



Requirements Specification

Project Name: *DISMS-13*

Team Number 13

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

Document Information

Project Name:	DISMS-13	Preparation Date:	15/03/2024
Prepared By:	Team 13		
Email / Phone:	b9037550@newcastle.ac.uk		
Document Version No:	2.4	Document Version Date:	09/05/2024

Version History

Ver. No.	Ver. Date	Revised By	Description
1.0	15/03/2024	Harry Hainsworth-Staples	Outlying the foundation (Purpose + Scope)
1.1	17/03/2024	Vanya Liang	Initial ideas for GUI + Analysis Process
1.2	19/03/2024	Shuyi Yang	Initial draft of Solution requirements, other considerations and part of Java class design
1.3	21/03/2024	Harry Hainsworth-Staples	Initial draft of 1.2 Analysis
2.0	01/05/2024	Harry Hainsworth-Staples	Update sections 5.1 and 5.2
2.1	04/05/2024	Harry Hainsworth-Staples	Update sections 2 and 4
2.2	08/05/2024	Vanya Liang and Harry Hainsworth-Staples	Update section 7.2 GUI design
2.3	8/05/2024	Soroush Alimohammadi	Update sections 1.1,1.2, 3 and 7.1.1
2.4	8/05/2024	Shuyi Yang	Update sections 7.1.2
2.5	9/05/2024	Shuyi Yang	Finish section 7.3(Diagram/ Text)
2.6	9/05/2024	Jocelyn	Finish section 7.3.1(Text)

Table of Contents

Purpose (Executive Summary)	4
1. Background & Analysis	4
1.1 Analysis Process	4
1.2 Analysis	4
1.3 Scope	7
2. Hardware & Software to be used	8
2.1 Hardware requirements	8
2.2 Software requirements	8
3. References	8
4. Definition of terms	9
5. Solution requirements	10
5.1 Functional requirements	10
5.2 Non-functional requirements	12
6. Other Considerations	15
6.1 Assumptions	15
6.2 Constraints & Dependencies	16
7. Initial requirements analysis of user interaction	17
7.1 UML Diagrams	17
7.2 Initial ideas for GUI	19
7.3 Use cases	32

Purpose (Executive summary)

This document is to be used as an outline of the requirements for designing and implementing a Student Management System for the Dyson Institute of Engineering and Technology (DIET). This is to be referred to throughout the development process as a key reference point in building this application. It provides an outline of the planning and research in designing the application, the scope of functionality to be included, and the design plans and diagrams that provided the direction for implementation.

This document will use the brief, as set out by Dr. Stuart Nicholson (on behalf of DIET), as a primary reference point (in terms of requirements) to create a Student Management System for the use of students and staff enrolled and working at DIET. As an educational institute still in its infancy, the application is aimed to be scalable, and reduce the complexity of managing an ever-growing number of students and staff. As stated in the project brief, the primary purpose of the application act as an all-in-one student management system, to aid in daily academic business, with both student and staff interfaces.

1. Background & Analysis

1.1. Analysis Process

The team was tasked to develop a student management system for the Dyson Institute of Engineering and Technology (DIET) – to provide the registered students and staff in DIET with an interface with multiple functionalities as requested in the project brief (See section 1.3).

From the beginning of the project a consensus was formed within the team that a similar system to the student management system at Newcastle university would be able to address all the requested requirements outlined in the project brief. Therefore, it was necessary to conduct a thorough analysis of the student management system within Newcastle University and assess its capabilities against the requested criteria in the brief.

1.2. Analysis

In order to conduct the analysis into the existent student management system at Newcastle University, two team members were asked to perform a variety of different tasks via their university accounts that would resemble the requested criteria in the brief. The two team members were both instructed to use thinking aloud method and spontaneously report and share everything that would go through their minds and share their thought process while completing the tasks (Ericsson, 2017). To elaborate, this process allowed the design team in the project to gain a deep understanding into users' behavior, preferences, and decision-making patterns. Moreover, it enabled designers to identify potential pain points, misunderstandings, or areas for improvement within the system which ultimately led to more user-centric and intuitive design solutions for final DIET student management system.

The full list of all the instructions that team members needed to perform is as followed:

- Log into your university account and find the programme you are currently enrolled in
- Navigate through the system and find your registered modules
- Show us how you would check your timetable
- Check your obtained marks for any of your module of choice that you have received the results back
- How would you submit for an extension in a module

In the experiment, when the team members were asked to show their registered modules, one student used the Newcastle University's Student Self Service Portal (S3P) while the other student used CANVAS – an online learning management system deployed by Newcastle university.

Canvas Dashboard (Left Image): The dashboard shows a grid of module tiles. Visible modules include: Software Development Techniques (CSC8012), Professional skills and Portfolio (CSC8013), Software Development Advanced (CSC8014), Cybersecurity (CSC8015), Advanced Programming (CSC8016), Database Systems (CSC8017), Web Technologies (CSC8018), Software Engineering and Team Project (CSC8019), and Computer Networks (CSC8021).

Newcastle University S3P (Right Image): The S3P portal displays a list of modules. The 'Module detail for 2023' table is shown below.

Code	Description	Status	Type	Semester 1	Semester 2	Semester 3
CSC8011	Introduction to Software Development	Approved	Compulsory	10	0	0
CSC8012	Software Development Techniques and Tool	Approved	Compulsory	10	0	0
CSC8013	Professional skills and Portfolio	Approved	Compulsory	10	0	0
CSC8014	Software Development Advanced Techniques	Approved	Compulsory	0	10	0
CSC8015	Cybersecurity	Approved	Compulsory	0	10	0
CSC8016	Advanced Programming	Approved	Compulsory	0	10	0
CSC8017	Database Systems	Approved	Compulsory	10	0	0
CSC8018	Web Technologies	Approved	Compulsory	10	0	0
CSC8019	Software Engineering and Team Project	Approved	Compulsory	0	20	0
CSC8021	Computer Networks	Approved	Compulsory	10	0	0
CSC8022	Human Computer Interaction	Approved	Compulsory	0	10	0
CSC8029	Project and Dissertation for HNC/Computer	Approved	Compulsory	0	0	60
BU6416	Writing for PG in Computing Sci Part A	Approved	Free Search	0	0	0
BU6426	Writing for PG in Computing Sci Part B	Approved	Free Search	0	0	0
Totals:				60	60	60
Grand Total:				180		

Figure 1: List of registered modules on CANVAS (left image). List of registered modules on S3P (right image).

However, unlike the previous example when two different approaches were taken to achieve to the same outcome, both students ended up using the same platform of Newcastle E-learning Support System (NESS) when they were asked to check their grades for a module

NESS Interface: The interface shows a sidebar with navigation options like Admin, Module, Coursework, and Project. The main content area displays 'Module Summary' for CSC8017 (Database Systems).

Module Summary (CSC8017) - Module Summary:

Type	Mark	Feedback
Examination		
Coursework		
Practical/lab report 1 - 50%		
Coursework 1 - 50%		View Feedback
Design/Creative proj 1 - 50%		
Coursework 2 - 50%		View Feedback

Figure 2: An example of accessing marks for a module on NESS

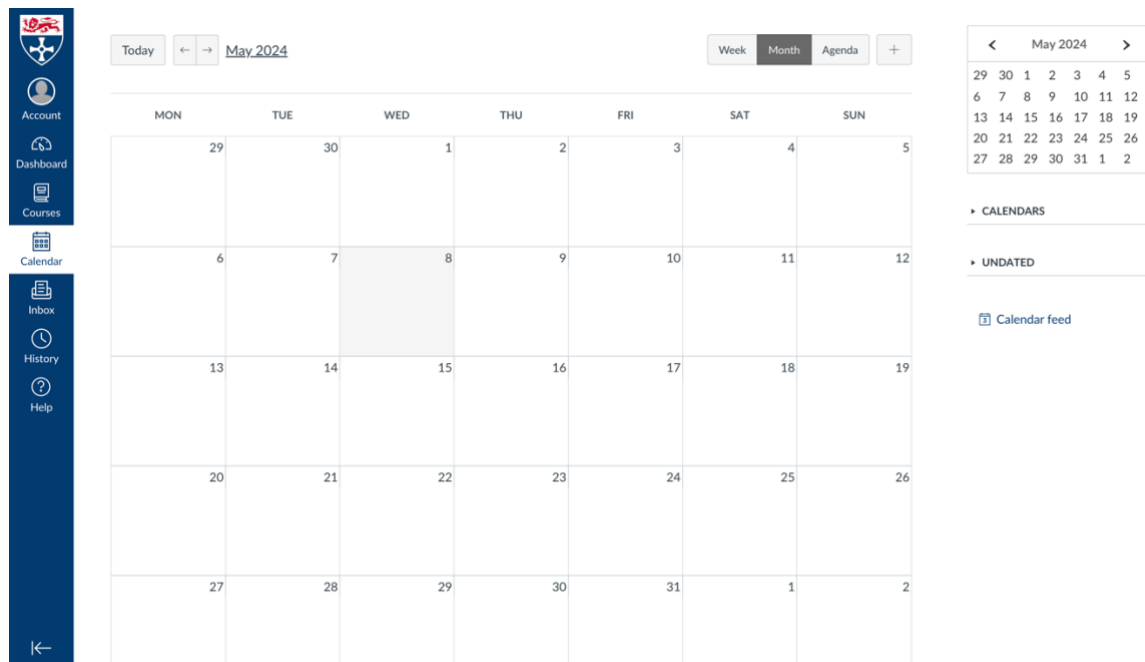


Figure 3: An example of accessing the timetable through CANVAS

As a result of the previous observations, the following suggestions were taken into consideration that would allow us to design a system for the DIET according to their needs:

- In order to limit the overcomplication, only one website would be implemented that would include all the functionalities specified for both students and staff in one platform.
- Due to the general positive feedback received regarding the simplicity of the user interface systems, a similar website design was proposed that would have a menu with all the available functionalities enabling users to access different functionalities within the system
- Similar to the Newcastle University systems, use of more neutral colours within the system was preferred over bright and vibrant colours

Lastly, the aforementioned suggestions were all obtained from the students' point of views and therefore without access to the Staff side at Newcastle University, we had to design the staff portal based on our assumptions.

Stakeholders

The main stakeholders are the end users of the system that will be using this software every day and are most directly affected by the system. In this case the direct users are:

- The students at DIET
- The staff at DIET

Therefore, it was very important to consider the usability and heuristics of GUI design when designing this system to enhance the end user experience.

Additionally, the management at DIET and their parent company Dyson Limited, would be considered the main clients for whom this Student Management System will be developed.

