**PRACTICES**

**Tutorial 1**

**(no questions in tutorial 1)**

**Tutorial 2**

**I. Final Project Planning**

**a)** Our objective why we chose to develop a Database System for the Master Course of ICT is because we would like to create a centralized database that will store, track, maintain, and retrieve information of ICT students, grades of students, units being offered, and lecturers. This database system will help the managing team’s work a lot easier as it will simplify their job.

**b)** Yes, this project is feasible because the likelihood of this project’s objective to be achieved in the following aspects are high:

## **Technical Feasibility:**

This project is technically feasible as the system will be developed using a database system which we know how to use and we don’t need to learn any new technical skills.

## **Economic Feasibility:**

This project is economically feasible since the development of the system requires a database management tool which is available for free. And, we have plenty of time to complete this project as we are a team of two people.

## **Organizational Feasibility:**

## Our system provides services which are developed to be flexible and can be easily expanded in the future. The proposed system will have high usage among the students and university staff.

## **Operational Feasibility:**

## The system will be developed using a free and open-source database and there are many free tools available to interact with the database. So very less economic and time resources will be consumed for the deploying and operating of the proposed system.

**c)** The variant of SDLC that we will use in this project is Agile Development as it will allow us to adjust priorities as the project progresses resulting in improved product quality delivery.

**d) Project plan**

1. **Project Scope**

**1.1 Scope Statement**

Western Sydney University wants us to develop a Database System for the Master Course of ICT. The database system will be released on October 19, 2020. The project will not cost anything as we are going to use MySQL and MySQL Workbench tool which is completely free.

**1.2 Objectives**

1. Create and develop a fully operational database system based on the features defined in section 1.3.
2. Make certain that the contents of the database should be very secure.
3. Release the database system on Week 14 (October 19, 2020).

**1.3 System Features**

In the scope of this project, the following features per user are included:

User: Features:

a. Managing Staff - Manage Students, Lecturers, and Units

- Manage Student Result

- Manage School Notification

- Define Unit Schedules and Units for Students and Lecturers

- Generate Fee Structure for University

- Generate Reports

b. Lecturers - Manage Students Assignment

- Assign Assignment

- Manage Student Grades

- Generate Remarks on Student Grades

- Generate Student Attendance

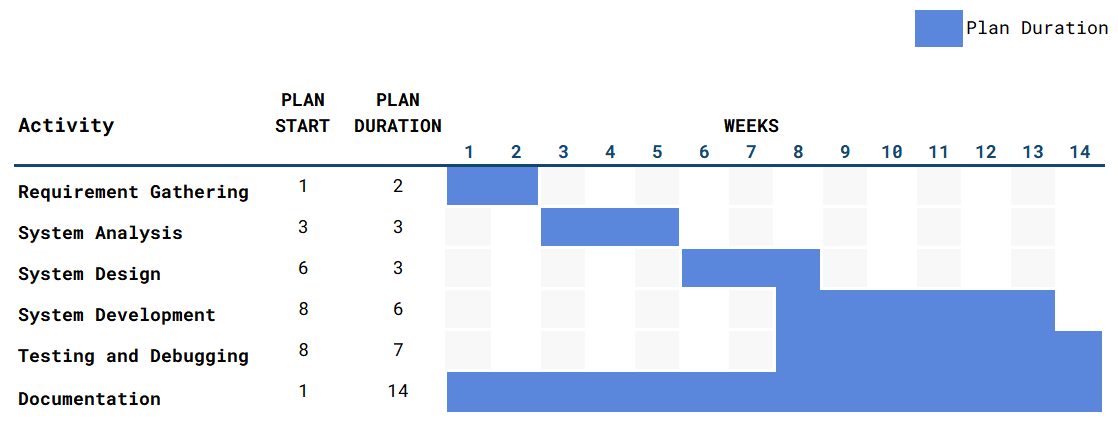
c. Students - Submit Assignments

- View Unit Schedules

- View Grades

- View Remarks

1. **Gantt Chart**

****

**3. Stakeholder Analysis**

The Project Team is identified as the internal stakeholders while the users (managing staff, teaching staff, and students) are the external stakeholders who contribute to this database system project.

**e)** Our group consists of two members Patricia Ann La Madrid and Shijan Kumar Shrestha who are enrolled in the unit Systems Analysis and Database Management at Western Sydney University.

**f)** **Communication Plan:**

- Project meetings are held in Zoom and the documentation is done in google docs.

**g) Risk Management Plan:**

It is critical to eliminate risks from the start to ensure that the database system project will be completed before the deadline. The following potential risks were identified by the team while developing the project:

## **Time Risk:**

Since the course requires us to finish the project within the spring semester, no extensions are allowed. This also means that any feature or improvements not implemented within the deadline may not go into the final product. Due to the time constraint, time must be treated as a resource and implementation of any feature or improvement needs to be calculated against the time available.

## **Resource Risk:**

This system doesn't require a huge resource for development. All the required resources are easily available, so there will be no resource risk.

## **Risk Management:**

## For the solution to the time risk, the team will work as the outline plan in a very strict manner. Also, as the process model being agile development, the database system will be developed in manageable parts.

**Tutorial 3**

**I. Requirement analysis.**

**a) Predefined Queries**

1. How many students are enrolled in a particular unit?

2. Who is the lecturer of a particular unit?

3. What is the average score of the students enrolled in a particular unit?

4. List all units offered on Monday.

5. List all students who get a D or HD grade in a particular unit.

6. List all students and units such that the students have failed the units.

7. List the lecturers who teach a particular student on Monday.

8. What are the units completed by a particular student?

9. What are the units failed by a particular student?

10. List all the lecturers who teach a particular unit.

**b) Interview**

Question by Patricia: What do you think about the business rules for our system?

