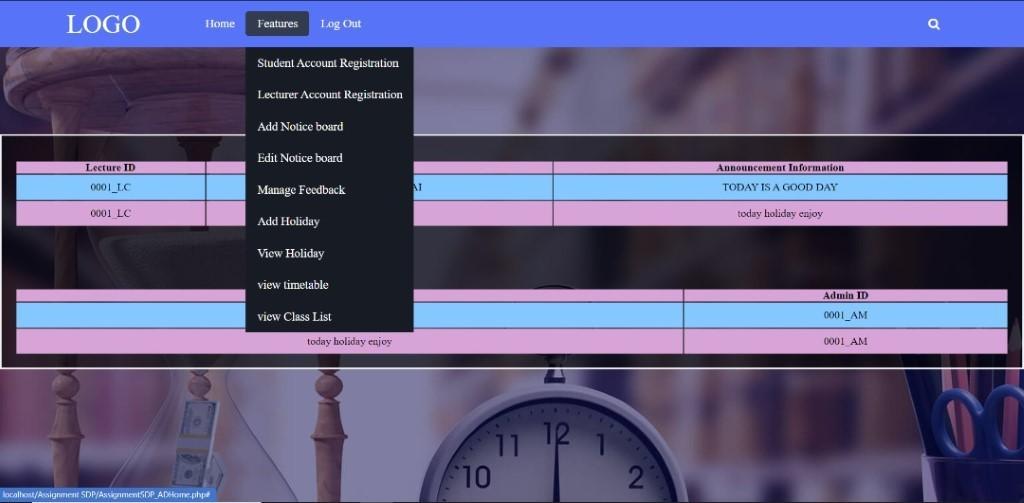
This is the edit announcement page. Lecturer can edit or delete the announcement that they have already send to the student after they click edit announcement in homepage.



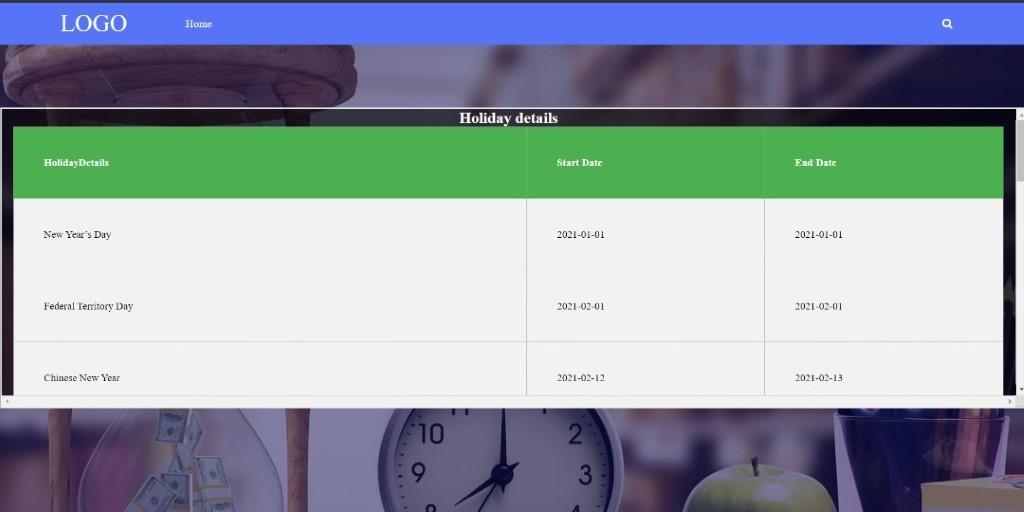
Above is an example of student who have taken and haven’t taken their attendance.



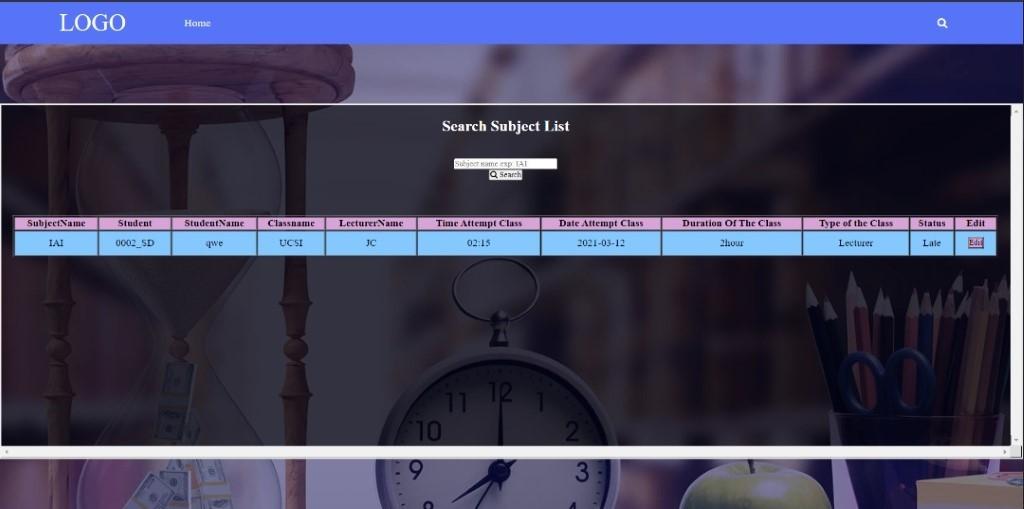
After adding the student into the class, lecturer will be able to view it at the search class list function at the homepage. Lecturer just have to search it using the class name and the information about the class will be shown.