1.3. Scope

In Scope:

- Should allow login functionality with username and passwords to a unique user account with different privilege levels dependent upon if they are student or staff
- Should allow students the ability to view their own personal timetable as a weekly schedule
- Should allow students the ability to view the programme they are enrolled in including access to programme information
- Should allow students the ability to view modules enrolled in and all the module information, such as syllabus, module teaching materials, and grading methods/results
- Should allow students the ability to view any coursework or exams they have been set along with the deadline
- Should allow students to view their grades for past exams and coursework
- Should allow students the ability to submit an extension to their tutor for coursework that has been set
- Should allow students to send a notification to their tutor making them aware of any absences (including time of absence, length of absence, and reason for absence)
- Should allow the student to book a meeting with their respective academic tutor
- Should allow students the ability to take actions over their modules/programme, these actions include - enrolling, suspending, withdrawing, and registering
- Should allow staff the ability to build and amend student timetables
- Should allow staff the ability to view the modules a student is enrolled in
- Should allow staff the ability to access and view a student's academic history - this includes exams and coursework undertaken, grades received, and attendance record
- Should allow for up to 15 students to be assigned to a member of staff as their academic tutor

Out of Scope:

- Auto-timetabling - timetables can be set up by staff as a repeating week for a student that can be altered by staff but will not be auto-generating or auto-changing week on week
- Locations - students will be able to check times for their timetables but assigning physical locations/rooms and room bookings is out of scope
- Security - security in terms of implementing cryptography and ensuring that the overall system that DIET deploy is secure, free of cyber-attack vulnerabilities, and penetration tested
- Data Protection - ensuring that the system complies with Data Protection laws is out of scope for this project.
- Emails - functionality to send emails reminders or for authentication is out of scope
- Load testing and concurrency optimisation

2. Hardware and software platforms to be used for developing and running your solution

2.1 Hardware Requirements

2.1.1 Development Hardware

The server will be run on an Azure Virtual Machine running on Linux running x64 architecture VM size Standard_B1s with 1 vCPU and 1 GiB of RAM and 30GiB OS disk.

2.1.2 End-user Hardware Requirements

To be able to access the application end users require:

- ⇒ Hardware to support modern browsers minimum of 1Ghz CPU, 512MB RAM, and 200MB disk space.
- ⇒ I/O devices such as mouse/touchpad, keyboard and monitor to interact with the application.
- ⇒ A stable internet connection

2.2 Software Requirements

2.2.1 Overall Development Software Requirements

- ⇒ Git (current latest version 2.45.0) for version control and remote collaborative development
- ⇒ Text Editor or IDE:
- ⇒ VSCode or Brackets for frontend
- ⇒ IntelliJ for backend (recommended)

2.2.2 Backend Software Requirements

- ⇒ Using a local MySQL database version 8.0 or above
- ⇒ Spring Boot framework version 3.2.5 using Apache Tomcat 10.1 to handle HTTP requests.
- ⇒ Java version 17 using JDK version 17.0.8
- ⇒ Maven as dependency manager version 3.6.3 or later
- ⇒ JUnit 5 for testing

2.2.3 Frontend Software Requirements

- ⇒ Using HTML, CSS, and vanilla JavaScript to build the dynamic web pages
- ⇒ Using the Fetch API to interact with the Spring Boot server
- ⇒ Using Node.js version 20 or above as a runtime environment and dependency manager (npm)
- ⇒ Jest version 29.7 for testing

3. References

CANVAS Newcastle University. Available at: <https://canvas.ncl.ac.uk/login/> (Accessed: 01 May 2024).

Ericsson, K.A., 2017. Protocol analysis. A companion to cognitive science, pp.425-432.

NESS Newcastle University. <https://ness.ncl.ac.uk/> (Accessed: 01 May 2024).

Newcastle University's Student Self Service Portal. Available at: <https://s3p.ncl.ac.uk/login/welcome.aspx> (Accessed: 03 May 2024).

Nielsen, J. (1994, updated Jan. 30, 2024) *10 Usability Heuristics for User Interface Design*. Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/> (Accessed: 4 May 2024).

Nielsen, J. (2010) *Website Response Times*. Available at: <https://www.nngroup.com/articles/website-response-times/> (Accessed: 5 May 2024).

4. Definition of terms

Apache Tomcat - free and open-source, it provides a “pure Java” HTTP web server environment in which Java can also be run.

API - stands for Application Programming Interface and is the way for two or more computer programs or components to communicate with each other using requests and responses.

Azure - Microsoft's cloud computing service, it offers access, management, and the development of applications and services through global data centers.

Brackets - a free and open-source, source code editor developed by Adobe Inc.

CPU - Acronym for Central Processing Unit (often thought of as the brain of a computer), its electronic circuitry executes the instructions of a computer program such as arithmetic, logic, controlling and input/output operations.

CSS - stands for Cascading Style Sheets and is a language for specifying the presentation and styling of a HTML document.

GiB - A Gibibyte often used synonymously with Gigabyte however they technically do not describe the same amount of capacity, 1 GiB is equal to 2^{30} bytes (roughly 1.074 billion bytes) whereas 1 GB is equal to 10^9 bytes (exactly 1 billion bytes).

Git - is a version control system that tracks changes in any set of computer files.

GUI - stands for Graphical User Interface and it is a digital interface with which a user interacts with using buttons, icons, and menus.

HTML - stands for HyperText Markup Language and is the standard markup language for displaying web pages.

HTTP - stands for HyperText Transfer Protocol and is the protocol for sending hypermedia documents such as HTML over a network.

IDE - stands for Integrated Development Environment and is software that provides comprehensive facilities for software development.

I/O - shorthand for Input/Output referencing to hardware devices that provide either input (such as a mouse and keyboard) or output (such as a visual display) for your computer. Note: I/O can also be used in reference to software in essence it can be used to refer anything that inputs or outputs information.

Java - a high-level, class-based, object-oriented programming language.

JavaScript - a high-level multi-paradigm language often referred to as the language of the web as most webpages use JavaScript on the client side to dictate web page behaviour.

JDK - stands for Java Development Kit and is a distribution of Java technology from the Oracle Corporation.

Jest - a JavaScript testing framework.

JUnit - a test automation framework for the Java programming language.

Linux - Linux is a group of open-source UNIX-like operating systems based on the Linux kernel originally developed by Linus Torvalds in 1991. There are many different Linux distributions that allow for certain out-of-the-box functionality dependent upon which (distro) you choose.

Maven - a build automation tool and dependency manager used mainly for Java projects.

MySQL - an open-source relational database management system. Using SQL (see SQL) as its querying language storing data in relational tables with rows and columns.

Node.js - a runtime environment for the JavaScript programming language.

OS - shorthand for Operating System, which is the low-level software that supports the main functionality of a computer controlling software and hardware resources and providing common services for computer programs.

RAM - stands for Random Access Memory is a form of computer memory typically used for storing working data needed to open applications and run files.

Spring Boot - an open-source Java framework used to create a micro-service.

SQL - stands for Structured Querying Language a language used for querying, retrieving, inputting, and deleting data from relational databases.

Standard_B1s - Azure's name for a type and size of VM.

VM, Virtual Machine - A Virtual Machine is the emulation of a computer system, often accessed through cloud computing this means you can use remote hardware to emulate a computer system of your choice with hardware specifications of your choosing.

VSCode - Visual Studio Code is free a source code editor developed by Microsoft

5. Solution requirements

5.1. Functional Requirements

NO.	Category	Description	Priority (H, M, L)	Comments
FR1	User Login	Students and Staff can login in with username and password	H	See NFR1

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

FR2	User Login	Users will have different privilege (Student/Staff) levels that will affect functionality on login	H	See NFR1
FR3	User Login	Students will have a unique student ID as username	H	See NFR1
FR4	User Login	Staff will have a unique staff ID as username	H	See NFR1
FR5	User Login	Users can change their passwords	H	See NFR2
FR6	Student	A student can view the course and modules they are enrolled in	H	See NFR3
FR7	Student	Students will be able to enroll, defer or withdraw from a course	H	See NFR4
FR8	Student	Students will be able to view the contents of a module they are enrolled in	H	See NFR5
FR9	Student	Students will be able to view their personal weekly timetable	H	See NFR6
FR10	Student	Students will be able to view the marks received from their exams and coursework	H	See NFR7
FR11	Student	Students will be able to view the due date of a coursework	M	See NFR7
FR12	Student	Students will be able to view feedback for an exam or coursework	M	See NFR7
FR13	Student	Students will be able to submit a file as a coursework document	M	See NFR8
FR14	Student	Students will be able to view module announcements	L	See NFR9
FR15	Student	Students will be able to request an extension for a coursework or exam	H	See NFR10
FR16	Student	Students will be able to submit questions about a module	L	
FR17	Student	Students will be able to book a meeting with their tutor	H	
FR18	Student	Students will be able to submit an absence request form	H	See NFR11
FR19	Staff	Staff will be able to create a timetable for a module	H	See NFR12
FR20	Staff	Staff will have the ability to modify a student or modules timetable	H	See NFR12
FR21	Staff	Staff will be able to view the modules undertaken by a Student	H	See NFR13
FR22	Staff	Staff will be able to set a coursework or exam for a module	M	
FR23	Staff	Staff will be able to add announcements to the announcement board for a module	M	

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

FR24	Staff	Staff will be able to answer student questions	L	See FR16
FR25	Staff	Staff will be able to see the grades achieved/ academic history of a Student	H	
FR26	Staff	Staff will be assigned up to 15 Students to be their academic tutor	H	
FR27	Staff	Staff will be able to approve/deny a Student absence request	M	See FR18
FR28	Staff	Staff will be able to approve/deny a Student extension request	M	See FR15
FR29	All users	Users will be able to use a voice over assistant for accessibility	L	
FR30	All users	Users will be able to report any bugs or errors that occur	L	

5.2. Non-Functional Requirements

NO.	Description	Priority (H, M, L)	Comments
NFR1	<p>Users will need to enter the following information to login:</p> <ul style="list-style-type: none"> ⇒ Unique username (student/staff number) ⇒ Password (twice during registration) ⇒ Press login button <p>The input will be checked for any illegal characters/ user inputs</p> <p>This will then be passed over and validated by the Spring Boot Rest API verifying against user data stored In the MySQL database.</p> <p>User feedback as a popup will be provided upon failed or successful login attempts.</p> <p>If successful it will load the landing page of the application dependent upon user privilege levels. (Student or Staff)</p>	H	Satisfies FR1-4
NFR2	<p>Users can change their password by:</p> <ul style="list-style-type: none"> ⇒ Selecting the “Forgot password?” link ⇒ Inputting id, email address and new password ⇒ Clicking the update button <p>This will validate the user with unique id against email address and send a POST request the Spring Boot Rest API to update the user's password.</p> <p>The user will have a visual confirmation as to the success of the update and if successful will be redirected back to the login page</p>	H	Satisfies FR5

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

NFR3	<p>Students will be able view the modules they are enrolled in by navigating to the “Subjects” page.</p> <p>Modules a student are enrolled in will be stored in the MySQL database. Navigating to the page will trigger a GET fetch request to the Spring Boot server which will retrieve all the modules connected to the Student’s ID.</p>	H	Satisfies FR6
NFR4	<p>Students will be able to enroll, suspend or withdraw from a module by clicking buttons of the same name on the “Subjects” page.</p> <p>If a student selects one of these options enroll will send a POST request to the Spring Boot server to add a module relationship to their Student ID stored in the MySQL database. Selecting withdraw or suspend will send a DELETE request the server to delete the relationship between a module code and Student ID stored in the database.</p>	H	Satisfies FR7
NFR5	<p>Students will be able to view module contents by navigating to the “subjects” section and choosing the module they wish to view.</p> <p>They can then view the module contents by clicking on “Teaching Material” which will retrieve all uploaded files and documents by module Staff upon page loading.</p>	H	Satisfies FR8
NFR6	<p>Students will be able to view their weekly timetable by navigating to the “Timetable” page.</p> <p>This will send a GET fetch request to the Spring Boot server and retrieve all the events for the modules the student is enrolled in. This will then display to the screen.</p> <p>The timetable will only show Monday to Friday and each event will be in blocks of 2 hours starting at 9am and finishing at 5pm meaning there will be a possible of 4 events per day.</p>	H	Satisfies FR9
NFR7	<p>Students will be able to view their upcoming coursework assignments for each module they are enrolled in by navigating to the “Examination” section.</p> <p>This will trigger a GET request to the Spring Boot server to retrieve grading methods (coursework/exam) for each module from the database, along with name, due date, and grade (if completed).</p> <p>There will also be a link to view marking comments from a Staff user that will open as a pop-up window. If no comments available the pop-up will state that there are no comments available.</p>	H	Satisfies FR10-12
NFR8	<p>Students will be able to submit a document by navigating to the “Submission” page of the examination section. They will be able to choose from a list of module buttons dependent upon which module they are submitting for.</p> <p>A student will then be able to click the “choose file” button which will bring up the user’s file directory where they can select the file they wish to submit. The name of the file they are submitting will show</p>	M	Satisfies FR13

	<p>next to the “choose file” button. They can then click the button “submit” to submit this as their coursework.</p> <p>Visual feedback will show the status of submission whether successfully submitted for that module or not.</p>		
NFR9	<p>Students will be able to view module announcements that have been posted by staff by navigating to the “subjects” section and selecting the module. This will open on the announcements section which will retrieve all the announcements (through the Spring Boot server retrieving stored announcements in the database) upon the page loading.</p>	L	Satisfies FR14
NFR10	<p>Student will be able to request for an extension by navigating to the examination section and selecting the “Extension” button in the navigation.</p> <p>This will open an input form in which the Student must complete all sections. This will have 3 sections:</p> <ul style="list-style-type: none"> ⇒ Reason for extension (as a check list) ⇒ Details of extension (including date start and date end, and brief description of circumstances behind the need for an extension) ⇒ Module for which it is required <p>This will then be submitted and stored in the database to be approved by the students Personal Tutor. From the information provided the tutor can determine if the reason for extension is valid.</p>	H	Satisfies FR15
NFR11	<p>Students can submit an absence request by clicking on “Personal Tutor” in the navigation. They can then click on “Absence” to bring up the absence request form.</p> <p>To complete the form a Student will need to input:</p> <ul style="list-style-type: none"> ⇒ Reason for absence (selected from a dropdown list) ⇒ Date start and date end of absence (in form dd/mm/yyyy) ⇒ Description as a text input field <p>This will send an email to the personal tutor which they can then authorise or not dependent on validation of the reason behind absence provided by Student.</p>	H	Satisfies FR18
NFR12	<p>Staff will be able to create a timetable by navigating to “Timetable” page.</p> <p>A staff will be able to set a timetable for a particular module by inputting the module they wish to edit.</p> <p>The Staff will then be able to select “Show Existing” to show what is currently timetabled for the module (if anything).</p> <p>For simplicity the timetable will be split up into a grid distinct time blocks from Monday to Friday (see NFR6 for more details). The Staff can click on this block to add a timetable event along with Staff member that will be taking the session.</p>		Satisfies FR19-20

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

	Once editing is finished the timetable can be published by selecting "Submit Timetable" which will overwrite the current timetable stored in the database. A popup message will give feedback as to the success of this action.		
NFR13	Staff will be able to retrieve modules undertaken by navigating to the "Undertaken Modules" page. They will be able to input a student number and click, which will send a request to the server and retrieve all the modules a student has undertaken. This will display to the screen as a list of module codes and module names.	H	Satisfies FR21
NFR14	Staff will be able to view a Student's academic history by navigating to the "Student Details" page. They can then input the student number and click submit which will send a request to the server and retrieve all the grades achieved per module by the student. This will display as a list of module and grade achieved.	H	Satisfies FR25
NFR15	Any system updates must occur at midnight when there will be least user traffic, and therefore least disruption.	M	
NFR16	The navigation of the system should follow Nielsen's time limits for page loading with a maximum of 1s of page load time to ensure the user feels in control of their experience and navigation still feels seamless. (Nielsen, 2010)	M	
NFR17	The system should satisfy Nielsen's 4 th heuristic of Consistency and Standards (Nielsen, 1994). This will be accomplished by: ⇒ Ensuring consistent navigation between pages by using navbar and hamburger pop out list navigation ⇒ Consistent theme throughout application	L	
NFR18	The system should satisfy Nielsen's 8 th heuristic of Aesthetic and Minimalist design (Nielsen, 1994). This will be accomplished by: ⇒ Ensuring consistent design between pages ⇒ Focusing only on the essential information that needs to be displayed (relating to FRs) ⇒ No use of confusing extraneous details	L	
NFR19	The system should satisfy Nielsen's 5 th heuristic of Error Prevention (Nielsen, 1994). This will be accomplished by: ⇒ Thorough unit and user testing to reduce the number of errors that may occur ⇒ Providing useful feedback when an error does occur	H	

6. Other considerations

6.1. Assumptions

- The application/web of this system is only accessible to Dyson employees who are based in the UK region and have been employed. The primary users of this system are employees who are mainly external tutors or senior staff, familiar with and have used similar types of information management systems.

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

- The client is Dyson UK, and the application is only allowed for local use in the UK and cannot be used in other regions. It is a student information management system under the Dyson umbrella. The application will protect user data from being leaked, and users cannot directly access the database, but they have the right to share their own information or learning-related content.
- Since we are specifically developing the Dyson Institute project for Dyson UK, we have estimated the peak number of visits, types of programs, and course designs based on the number of employees and internal organizational structure of Dyson UK. The estimated number of people is approximately 200, with programs including product design, hardware engineering, and software engineering, and modules designed for each of these programs.
- There may be intersections in module designs for different programs, such as HCI modules in both the product design and software engineering programs' outlines, and multiple computer-related courses intersecting between hardware engineers and software engineers. Additionally, in the concept design of elective modules, we do not establish separate study programs but only offer individual modules for everyone to choose from. Furthermore, students from one program can choose mandatory modules from another program as electives. In the database setup, there will be attributes under the module entity to determine whether it is an elective module or not.

6.2 Constraints & Dependencies

- Maintenance of the system application is the responsibility of the client. After delivering the product and successful testing, the system will be fully handed over to Dyson UK for management. Our team will only provide technical support and functional updates.
- The terminal servers running the application will ultimately be provided by Dyson. The deliverables for this delivery only include testing of the relevant hardware and software mentioned in the technical background section and can only guarantee optimal performance under existing conditions. Therefore, the performance tested in this round does not represent the final performance of the program, and there may be instability during final delivery testing. Final performance confirmation should be conducted on servers provided by the client, and a release version will be provided.
- To avoid breaching any regulations, the system will not store or process any personal identity data. Login credentials will consist of a username, nickname, and password to ensure privacy and security. Maintenance of the database will also be handled by Dyson's relevant technical personnel.

7. Initial requirements analysis of user interaction

7.1 UML diagrams

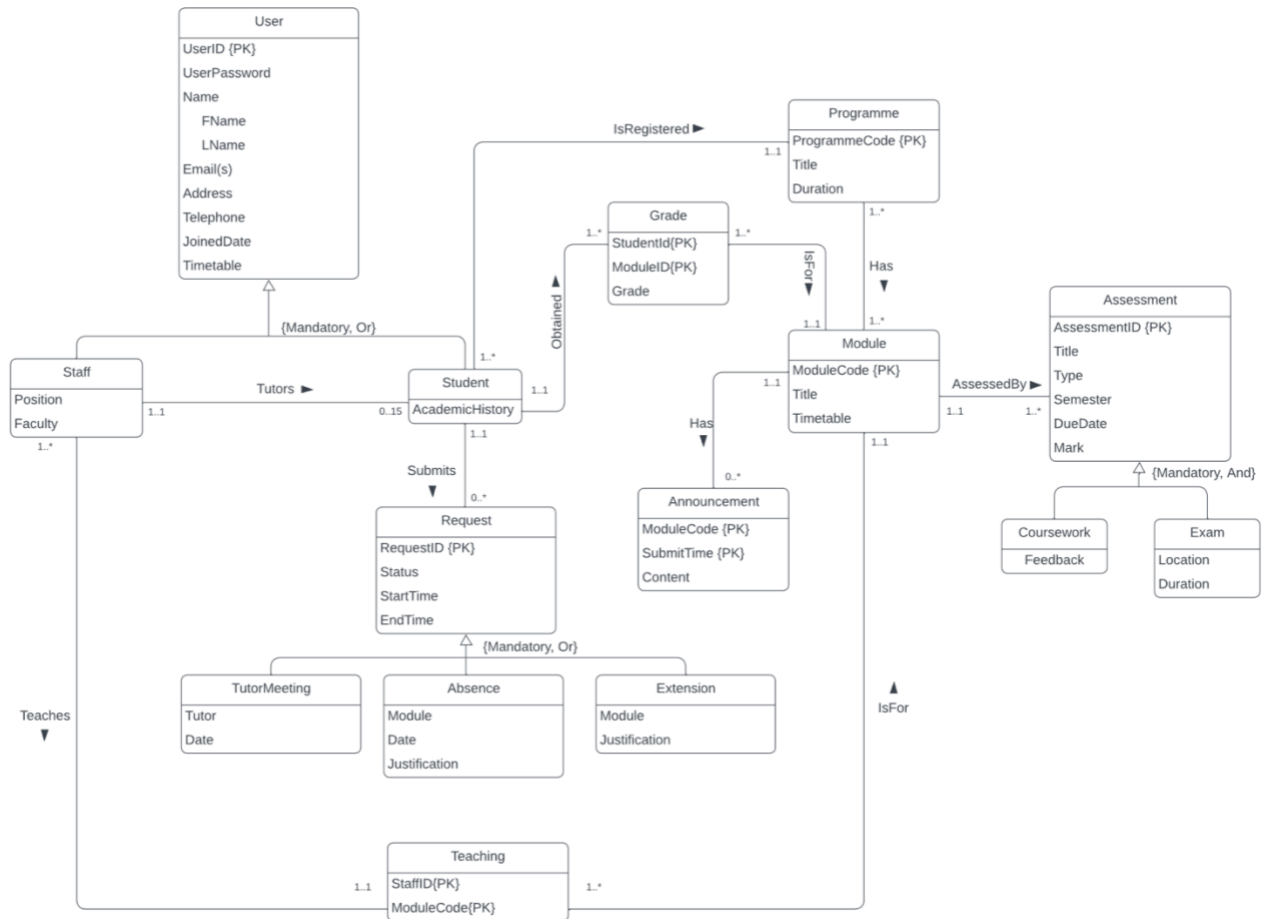


Figure 7.1.1 showing the UML entity relationship diagram of the database design.

7.2 Initial ideas for a GUI.

7.2.1 Version 1 of design

Version 1 of the student web pages design just about some basic functions, like login pages, to see which modules that a student has, a student can download the teaching materials and recaps. Also, can check the coursework result. What's more, can use side panel to switch pages.