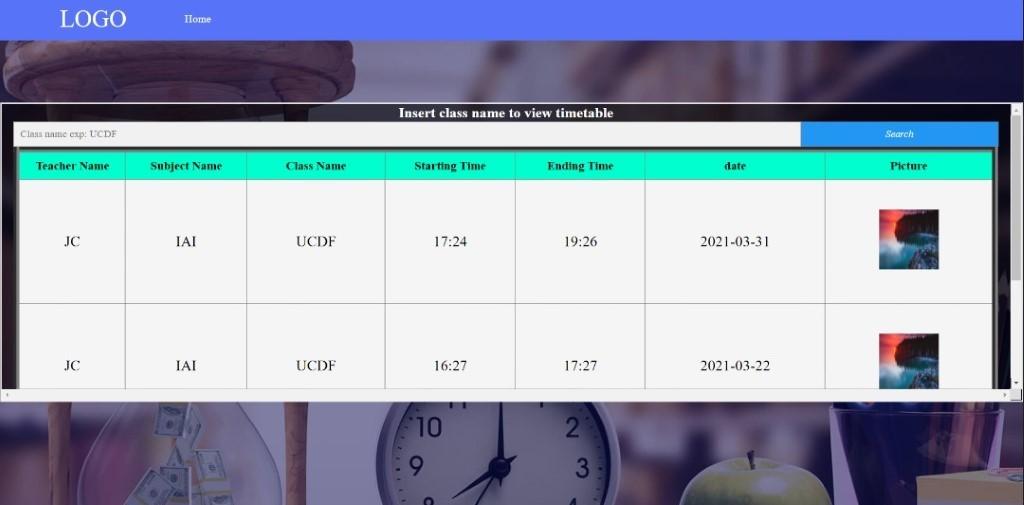
This is the homepage after the user login as a admin. The middle section of the website will display any announcement post by the lecturer and any notices releases by the website admin. At the top of the website, there will be a few buttons available for the admin. The home button will simply refresh the page. The features button when click will show all the functions available at the website for admin. Admin can click on the logout button to exit the website.



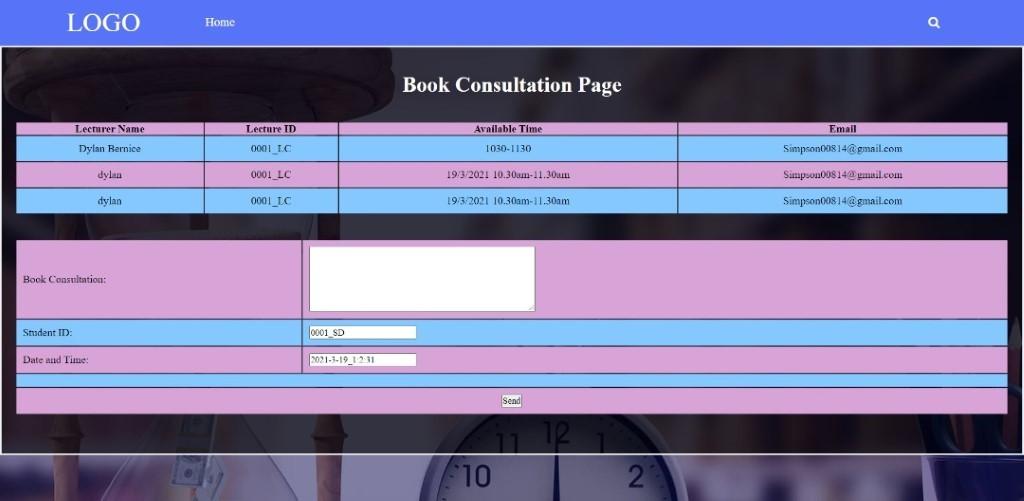
This is the check Holiday page. Lecturer are able to view Holiday details such as the start date and the end date of the Holiday via the view holiday button in the features at the homepage.



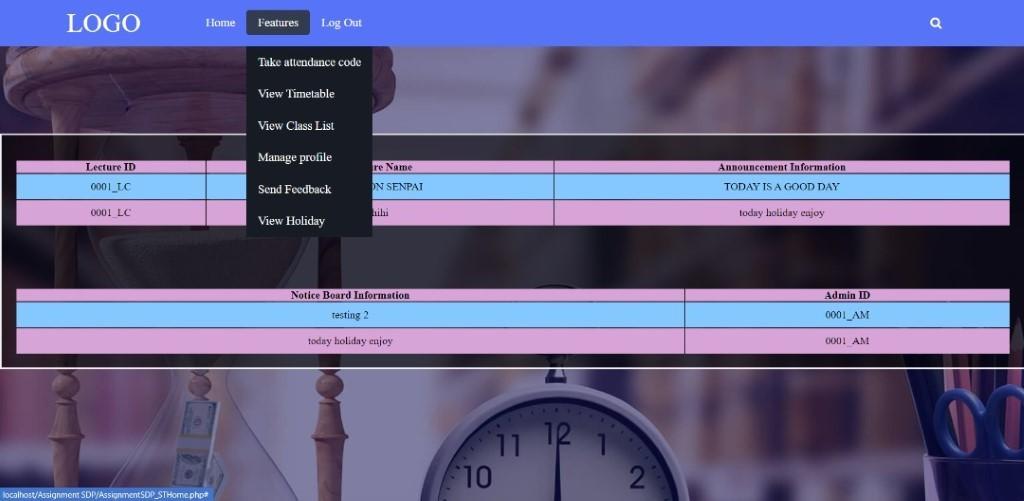
After taking attendance, lecturer will be able to view the class information by searching it with the subject name in the search subject list at the homepage, information such as the attendance status of the student, class time and class duration and many more will be shown. The lecture is also allowed to edit the attendance status for student manually by pressing the “edit” button.