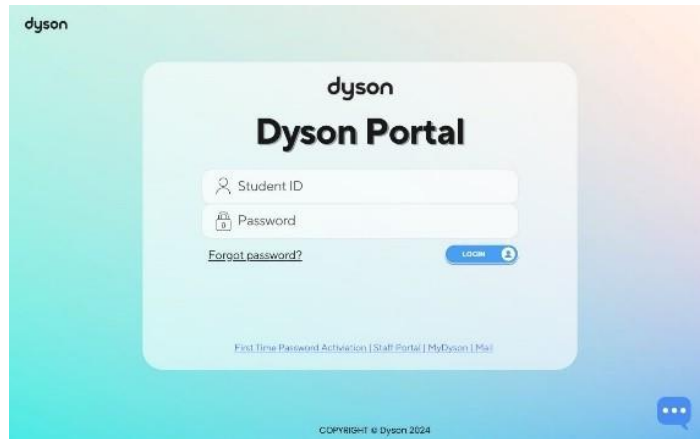


Figure 7.2.1.1 Student Login Portal

Student login page

This is the landing page concept for our system. Students need to use their StudentID and Password to login system.

Hyperlink areas:

- + "Login" button: When the input data matches the data in the database, it will jump to another page.
- + "forget password" link: send an email to student's mailbox to reset the password.

Bottom:

- + "First Time Password Activation": send email to students' mailbox to activate their account.
- + "Staff Portal": jump to Staff login page.
- + "MyDyson": jump to Dyson's website.

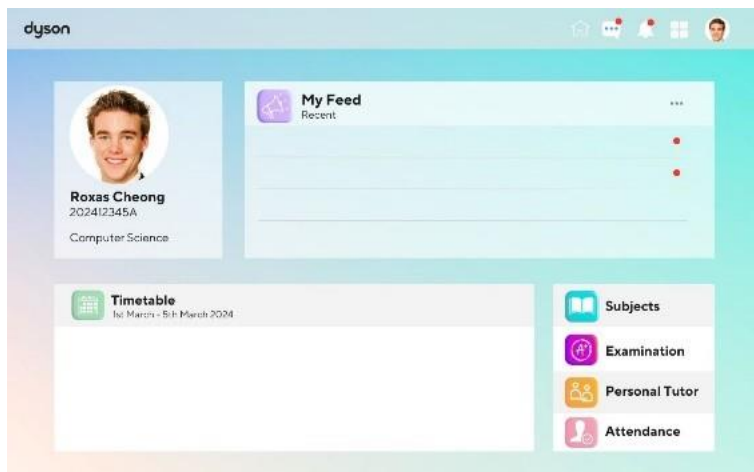


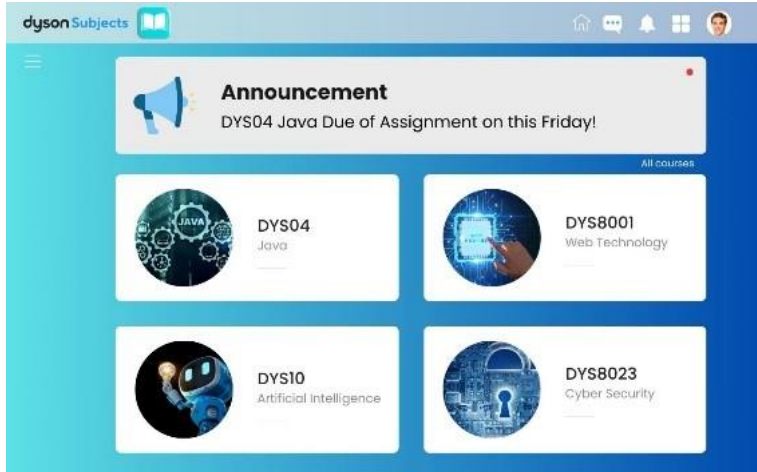
Figure 7.2.1.2 Student Dashboard

Student main page

Functionally, students can choose the modules after they login.

Hyperlink areas:

- + The profile block contains the student's profile photo, student number, and department.
- + The timetable block is viewing the semester's class schedule.
- + Announcement block is showing any details about their subjects.
- + Module block is checking their Coursework Result.
- + Attendance/ student survey block is to notify tutor of absence / booking meeting with tutor.



Modules pages

This page shows all the modules that the student has taken during the semester.

Hyperlink area:

- + On the right side of the page, students can check every detail about their subjects.
- + On the left side of the page, is a navigation bar. Including, the notices sent by lecturers, the upcoming due assignments, teaching materials, discussion board.

Figure 7.2.1.3 Module Page

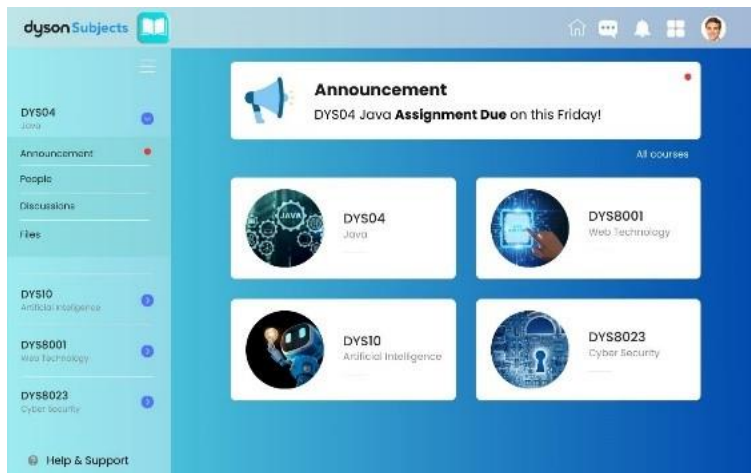


Figure 7.2.1.4 Modules Page with navigation bar

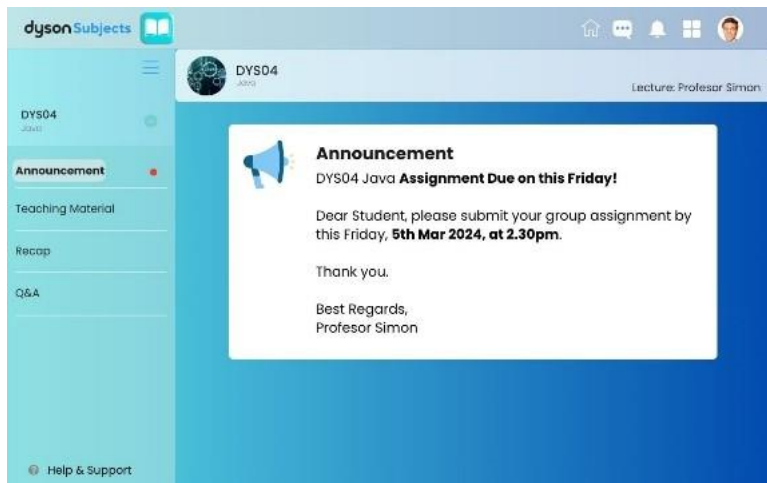
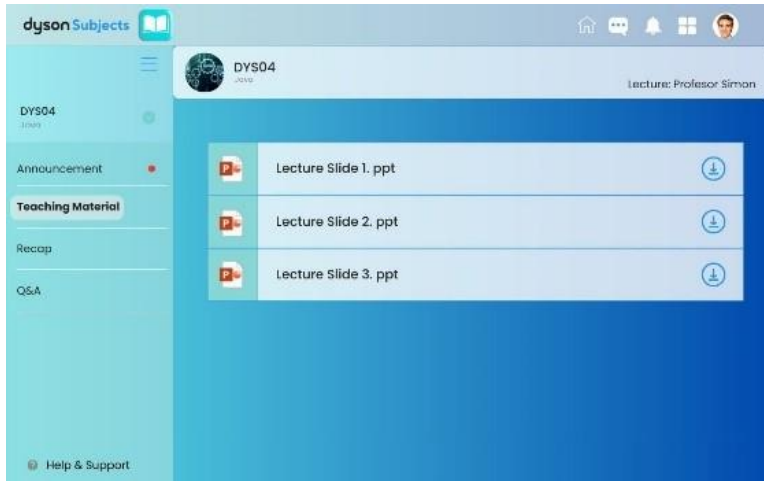


Figure 7.2.1.5 Module Announcement page



Module Information

This page shows details in "Subject 1", it includes Lecture slides and other materials that students can download on their devices.

+ On the right side of the button - "grid/list", Click it to convert the arrangement layout.

Hyperlink area:

+ On the left side of the page, is a navigation bar. Including, the announcement of lecturers, teaching materials, recap, Q&A with professors.

Figure 7.2.1.6 Module Teaching Material page

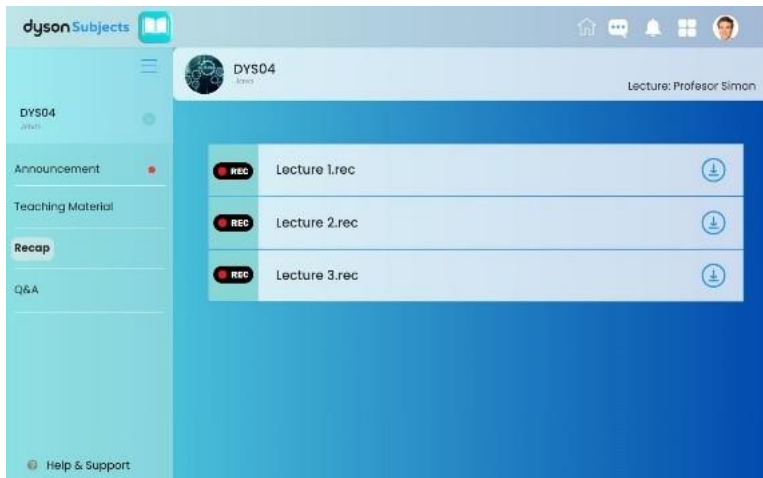


Figure 7.2.1.7 Module Recap page

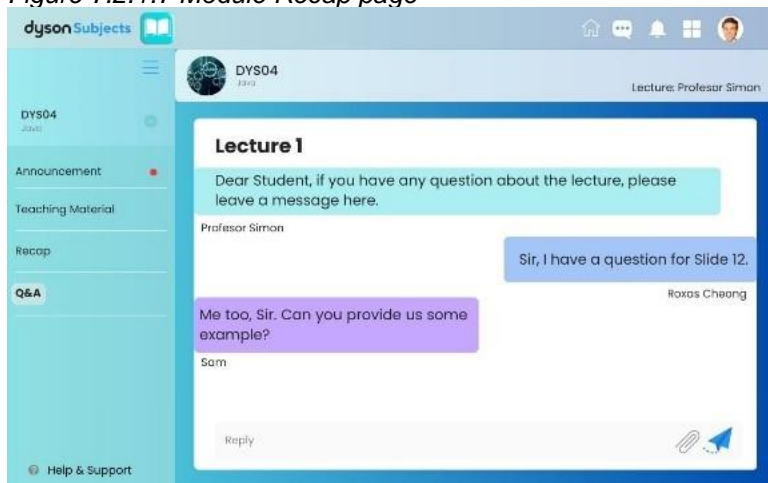


Figure 7.2.1.8 QnA with module staff page



Student coursework result page

This page gives students assignment submissions, apply for assignment extensions, checking exam times and exam locations.