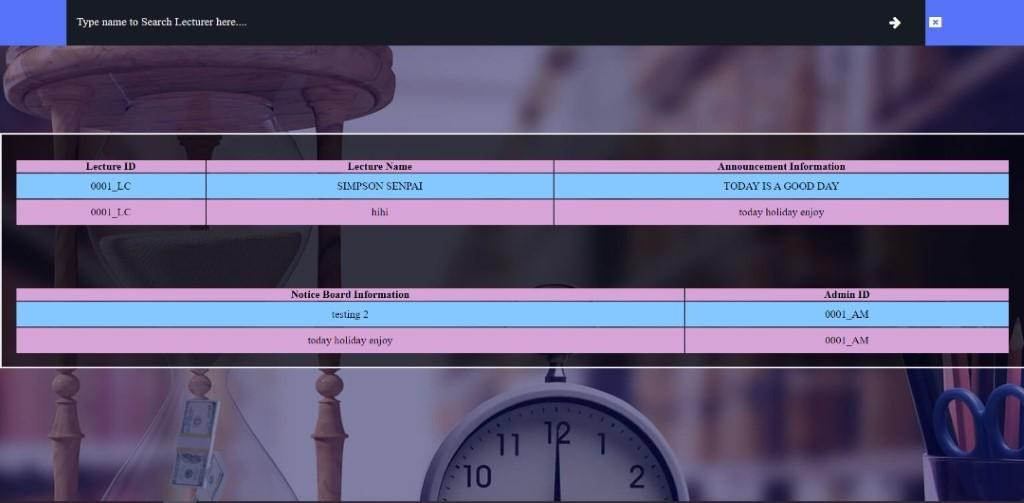
After the timetable was successfully updated into the database. Lecturer are able to view the timetable by searching for it with class name.



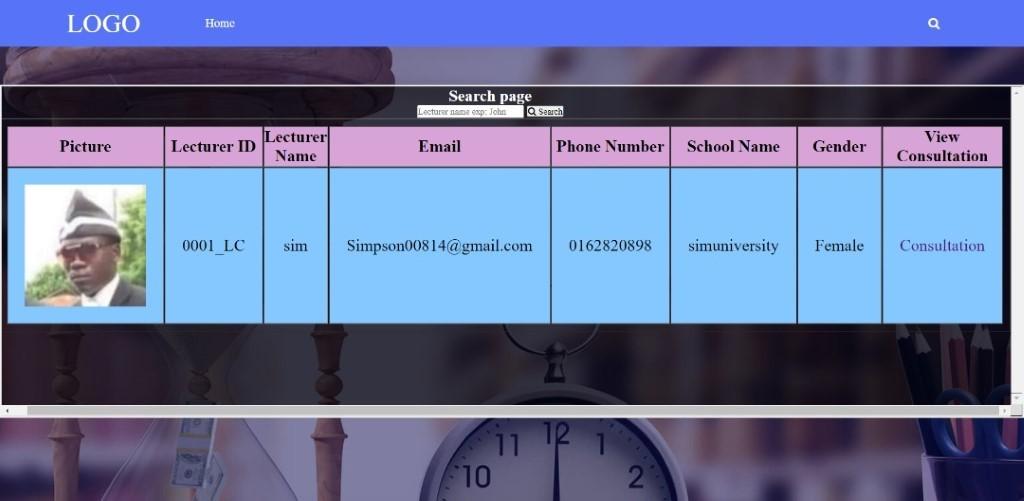
This is the book consultation page. Student will be able to see the section available for the lecturer they searched for. If the student wanted to book the consultation, they can type in their information and the time they would like to have and submit it. Our system will forward the message via an email to the lecturer.



At the top of the website, there will be a few buttons available for the students. The home button will simply refresh the page. The features button when click will show all the functions available at the website to the student. Students can click on the logout button to exit the website.



The search bar when pressed will bring the student to the search page where the student can search for lecturer information via lecturer’s name.



The full information about the lecturer will be displayed. Student can then proceed to view consultation time available for that particular lecturer by pressing on the consultation button. The home button on the top will redirect student back to the homepage when pressed.