Hyperlink area:

Middle area:

+ submission button: submit assignments
+ extension button: apply for an extension for coursework.

Bottom:

+ Feedback button: for checking the feedback from lecturers.

+ Summary button: We haven't finalised the features yet.

+ Performance: We haven't finalised the features yet.

Figure 7.2.1.9 Coursework Results page

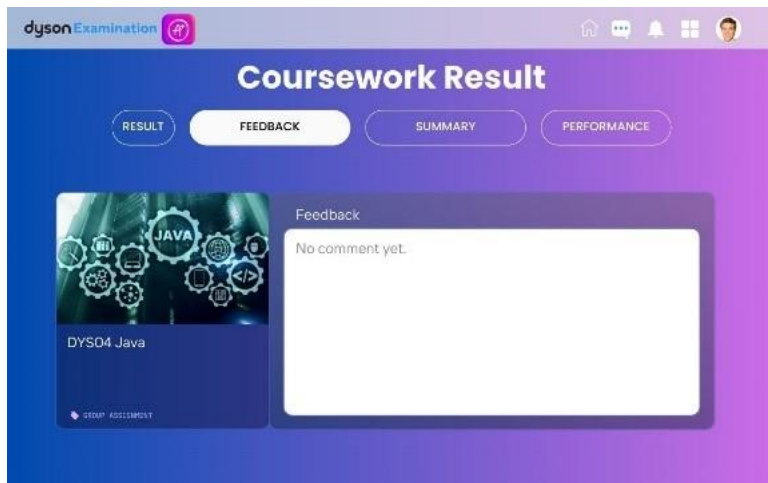


Figure 7.2.1.10 Coursework Result Feedback page

Version 1 of the staff web pages design. There are only a few sketches of the staff system, but they are all basically based around the requirement document. It includes pages for the login page, main page, timetable, viewing student information, and pages for viewing students assigned to teachers.



Figure 7.2.1.11 Staff Portal

Staff login page

This is the staff's login page, and the link at the bottom takes you to the student's login page. Most of the functionality is like the student side.

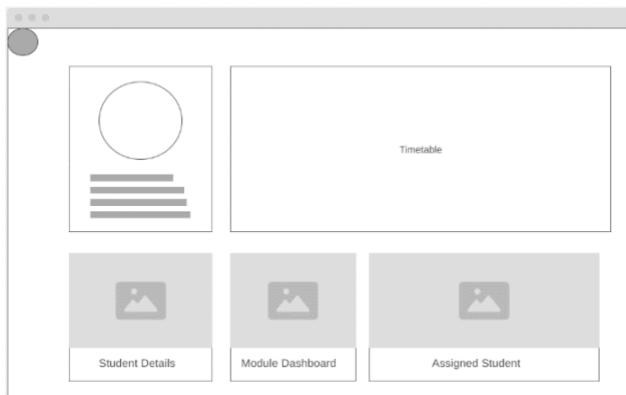


Figure 7.2.1.12 Staff Main Dashboard

Main Dashboard page

On this page, employees can see their class schedules, personal information about their students, the courses they teach, and the students they are assigned to.

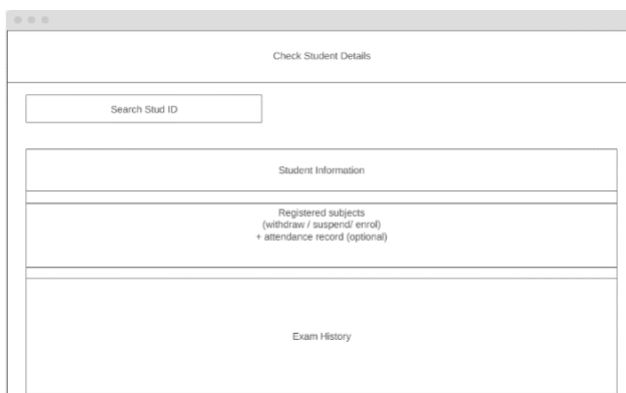
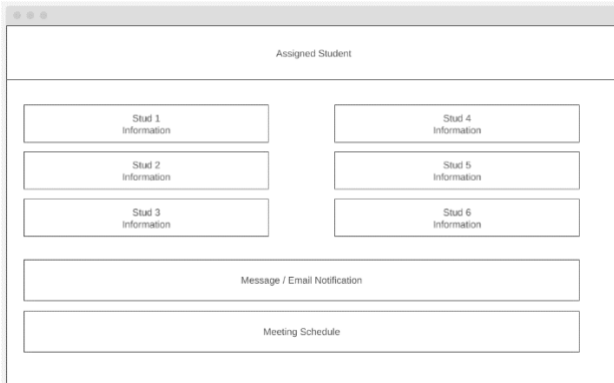


Figure 7.2.1.13 Student Details page

Student Details page

On this page, the staff can search for the student's student number to get his/her personal information, enrolled majors, courses and attendance, and exam information.

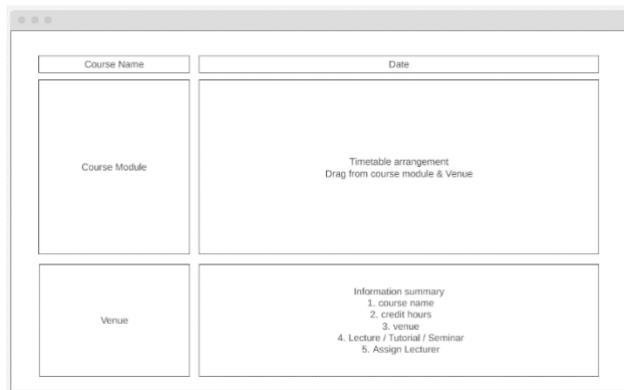
This data may be obtained from the database.



Assigned Student page

On this page, staff can view the personal information of their assigned students, as well as make appointment

Figure 7.2.1.14 Assigned Students page



Timetable page

This page is for assigning courses to students, with specific information including location, course module, etc., and two tables to preview the scheduled courses.

Figure 7.2.1.15 Timetable page

7.2.2 Version 2 of the design.

In Version 2 of the student system, we have added some features and streamlined some pages based on the foundation established in Version 1. These changes make the student system more logical and enhance the user experience for students. I will discuss the design modifications in Table 7.2.3.

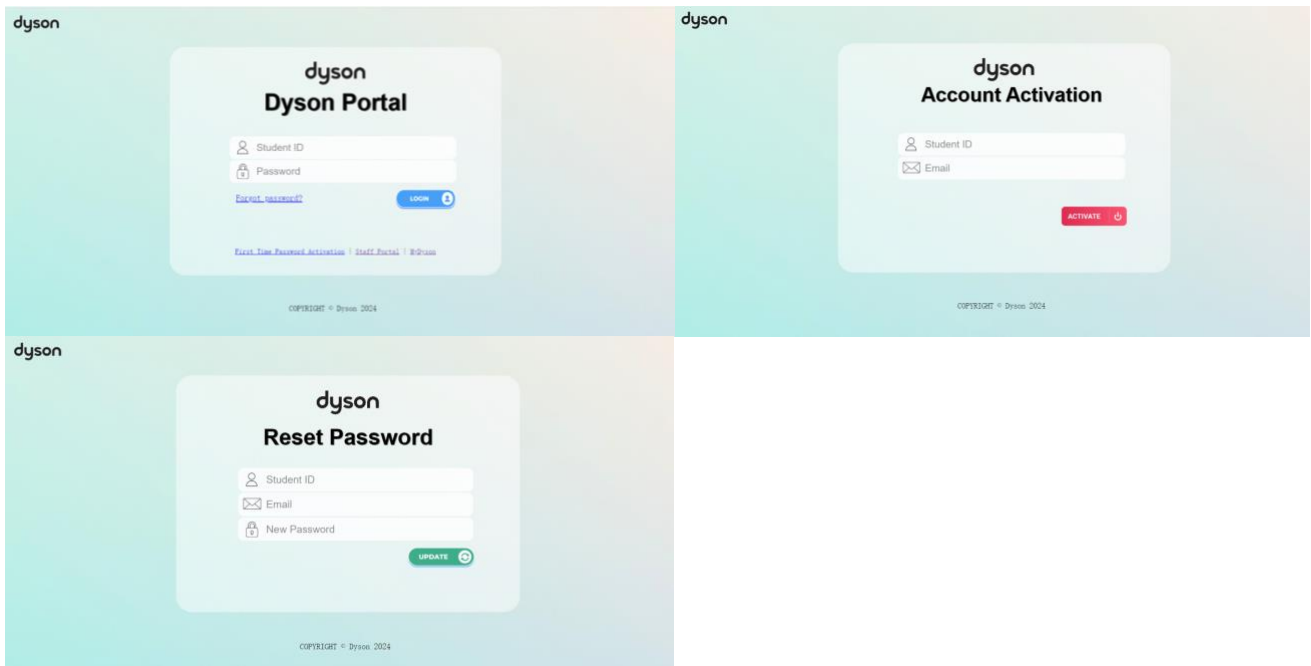


Figure 7.2.2.1 Updated Student Login Portal Pages (Top left - Login, Top Right - Account Activation, Bottom Left- Reset Password)

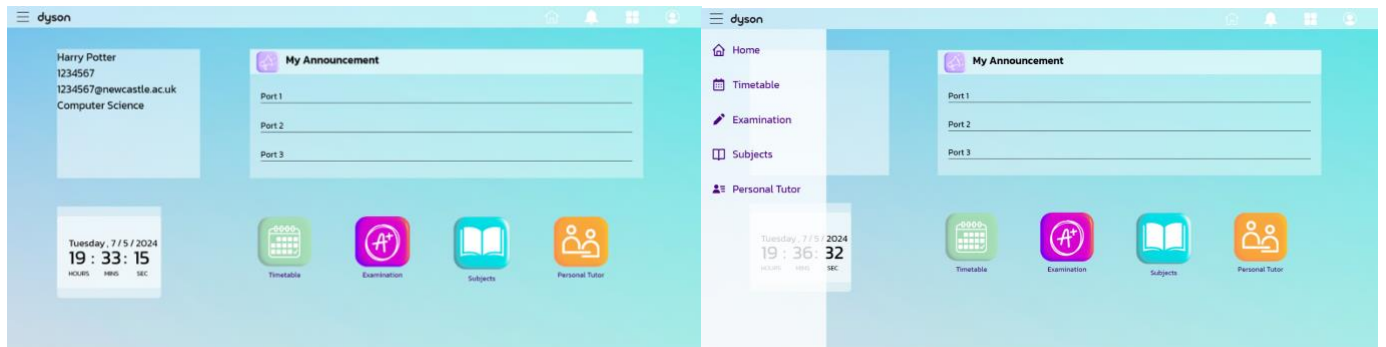


Figure 7.2.2.2 Updated Student Dashboard (Left without navigation bar, Right includes navigation bar)