# **Test Plan**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Test Unit** | **Test Procedures** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Remarks** |
| 001 | Student Login Process | -Insert Username and password.    -Click login button. | **Correct input data**  Student ID  -0001\_SD  Password  -123123 | Successfully login into system homepage according to their roles. | Can function | Done |  |
| 002 | Student Login Process | -Insert wrong Username and password.    -Click login button. | **Incorrect input data**  Username  -0001\_STU  Password  -wrongPass | Fail to login to our system. | Can function | Done |  |
| 003 | Lecturer Login Process | -Insert Username and password.    -Click login button. | **Correct input data**  Lecturer ID  -0001\_LC  Password  -123123 | Successfully login into system homepage according to their roles. | Can function | Done |  |
| 004 | Lecturer Login Process | -Insert wrong Username and password.    -Click login button. | **Incorrect input data**  Lecturer ID  -0001\_LCT  Password  - wrongPass | Fail to login to our system. | Can function | Done |  |
| 005 | Admin Login Process | -Insert Username and password.    -Click login button. | **Correct input data**  Admin ID  -0001\_AD  Password  -123123 | Successfully login into system homepage according to their roles. | Can function | Done |  |
| 006 | Admin Login Process | -Insert wrong Username and password.    -Click login button. | **Incorrect input data**  Admin ID  -0001\_ADM  Password  - wrongPass | Fail to login to our system. | Can function | Done |  |
| 007 | Forget Password Process | -Provide email    -Click submit button    -send link to email provided | **Insert email**  -Simpson00814@gmail.com | Reset password link will provide in the email. | Get Reset password link in the email. | Done |  |
| 008 | Reset password process | -provide ID that related with the email given in forget password according to database      -provide new password    -Click update button | **Insert UserID that related with email.**  EXP User ID:  -0001\_SD  New Password:  -qweqwe  Double check password:  -qweqwe | Successfully reset password according to ID and bring user to login page. | Reset password successfully and bring user to login page. | Done |  |
| 009 | Reset password process | -provide ID that **not** related with the email given in forget password according to database    -provide new password    -Click update button | **Insert UserID that related with email and different password.**  EXP User ID:  -0001\_SD  New Password:  -qweqwe  Double check password:  -qqwwee | Show wrong password entered and fail to update new password to database | Reset password fail. | Done |  |
| 010 | Registration process | -Insert data about account    -Click register button | **Correct input data** | Successfully register into database | Can function | Done |  |
| 011 | Registration process | -Insert data about existing account    -Click register button | **Incorrect   input data** | Fail to register account | Can function | Done |  |
| 012 | Announcement Process | -Insert announcement    -Click add announcement button | **Correct input data** | Successfully add announcement into database | Can function | Done |  |
| 013 | Notice Board Process | -Insert Notice Board Details    -Click Add Notice Board button | **Correct input data** | Successfully add notice board details into database | Can function | Done |  |
| 014 | Find subject Process | -Insert subject Details    -Click Search Student button | **Correct input data** | Show subject list | Can function | Done |  |
| 015 | Find subject Process | -Insert subject name    -Click Search Student button | **Incorrect input data** | Not result found | Can function | Done |  |
| 016 | Find class process | -Insert class name    -click search class button | **Correct input data** | Show class list | Can function | Done |  |
| 017 | Find class process | -Insert class name    -click search class button | **Incorrect input data** | Fail to find class details | Can function | Done |  |
| 018 | Update consultation table | Insert consultation details | **Correct input data** | Successfully updated consultation database | Can function | Done |  |
| 019 | Find consultation details | Insert consultation details | **Correct input data** | Successfully showed consultation information | Can function | Done |  |
| 020 | Find consultation details | Insert consultation details | **Incorrect input data** | Fail to show consultation details | Can function | Done |  |
| 021 | Student book consultation function | Search lecturer, click consultation button, send details to lecturer | **Insert details to lecturer**  **-**Hi sir, I am simpson from UCDF and would like to book slot 10am-11am. | Successfully sent details to lecturer email. | Can function | Done |  |
| 022 | Generate attendance code | Insert class information | **Correct input data** | Show attendance code | Can function | Done |  |
| 023 | Generate attendance code | Insert class information | **Incorrect input data** | Fail: unable to submit data | Can function | Done |  |
| 024 | Insert timetable details | Enter class information and date | **Correct input data** | Updated successful | Can function | Done |  |
| 025 | Insert timetable details | Enter class information and date | **Incorrect input data** | Back to insert page | Can function | Done |  |
| 026 | Update attendance | Enter attendance code | **Correct input data** | Attendance Updated successful | Can function | Done |  |
| 027 | Update attendance | Enter attendance  code | **Incorrect input data** | Attendance Updated Failed | Can function | Done |  |
| 028 | Timetable Process | Enter class name | **Correct input data** | Show timetable follow the class name | Can function | Done |  |
| 029 | Timetable Process | Enter class name | **Incorrect input data** | No result found | Can function | Done |  |
| 030 | Holiday details process | Click view holiday details button | **Correct input data** | Show updated holiday info | Can function | Done |  |
| 031 | Add holiday details | Insert holiday information | **Correct input data** | Updated holiday | Can function | Done |  |
| 032 | Add holiday details | Insert holiday information | **Incorrect input data** | Fail: unable to submit data | Can function | Done |  |
| 033 | Updated student attendance status | Change the status of attendance | **Correct input data** | Successfully update | Can function | Done |  |
| 034 | Add student to class | Insert student information | **Correct input data** | Added into class | Can function | Done |  |
| 035 | Add student to class | Insert student information | **Incorrect input data** | Failed | Can function | Done |  |
| 036 | Lecturer Search Student Function | Insert student name | **input name that consists in database** | Display student | Can function | Done |  |
| 037 | Lecturer Search Student Function | Insert student name | **input name that does not consist in database** | Display error message | Can function | Done |  |
| 038 | Student Search Lecturer Function | Insert Lecturer name | **input name that consists in database** | Display Lecturer | Can function | Done |  |
| 039 | Student Search Lecturer Function | Insert Lecturer name | **input name that does not consist in database** | Display User | Can function | Done |  |
| 040 | Admin Search User Function | Insert Username | **input name that consists in database** | Display error message | Can function | Done |  |
| 041 | Admin Search User Function | Insert Username | **input name that does not consist in database** | Display error message | Can function | Done |  |
| 042 | Admin Delete User Function | Delete selected user | **Click delete button on selected user** | Delete from database | Can function | Done |  |
| 043 | Students send feedback function | Feedback sent to admin | **input comment and click send button** | Feedback sent | Can function | Done |  |
| 044 | Lecturers send feedback function | Feedback sent to admin | **input comment and click send button** | Feedback sent | Can function | Done |  |
| 045 | Admin view feedback function | View all feedback sent by user | **Click manage feedback button** | Able to view all feedback sent by user | Can function | Done |  |
| 046 | Admins delete feedback function | Delete selected feedback | **Click delete button on selected feedback** | Delete from database | Can function | Done |  |
| 047 | Admins reply feedback function | Send reply to user email | **Click reply button, type comment and click send button** | User able to get reply trough their email | Can function | Done |  |
| 048 | Students manage profile function | View and manage profile detail | **Click manage profile and change desirable detail then click update button.** | Student able to change their profile | Can function | Done |  |
| 049 | Lecturers manage profile function | View and manage profile detail | **Click manage profile and change desirable detail then click update button.** | Lecturer able to change their profile | Can function | Done |  |

# **User Acceptance Testing**

\*The following is the scale for the answers.

* 1-Poor
* 2-Moderate
* 3-Satisfied
* 4-Good
* 5-Excellent

|  |  |  |
| --- | --- | --- |
| **Tester Profile** | | |
| Name | : | See Xu Yan |
| Email | : | [seedulan@gmail.com](mailto:seedulan@gmail.com) |
| Date | : | 12-3-2021 |
| Time | : | 5pm |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Highlight appropriate answer** |
| **Admin Part** | | |
| 1 | Login into System as Admin | 1 2 3 4 5 |
| 2 | View Announcement | 1 2 3 4 5 |
| 3 | View Notice Board | 1 2 3 4 5 |
| 4 | Edit User Detail | 1 2 3 4 5 |
| 5 | Delete User Detail | 1 2 3 4 5 |
| 6 | Reply feedback | 1 2 3 4 5 |
| 7 | Register Student Account | 1 2 3 4 5 |
| 8 | Register Lecturer Account | 1 2 3 4 5 |
| 9 | Send Announcement | 1 2 3 4 5 |
| **Student** | | |  |  |
| 10 | Login into System as Student | 1 2 3 4 5 |
| 11 | View own profile detail | 1 2 3 4 5 |
| 12 | Search for Lecturer Detail | 1 2 3 4 5 |
| 13 | Send Feedback | 1 2 3 4 5 |
| 14 | Take Attendance | 1 2 3 4 5 |
| 15 | View Attendance | 1 2 3 4 5 |
| 16 | View Announcement | 1 2 3 4 5 |
| 17 | View Notice Board | 1 2 3 4 5 |
| **Lecturer** | | |  |  |
| 18 | Login into System as Lecturer | 1 2 3 4 5 |
| 19 | View own profile detail | 1 2 3 4 5 |
| 20 | Create Attendance | 1 2 3 4 5 |
| 21 | View Attendance status | 1 2 3 4 5 |
| 22 | View Student List | 1 2 3 4 5 |
| 23 | Send Notice Board | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **Tester Profile** | | |
| Name | : | Low Guo Hong |
| Email | : | [lowgh123@gmail.com](mailto:lowgh123@gmail.com) |
| Date | : | 12-3-2021 |
| Time | : | 5pm |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Highlight appropriate answer** |
| **Admin Part** | | |
| 1 | Login into System as Admin | 1 2 3 4 5 |
| 2 | View Announcement | 1 2 3 4 5 |
| 3 | View Notice Board | 1 2 3 4 5 |
| 4 | Edit User Detail | 1 2 3 4 5 |
| 5 | Delete User Detail | 1 2 3 4 5 |
| 6 | Reply feedback | 1 2 3 4 5 |
| 7 | Register Student Account | 1 2 3 4 5 |
| 8 | Register Lecturer Account | 1 2 3 4 5 |
| 9 | Send Announcement | 1 2 3 4 5 |
| **Student** | | |  |  |
| 10 | Login into System as Student | 1 2 3 4 5 |
| 11 | View own profile detail | 1 2 3 4 5 |
| 12 | Search for Lecturer Detail | 1 2 3 4 5 |
| 13 | Send Feedback | 1 2 3 4 5 |
| 14 | Take Attendance | 1 2 3 4 5 |
| 15 | View Attendance | 1 2 3 4 5 |
| 16 | View Announcement | 1 2 3 4 5 |
| 17 | View Notice Board | 1 2 3 4 5 |
| **Lecturer** | | |  |  |
| 18 | Login into System as Lecturer | 1 2 3 4 5 |
| 19 | View own profile detail | 1 2 3 4 5 |
| 20 | Create Attendance | 1 2 3 4 5 |
| 21 | View Attendance status | 1 2 3 4 5 |
| 22 | View Student List | 1 2 3 4 5 |
| 23 | Send Notice Board | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **Tester Profile** | | |
| Name | : | Chuah Xun Zhi |
| Email | : | [xunzhi123@gmail.com](mailto:xunzhi123@gmail.com) |
| Date | : | 13-3-2021 |
| Time | : | 4pm |