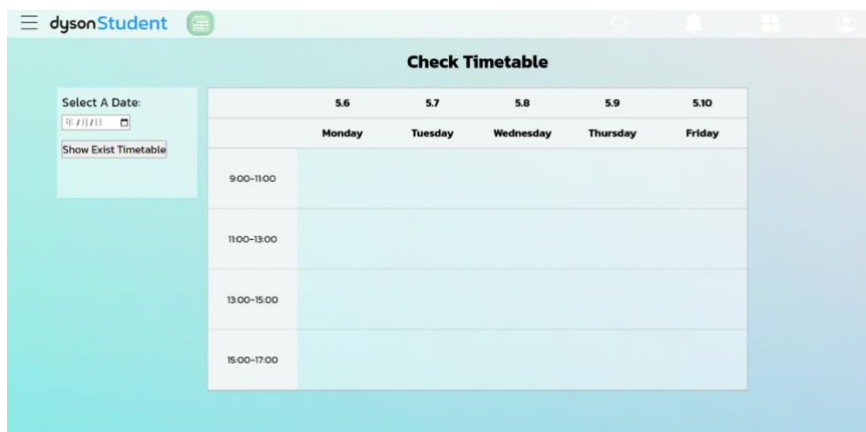


Figure 7.2.2.3 Student Timetable page

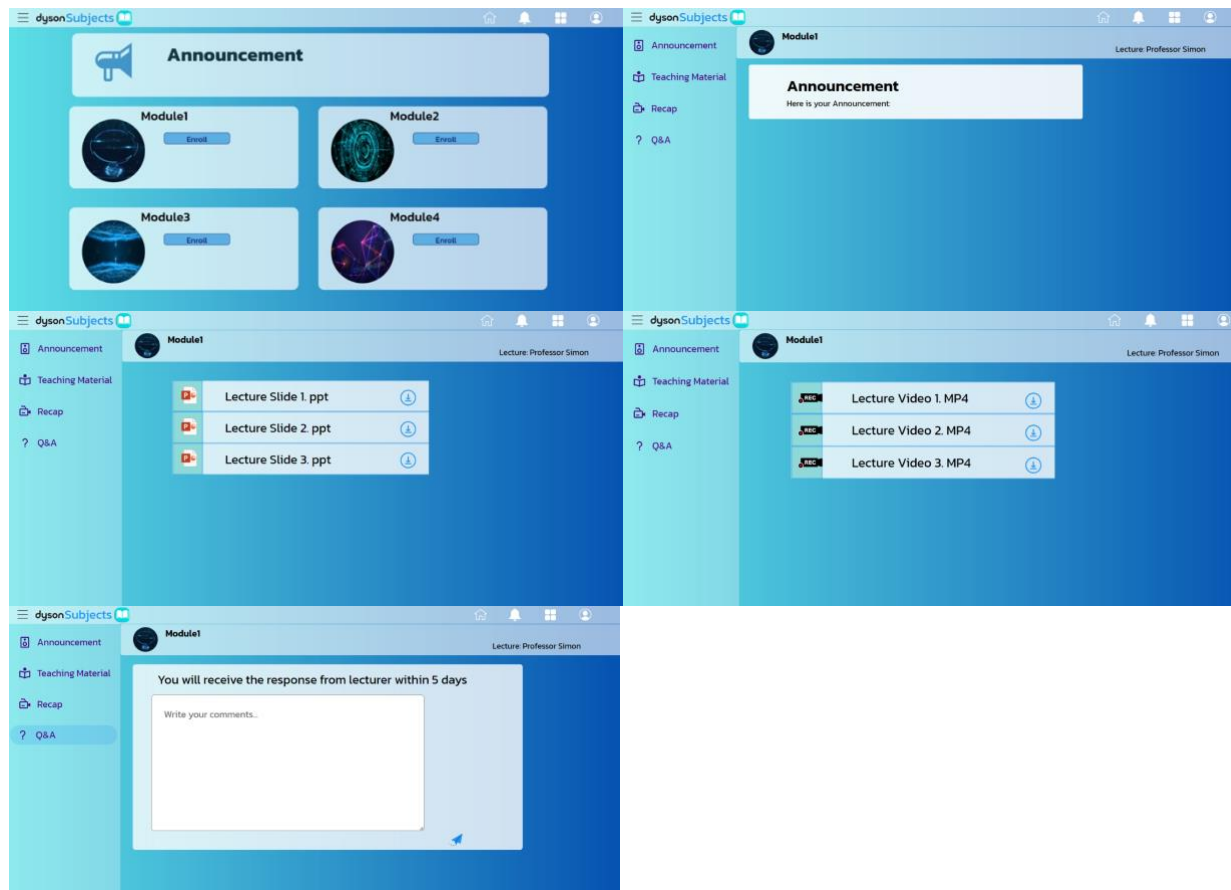


Figure 7.2.2.4 Updated Student Module Section (Top Left - Choose Module, Top Right - Module Announcements, Middle Left - Module Content, Middle Right - Module Recap, Bottom Left - QnA)

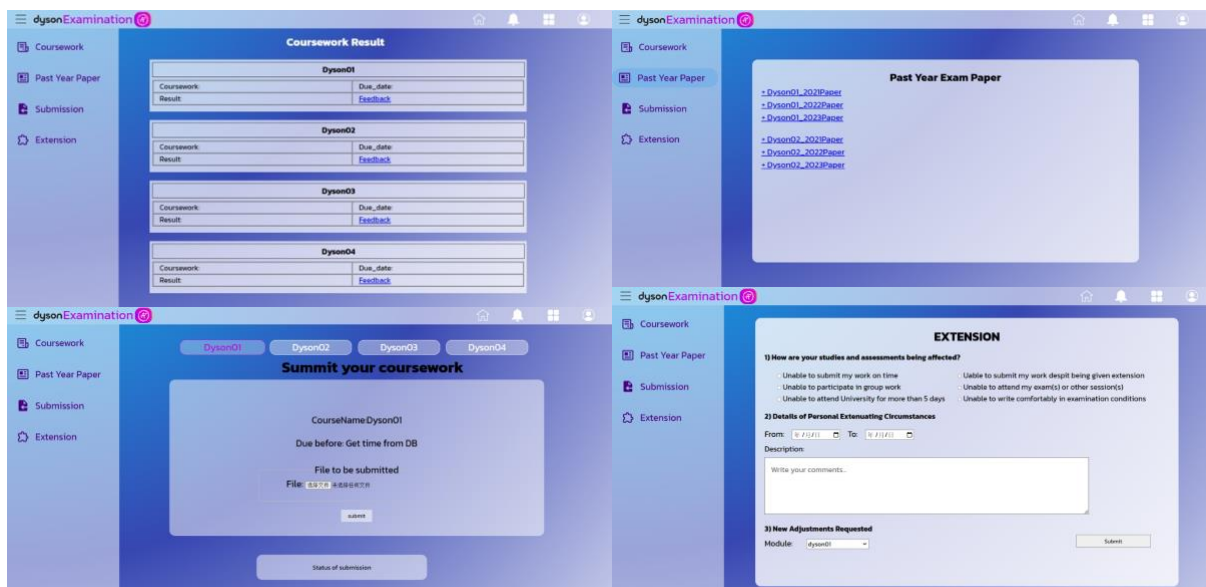


Figure 7.2.2.5 Updated Examination section (Top Left - Coursework Results, Top Right - Past Papers, Bottom Left - Coursework Submission, Bottom Right - Extension request)

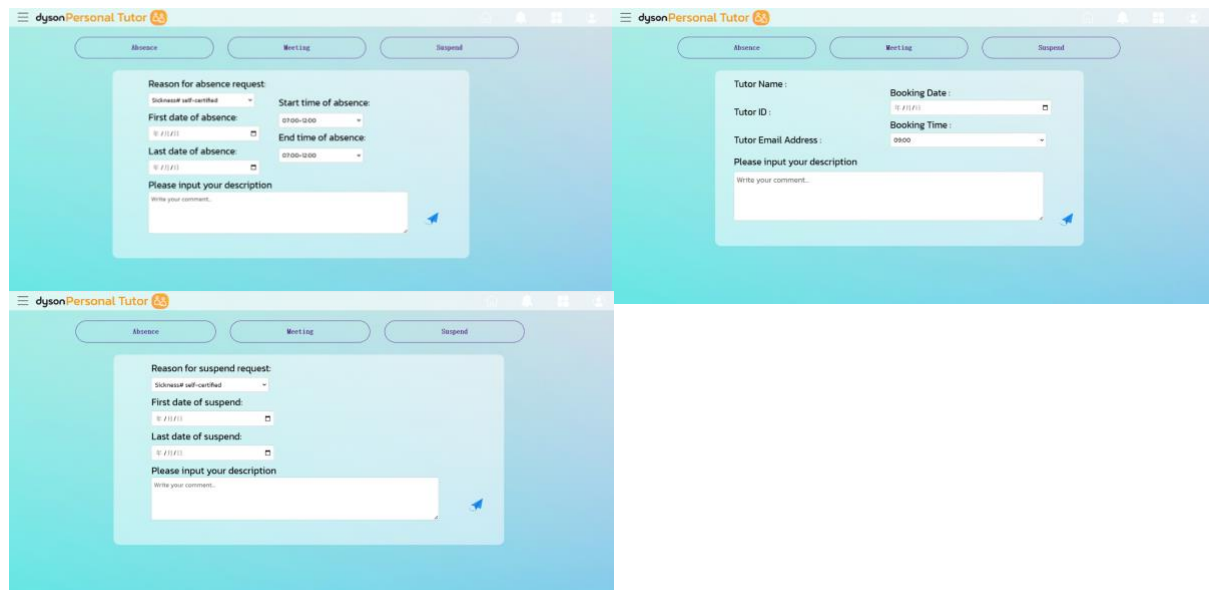


Figure 7.2.2.6 Updated Personal Tutor section (Top Left - Absence request, Top Right - Meeting Request, Bottom Left - Suspend request)

In Version 2 of the staff page, we did not strictly adhere to the design sketches. We simplified the interface and added new features: for example, staff members can now upload teaching materials and download student coursework for grading and providing feedback, to align with the functionalities available on the student side. The details of these functionalities will be described in Table 7.2.3.