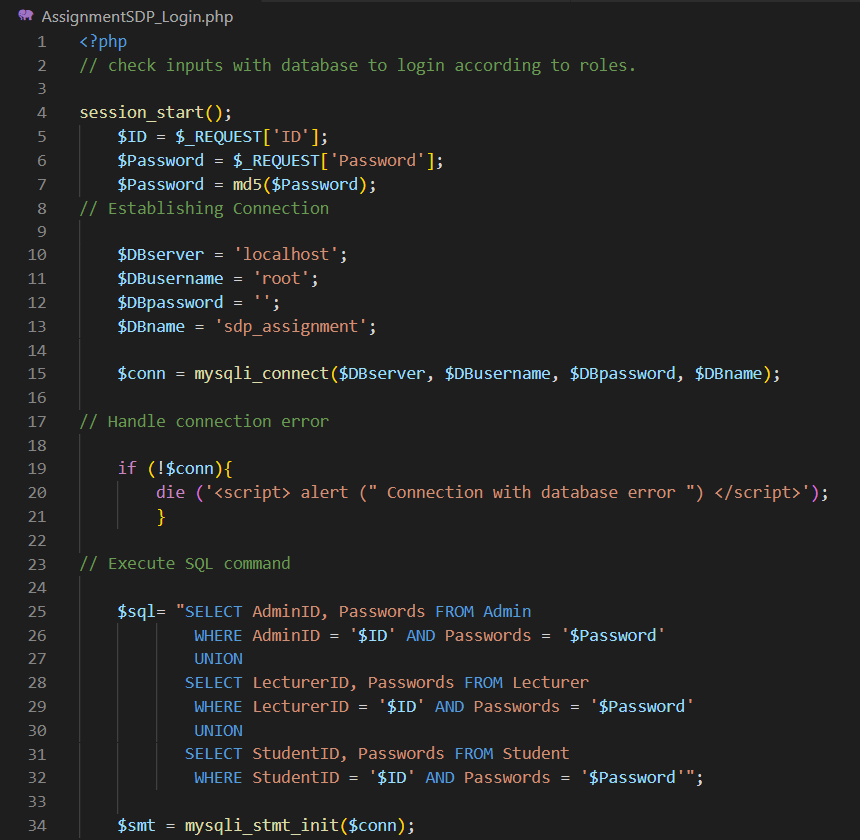
|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Highlight appropriate answer** |
| **Admin Part** | | |
| 1 | Login into System as Admin | 1 2 3 4 5 |
| 2 | View Announcement | 1 2 3 4 5 |
| 3 | View Notice Board | 1 2 3 4 5 |
| 4 | Edit User Detail | 1 2 3 4 5 |
| 5 | Delete User Detail | 1 2 3 4 5 |
| 6 | Reply feedback | 1 2 3 4 5 |
| 7 | Register Student Account | 1 2 3 4 5 |
| 8 | Register Lecturer Account | 1 2 3 4 5 |
| 9 | Send Announcement | 1 2 3 4 5 |
| **Student** | | |  |  |
| 10 | Login into System as Student | 1 2 3 4 5 |
| 11 | View own profile detail | 1 2 3 4 5 |
| 12 | Search for Lecturer Detail | 1 2 3 4 5 |
| 13 | Send Feedback | 1 2 3 4 5 |
| 14 | Take Attendance | 1 2 3 4 5 |
| 15 | View Attendance | 1 2 3 4 5 |
| 16 | View Announcement | 1 2 3 4 5 |
| 17 | View Notice Board | 1 2 3 4 5 |
| **Lecturer** | | |  |  |
| 18 | Login into System as Lecturer | 1 2 3 4 5 |
| 19 | View own profile detail | 1 2 3 4 5 |
| 20 | Create Attendance | 1 2 3 4 5 |
| 21 | View Attendance status | 1 2 3 4 5 |
| 22 | View Student List | 1 2 3 4 5 |
| 23 | Send Notice Board | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **Tester Profile** | | |
| Name | : | Ng Ken Leong |
| Email | : | ngkengleong[1217@gmail.com](mailto:1217@gmail.com) |
| Date | : | 17-3-2021 |
| Time | : | 12pm |

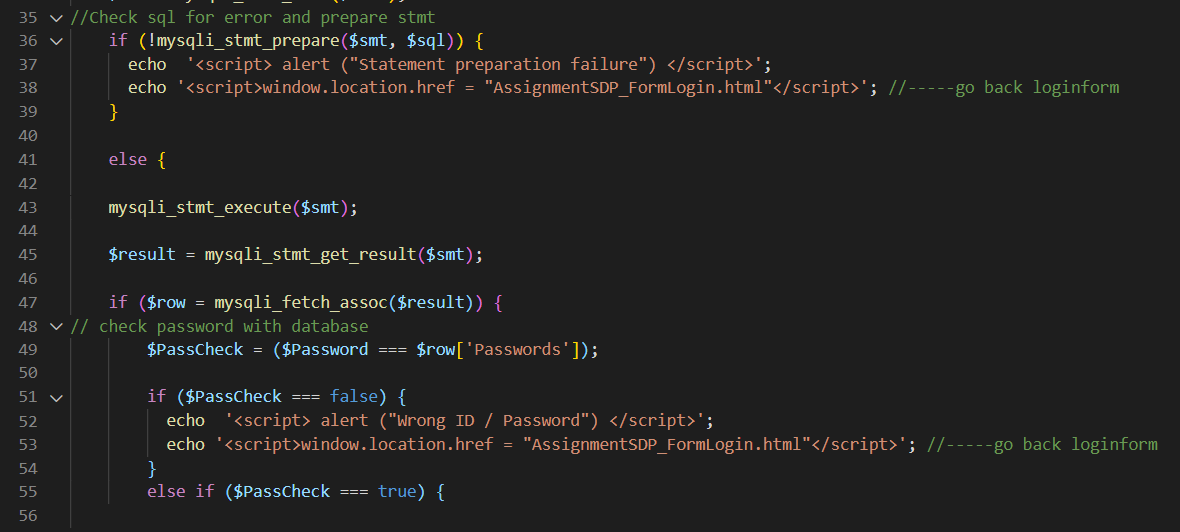
|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Highlight appropriate answer** |
| **Admin Part** | | |
| 1 | Login into System as Admin | 1 2 3 4 5 |
| 2 | View Announcement | 1 2 3 4 5 |
| 3 | View Notice Board | 1 2 3 4 5 |
| 4 | Edit User Detail | 1 2 3 4 5 |
| 5 | Delete User Detail | 1 2 3 4 5 |
| 6 | Reply feedback | 1 2 3 4 5 |
| 7 | Register Student Account | 1 2 3 4 5 |
| 8 | Register Lecturer Account | 1 2 3 4 5 |
| 9 | Send Announcement | 1 2 3 4 5 |
| **Student** | | |  |  |
| 10 | Login into System as Student | 1 2 3 4 5 |
| 11 | View own profile detail | 1 2 3 4 5 |
| 12 | Search for Lecturer Detail | 1 2 3 4 5 |
| 13 | Send Feedback | 1 2 3 4 5 |
| 14 | Take Attendance | 1 2 3 4 5 |
| 15 | View Attendance | 1 2 3 4 5 |
| 16 | View Announcement | 1 2 3 4 5 |
| 17 | View Notice Board | 1 2 3 4 5 |
| **Lecturer** | | |  |  |
| 18 | Login into System as Lecturer | 1 2 3 4 5 |
| 19 | View own profile detail | 1 2 3 4 5 |
| 20 | Create Attendance | 1 2 3 4 5 |
| 21 | View Attendance status | 1 2 3 4 5 |
| 22 | View Student List | 1 2 3 4 5 |
| 23 | Send Notice Board | 1 2 3 4 5 |

# **Significant Source Codes**

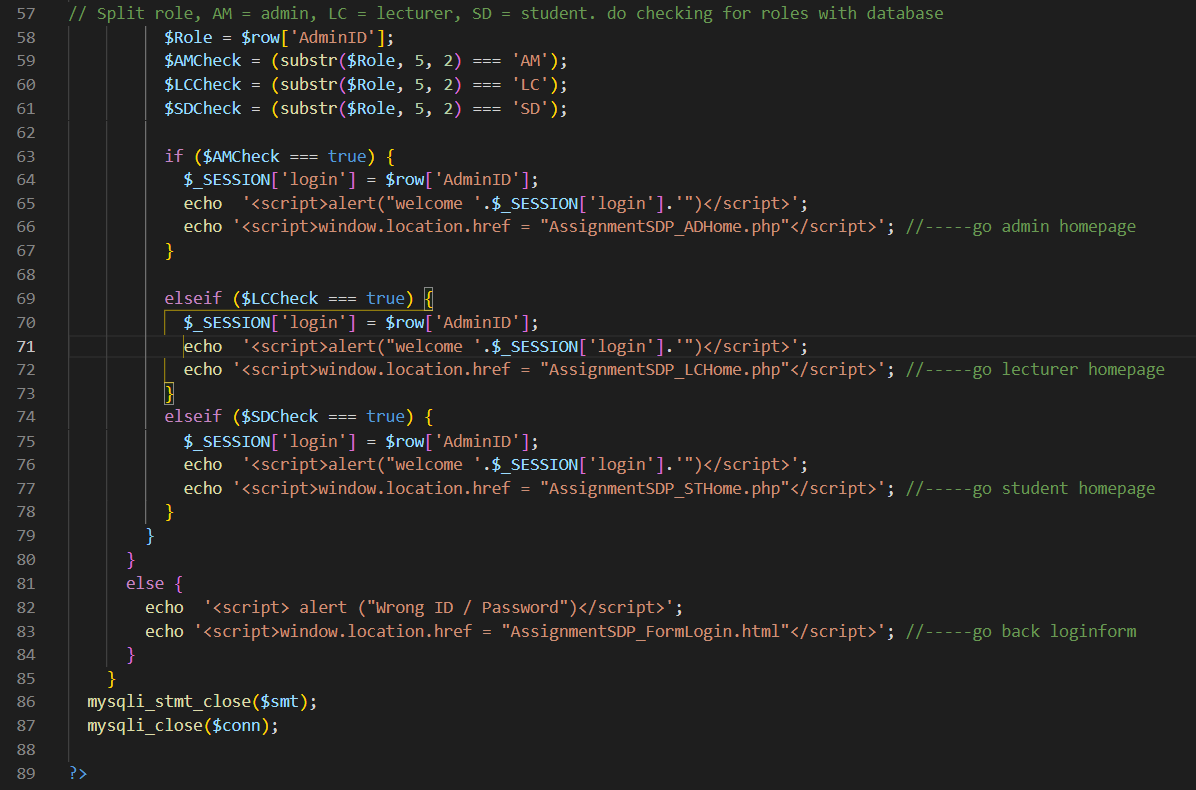
**Login**



The about code is where our system need to decide and check the input user type is which role and go to which homepage. This code is consisting within the login page of our system. It is executed after user had type out his/her ID and password in the login page and clicked the login button. Firstly, it will connect to the database using $conn and handle the error if can’t connect to the database with the if statement at row 19 to 21 above. After that, it will declare the SQL statement which will get the ID and the password in the database that match with input that user gave and start preparing the statement which in our system is called $smt by giving it connection to database.



Continuing prepare the statement by checking it with if function. If the statement and the SQL code is good to go, it will start executing the statement get the result and store it in $result. After that, the result will be fetched by the if function and start checking the password that user input with the password consist in database.

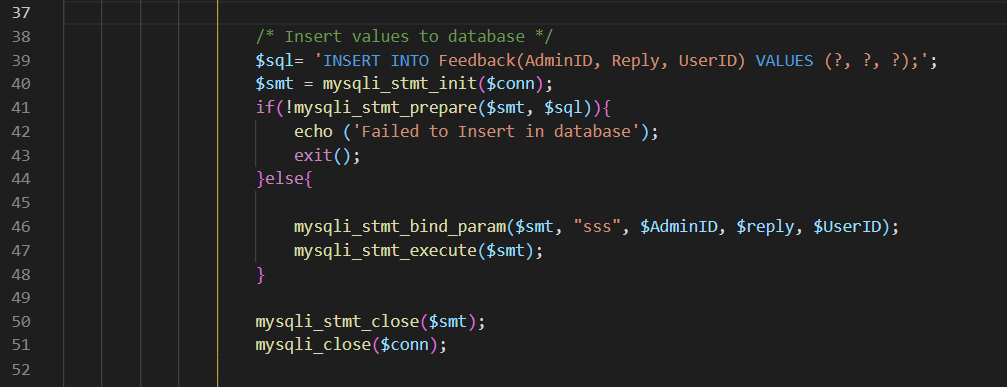


After checking the password, the code then checks the ID to see which homepage it will go by first get the ID from the database result. It only needs to get $row[‘AdminID ’] because of the union function in the SQL code that combine the table name to the first selected table with is AdminID. After that it will start check the alphabet behind the ID by using substr as shown above and put it in to an if function to direct user to their homepage and welcome them using session with their ID on it.