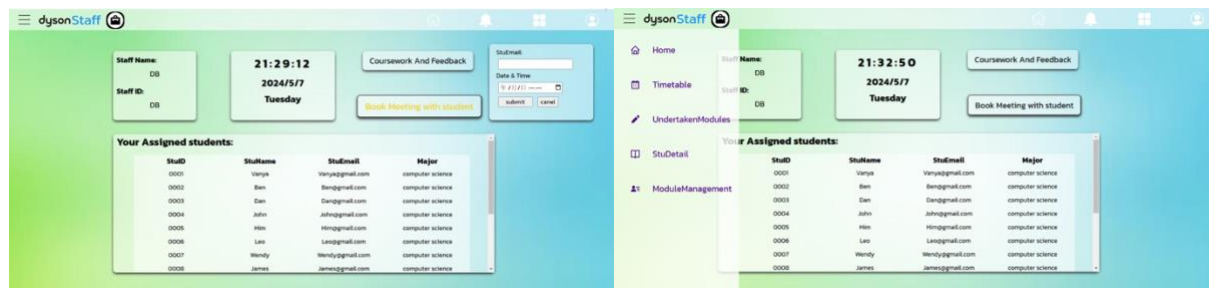


Figure 7.2.2.7 Updated Staff Dashboard (Left without navigation bar, Right with navigation bar)

Assigned Timetable

	4.29	4.30	5.1	5.2	5.3
	Monday	Tuesday	Wednesday	Thursday	Friday
9:00-11:00					
11:00-13:00					
13:00-15:00					
15:00-17:00					

Figure 7.2.2.8 Staff timetable page

Students Coursework:

- + C300000789_Coursework
- + C300000456_Coursework

Select Module: Input Marks: Input StuID:

Input Feedback

Figure 7.2.2.9 Staff Coursework Feedback Page

Student ID: 0001

Detail:

StuID	StuName	ModuleCode	ModuleName
0001	Vanya	Dyson01	Advanced Programming

Figure 7.2.2.10 Staff Check Student Modules page

dysonStaff

Input Student ID:

Student Name: Student Email:

Assessment Detail:

ModuleCode	AssessmentType	AssessmentName	AssessmentGrade
0001	Coursework	Dyson01_TeamProject	70

Student Personal Info:

Academic History

No related BG

Figure 7.2.2.11 Staff Check Student Academic History page

dysonStaff

Select A Module:

Teaching Material

Recap

Announcement

Figure 7.2.2.12 Staff Module Management page

7.2.3 Changes Incorporated Table

No.	Page Name	Major Changes	Minor Changes
1	Login Page (including Student and staff)	Design two more pages about activate accounts and forget password. This kind of design may make more reasonable.	
2	Side Panel		Implement the side panel across all pages, replacing the sub-list of courses with the names of each page. This design is more logical and allows users to navigate to each page through the side panel. The side panels for students and teachers will be adjusted according to their respective functionalities.
3	Student Dashboard		The interface has been simplified. In the top left corner, a box displays the student's personal information, excluding their photo. In the bottom left corner, the timetable has been replaced with the date and time, as it is not

			feasible to display the timetable within such a narrow space.
4	Student Timetable	The interface has been enhanced. After a user selects a date and clicks the button, their class schedule for that date will appear.	
5	Subject Dashboard	An 'Enroll' button has been added, allowing users to enroll in the course.	
6	Coursework Result	The functionalities for submitting coursework and requesting extensions have been removed to simplify the page. Additionally, two tables have been added to display the grades for the courses in which the user is enrolled.	
7	Past Year Paper	A new page has been added to enhance the functionality of the student system. Users can download past papers for practice.	
8	Submit coursework	A new page has been added, designed to mimic the Ness system. Students can submit assignments by selecting the relevant course name.	
9	Extension	A new page has been added, modeled after the S3P system. After completing the form, students can click the send button to email it directly to their teacher's inbox.	
10	Student Absence	A new page has been added. After students complete the form, submitting the application will send an email to the teacher.	
11	Student Meeting with tutor	A new page has been added. After students complete the form, submitting the application will send an email to the teacher.	
12	Student Suspend	A new page has been added. After students complete the form, submitting the application will send an email to the teacher.	
13	Q&A	A new page has been added. Students can upload their questions about courses or assignments to this page.	

14	Staff Dashboard	The top left corner is linked to user information after login. This is primarily for users to view the 15 students assigned to them and to send meeting emails by entering student email addresses. The data in this table is added solely for testing functionality.	
15	Staff Timetable	Users can assign courses by entering an ID and selecting the course start time. They can also view existing timetables by clicking the 'Check Exist Timetable' button and make changes by clicking on the appropriate cells.	
16	Coursework	A new page has been added. Users can download student documents, grade them, and provide feedback	
17	Undertaken Courses	A new page has been added. Users can display and view the courses a student is enrolled in by entering the student's ID.	
18	Check Student Details	By entering a student's ID, the system displays the student's grades along with their relevant personal information.	
19	Module Management	A new page has been added. After selecting the assigned module, users can upload relevant teaching materials and course-related announcements.	

7.3 Use cases

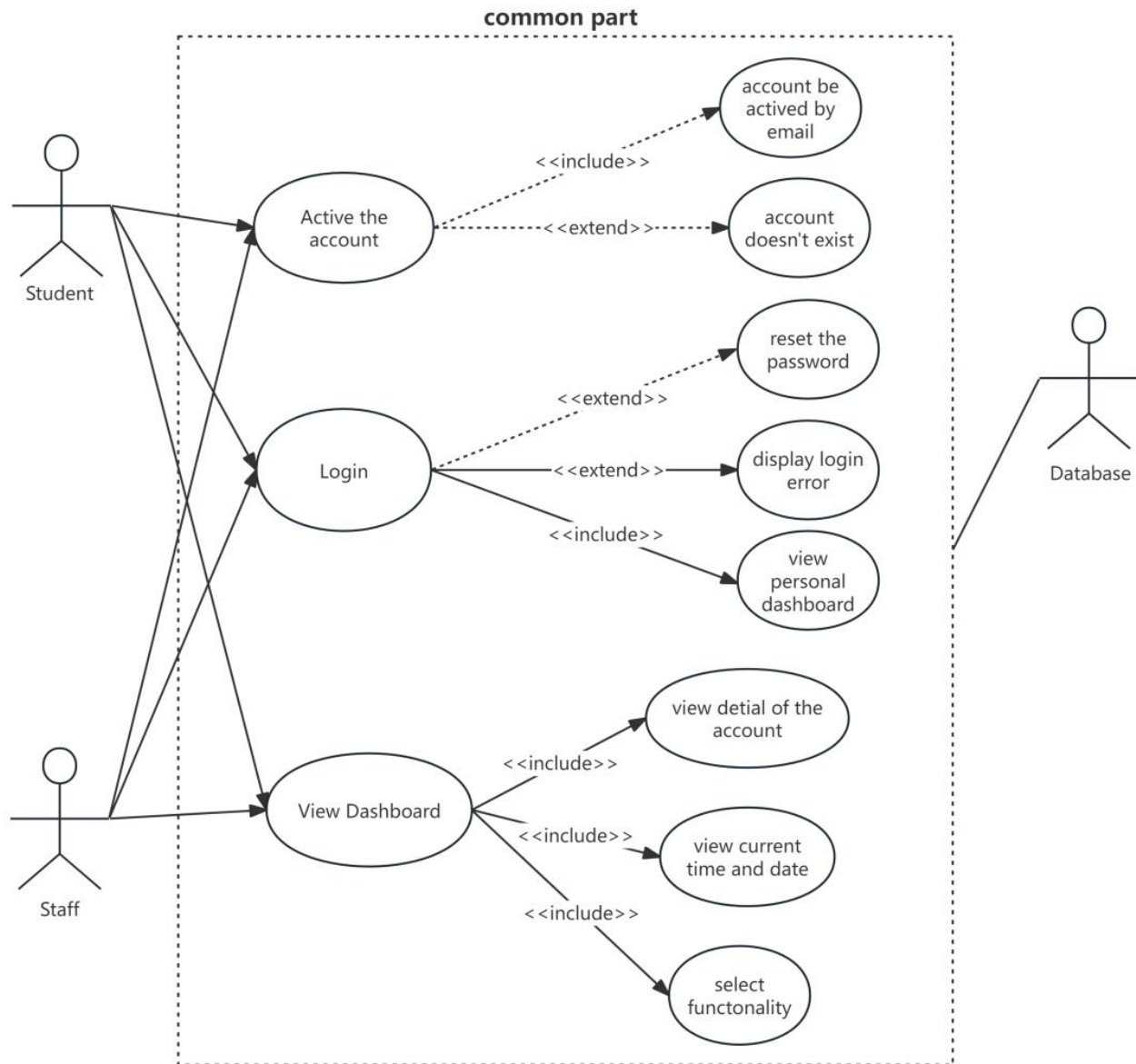


Figure 7.3.1 Common use cases.

I. Common action

(1) Activate User (Student/ Staff) account

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

<u>User action</u>	<u>System response</u>
(1) User clicks the “First Time Password Activation” button. (2) Enter Student/ Staff ID and email. (3) Click “Activate” button to continue.	<ul style="list-style-type: none"> - System send an email to notify the account register successful. - The default password is 123456.

(2) User logs into system

<u>User action</u>	<u>System response</u>
(1) User enters ID and password. (2) Click “Login” button.	<ul style="list-style-type: none"> - System checks the Use ID and the password. - If correct login to Main page.

(3) User reset the password

<u>User action</u>	<u>System response</u>
(1) User clicks “forget” button. (2) Enter ID /email and new password. (3) Click “update” button.	<ul style="list-style-type: none"> - System checks the ID and new password. - Send e-mail to e-mail address. - Update the password info in database.

(4) User view dashboard

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

<u>User action</u> - There are two ways to see the detail: (1) User can see functionality on the web page. (2) User clicks the side navigation bar to see the functionality	<u>System response</u> - System shows all the functionality on the page. - If user clicks the side navigation bar, the system displays all the functionality.
---	--

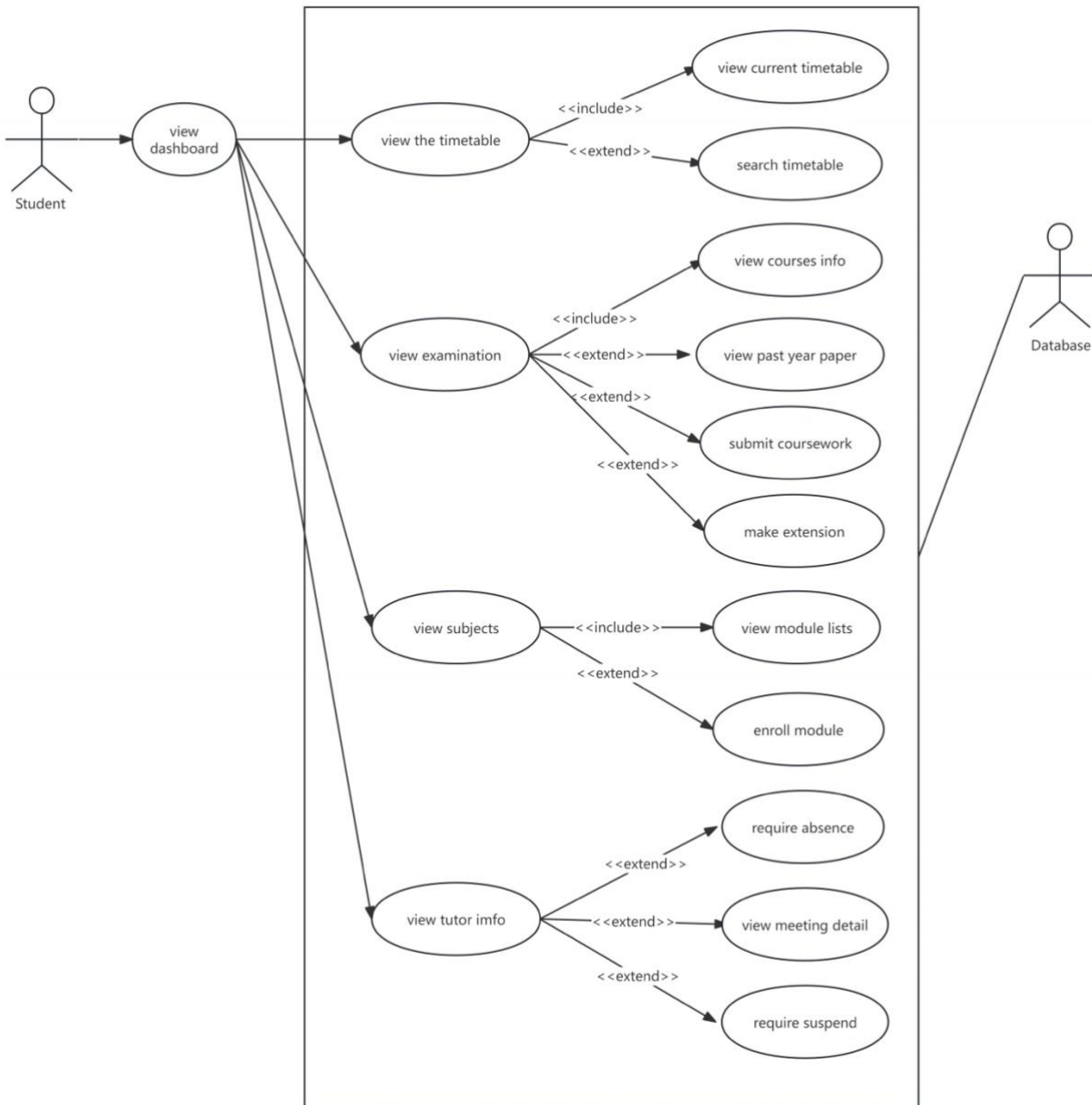


Figure 7.3.2 Student use cases.