**Admin Reply Feedback**



This code will start executing after Admin type out his/her reply and click the submit button. Firstly, it will connect to database using the code from line 2 to 10. After that, it will use post function to get the information needed for this to work from the reply feedback form. It will then require and get the PHPMailer file for the code to work. From line 21 to 34 is the code that start setting up the PHPMailer, give the Username and Password of the Sender’s email to it and get the receiver email which is the email from the user who send the feedback from the form.



After that, it will declare and execute the SQL and SMT statement, if one of them failed to prepare or have error it will display a message and exit the system. If no problem them, it will continue to enter the value into the question marks in the SQL statement and execute it.



If everything is good until now, it will start executing the code which is from line 55 until 64 that will send the reply to the user by setting it up according to the email format, put the reply details in the body part and send it to the user’s email.

# **Conclusion**

## **Assumptions**

Our system will be able to help many schools to get rid of the old way of getting attendance such as by calling names and checking their names on papers which is very time consuming and unreliable. It will also make both the lecturers and students’ life easier by allowing them to make or check timetables and announcement via the website. This website also allow lecturer to set up his consultation time for students to book consultation which make booking consultation process smoother and easier for both lecturers and students.

## **Limitation**

One of the limitations of our website is that it is not phone and tablet friendly. The other limitation for this system is that it required everyone to have a device that can browse the internet to use it. With this also create another limitation which is internet connection. Our system required internet connection to work properly. Other limitation is our system having very limited functions like taking attendance, check user details, check timetable and make announcement.

## **Enhancements**

Example of enhancement that can be made towards the system are lecturer able to get the number of students that get attendance by a pie chart format which help lecturer to study the stats easily. Other functions like enabling students to view their exam result via the website is also an enhancement available to the website. Another improvement available to the website would be enabling the lecturers to upload study materials onto the website for students to download and use. Offline attendance mode will also be a huge enhancement for the system as the website will still be able to take attendance despite having no internet connection but will upload the attendance data to the database as soon as the device receive internet connection.

# **Reference**

# **Appendix**

## **Meeting Details**

|  |  |  |
| --- | --- | --- |
| Meeting Dates | Meeting Duration | Description (Remarks from Lecture) |
| 12/17/2020 | 15 Minutes | 1. Attendance System – Online and web based.  2. 3 Users, lecture, student and admin  3. Next, finalize all the functionalities per users, completing the proposal and completing the progress report. |
| 21/12/2020 | 15 Minutes | 1. Hierarchy Chart: update require on the report, feedback  2. Project Scope: need to update  3. Gantt Chart: Need to modify based on selected methodology (more details) |
| 14/1/2021 | 15 Minutes | Complete all the flowcharts (currently only several were done) |
| 21/1/2021 | 15 Minutes | Done Context Diagram. Need to do correction on the Context Diagram data flows. After that may continue to other level |
| 4/2/2021 | 15 Minutes | Completed the ERD |
| 18/2/2021 | 15 Minutes | Drafted the unit test plan. Not start with UAT plan yet. |
| 4/3/2021 | 15 Minutes | Development of the system currently at 20%. |
| 11/3/2021 | 15 Minutes | The development of the system currently at 85%. Good Progress. |

## **Installation Guide**

SIMPLE ATTENDANCE SYSTEM IN PHP/HTML/CSS

\*ADMIN LOGIN DETAILS\*

Username : 0001\_AM

Password : 123123

\*Lecturer LOGIN DETAILS\*

Username : 0001\_SD

Password : 123123

\*Student LOGIN DETAILS\*

Username : 0001\_LC

Password : 123123

\*Developed By Group 17\*

DON'T FORGET TO CREATE A DATABASE NAMING "sdp\_assignment" AND IMPORT THE DATBASE SQL FILE.

WITHOUT THE DATABASE THE PROJECT WON'T RUN.

>> DATABASE FILE IS INSIDE "DATABASE" FOLDER!!

\*\* IF YOU FIND ANY ERRORS OR ANY PROBLEMS RELATED THIS PROGRAM, FEEL FREE TO CONTACT US USING SEND FEEDBACK FUNCTION INSIDE OUR PROJECT \*\*

#THANK YOU FOR DOWNLOADING

(Bikash, 2020)

**Counts Words as user type**

<https://www.youtube.com/watch?v=Pcsy93pk7_Q>

**How To Create A Forgotten Password System In PHP**

<https://youtu.be/wUkKCMEYj9M>

**How to do upload button using php**

<https://www.youtube.com/watch?v=JaRq73y5MJk>

**How to do a search function using php**

<https://www.youtube.com/watch?v=PBLuP2JZcEg>

# References

Bikash, 2020. *Display Current Date and Time in HTML using JavaScript.* [Online]   
Available at: https://phpcoder.tech/display-current-date-and-time-in-html-using-javascript/  
[Accessed 22 2 2021].