II. Actor: Student

(1) Student check Timetable

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

<u>User action</u>	<u>System response</u>
<p>(1) Student clicks “timetable” button on the side navigation bar or click icon.</p> <p>(2) Student selects a date from the "Select A Date" field.</p> <p>(3) Student clicks the "Show Exist Timetable" button after selecting a date.</p>	<ul style="list-style-type: none"> - The system updates the timetable to show the schedule for the selected date. - Displays the existing timetable entries for the selected date in the timetable area.

(2) Student check Coursework result

<u>User action</u>	<u>System response</u>
<p>(1) Student clicks “Examination” button on the side navigation bar or click icon.</p> <p>(2) Student clicks “Coursework” button on the side navigation bar.</p> <p>(3) Student clicks “Feedback” button.</p>	<ul style="list-style-type: none"> - The system navigates to the “Examination” page. - The system displays the details of each coursework. - The system displays feedback on specific coursework from staff.

(3) Student finds exam paper of past years

<u>User action</u>	<u>System response</u>
<p>(1) Student clicks “past year paper” button on the side navigation bar on “Examination” page</p> <p>(2) Student clicks “Coursework” button on the side navigation bar or click icon</p> <p>(3) Student clicks “Feedback” button</p>	<ul style="list-style-type: none"> - The system navigates to the “past year paper” page. - The system displays paper of past years

(4) Student submit coursework

<u>User action</u>	<u>System response</u>
<p>(1) Student clicks “submission” button on the side navigation bar on “Examination” page</p> <p>(2) Student selects module button on the navigation bar</p> <p>(3) Student clicks “find file” button to select file in local device</p> <p>(4) Student clicks “submit” button to submit coursework</p>	<ul style="list-style-type: none"> - Navigates to the submission page - A file explorer window opens, allowing the student to browse and select the desired file to be submitted. - Uploads the selected file, and displays a confirmation message along with the submission details.

(5) Student request extension

<u>User action</u>	<u>System response</u>
<p>(1) Student clicks “Extension” button on the side navigation bar on “Extension” page</p> <p>(2) Student selects module button on the navigation bar</p> <p>(3) Student selects reasons for the extension</p> <p>(4) Student enters the duration of their personal extenuating circumstances in the "From" and "To" date fields</p> <p>(5) Student describes their personal extenuating circumstances in the "Description" text area.</p> <p>(6) Student submits the completed form by clicking the "Submit" button.</p>	<ul style="list-style-type: none"> - The system navigates to the extension page. The system displays the extension for. - The system records the selections for processing and review. - The system logs the duration for which the extension is requested. - The system stores the detailed description provided by the student for review by the administration. - The system processes the request, saves all provided information, and sends a confirmation message to the student. - An email notification is sent to the relevant academic department for further action.

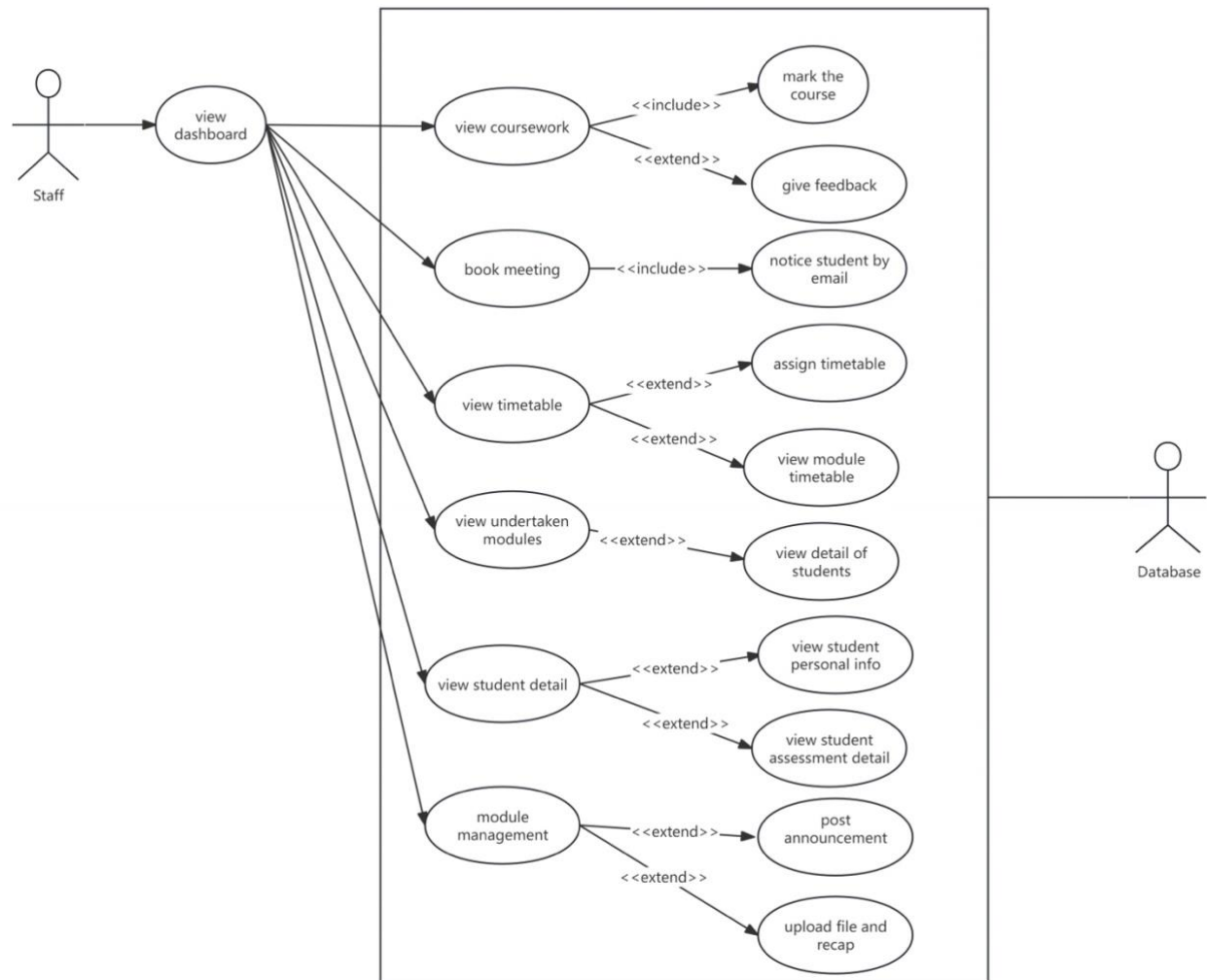


Figure 7.3.3 Staff use cases.

III. Actor: Staff

(1) Staff mark coursework and give feedback

<u>User action</u>	<u>System response</u>
(1) Staff click "coursework and feedback" button at the right in "Dashboard" page	- Navigates to the coursework management page, displaying a list of all available coursework items.
(2) Select coursework form coursework lists and click coursework	- Opens the selected coursework file and give studentID with module automatically.
(3) Type mark in the "Input Marks" box	- Send the input mark in the system for that specific student and coursework in their module.
(4) Write feedback info in feedback in text Area	

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

(5) Click “submit” button	<ul style="list-style-type: none"> - Stores the written feedback in association with the student's coursework record. - Saves all info and sends a notification to the student regarding the update on their coursework assessment.
---------------------------	---

(2) Staff create meeting

<u>User action</u>	<u>System response</u>
(1) Staff click “Book meeting with student” button at the right in “Dashboard” page (2) Enter student email (3) Select date of their meeting (4) Click “submit” button	<ul style="list-style-type: none"> - Opens the meeting booking interface - Validates the entered email address if it is registered in the student database. - Displays a calendar for the staff to choose a suitable date - Schedules the meeting, sends a confirmation email

(3) Build timetable

<u>User action</u>	<u>System response</u>
(1) Staff click on the chosen time and date on the timetable (2) Click the dropdown arrow on Classroom, Module and Lecturer, select options and then click “Submit” button	<ul style="list-style-type: none"> - After Clicking the dropdown arrow, the option will be displayed - When staff clicks submit button, the course will be added to the timetable

(4) Modify timetable

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

<u>User action</u>	<u>System response</u>
<p>(1) Staff click the course which you want to modify that has been added.</p> <p>(2) Click the delete button.</p>	<ul style="list-style-type: none"> - After clicking the course, the delete button shows on the list. - Press the delete button then the course will be removed.

(5) Check student information

<u>User action</u>	<u>System response</u>
<p>(1) Staff click "StuDetail" button at the right in "Dashboard" page.</p> <p>(2) Enter studentID.</p> <p>(3) Click "Check" button.</p>	<ul style="list-style-type: none"> - Navigates to the student search page. - Validates the entered email address if it is registered in the student database. - Displays the details of student.

(6) Module management

<u>User action</u>	<u>System response</u>
<p>(1) Staff click "Module management" button at the right in "Dashboard" page.</p> <p>(2) Selects a module from the "Select A Module" dropdown menu.</p> <p>(3) Clicks the "Upload Files" button under the "Teaching Material" section.</p> <p>(4) Uploads a file and confirms the upload.</p> <p>(5) Clicks the "Upload Recap" button under the "Recap" section.</p> <p>(6) Writes an announcement in the "Announcement" text area.</p>	<ul style="list-style-type: none"> - Navigates to the Module Management page. - Displays the selected module's specific options and current status. - Opens a file selection dialog where the faculty member can choose one or more files to upload as teaching materials. - Uploads the file to the server, associates it with the selected module, and displays a confirmation message indicating successful upload. - Similar to the teaching material upload.

[Team 13]	Specification Document	[09/05/2024]
-----------	-------------------------------	--------------

(7) Clicks the "post" button after typing the announcement.	<ul style="list-style-type: none"> - Enables text input in the designated area. - Posts the announcement to all students enrolled in the module, updates the course announcement section. - Sends an email or notification to the students.
---	--