Shijan’s Answer: 1. To use the system the user must be registered in the system.

2. Students cannot select the units they are already registered in.

Question by Shijan: What could be the business requirements for our system?

Patricia’s Answer: 1. Validating the users while they log into the system.

2. For every unit a student selects there should be validation that checks if the student is already registered in that particular unit.

**c) JAD Session**

We conducted a JAD session which included following steps:

1. Defining the objectives for JAD session:

i. Session objectives:

To find more business rules and business requirements for our system.

ii. Session deliverables:

New business rules and business requirements.

iii. Participant list:

Project members and the other stakeholders.

2. Preparing for the session.

i. We conducted a pre-session research.

ii. We created a session agenda.

iii. We arranged session logistics.

iv. We prepared the participants for the session.

3. Conducting the JAD session.

We started the meeting on time and all the participants were asked to generate new ideas to explore potential opportunities.

4. Producing the documents

We completed and organized the final document for easy use by project members.

**d) Questionnaire**

1. The development of this MICT system would be very helpful for the student, teachers and managing staff.
2. Strongly Agree
3. Agree
4. Uncertain
5. Disagree
6. Strongly Disagree
7. It would be helpful if you get an email about the various events and announcements in the system.
   1. Strongly Agree
   2. Agree
   3. Uncertain
   4. Disagree
   5. Strongly Disagree

**Tutorial 4**

**I.**

a) Student Use Case:

|  |  |
| --- | --- |
| **Title:** | Register for units |
| **Description:** | Actor wants to register for a unit provided for the current semester. |
| **Actor:** | Student |
| **Pre-Condition:** | Actor must log in to the system before this use case begins. |
| **Flow of events:** | 1. The System shows the list of units available for the current semester.  **Business rule: Students cannot select the units they are already registered in.**  2. The actor selects the units from the list of available offerings.  3. The system displays the unit schedule available for the selected units.  4. The actor selects the schedule from the list of available schedules. The actor can also change the schedules later up to for a fixed duration.  5. The units are saved in the system. |
| **Frequency of Use:** | This use case applies every time an actor wants to register for a unit. |
| **Post Condition:** | The student is registered in a unit and can view details and the schedule. |
| **Alternative Flow:** | If all the units are full, the actor cannot register for units and the use case ends. |

|  |  |
| --- | --- |
| **Title:** | Submit Assignment |
| **Description:** | Actor wants to submit an assignment for a unit. |
| **Actor:** | Student |
| **Pre-Condition:** | Actor must log in to the system before this use case begins. |
| **Flow of events:** | 1. The system shows the list of units that the actor is enrolled in.  2. The actor selects the units for which he/she wants to submit the assignment.  3. The system provides the list of options to the user.  4. The actor selects the option to submit the assignment.  5. The system displays the field to upload the files.  6. The actor uploads the files in the system.  7. The submitted assignments are saved in the system. |
| **Frequency of Use:** | This use case applies every time an actor wants to submit an assignment. |
| **Alternative Flow:** | If the assignment submission is closed, the actor cannot submit the assignment and the use case ends. |

b) Teaching Staff:

|  |  |
| --- | --- |
| **Title:** | Submit Grades |
| **Description:** | Actor wants to submit grades to the students |
| **Actor:** | Teaching Staff |
| **Preconditions:** | The actor must log in to the system. |
| **Flow of events:** | 1. The system displays the list of units taught by the actor.  2. The actor selects the unit.  3. The system displays the list of all the students enrolled in the unit.  4. The actor enters the grade for each student in the list. The actor can also change the grades later.  5. The grades are saved in the system. |
| **Frequency of Use:** | This use case applies every time an actor wants to submit grades. |
| **Alternative Flow:** | If the actor does not teach any units, the use case ends. |

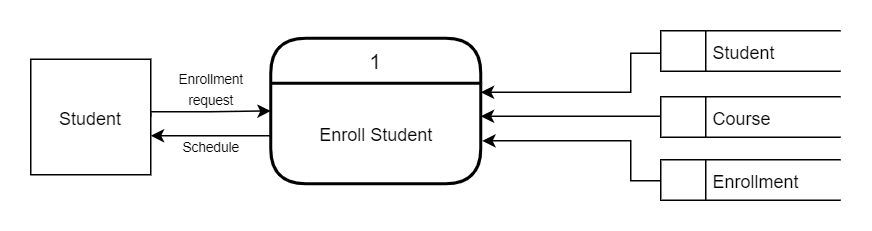
c) Managing Staff:

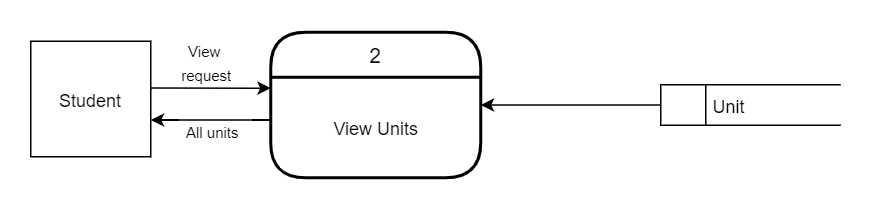
|  |  |
| --- | --- |
| **Title:** | Verify Student Fee Payment. |
| **Description:** | Actor wants to check if the student has paid the fee. |
| **Actor:** | Managing Staff |
| **Pre-Condition:** | Actor must log in to the system before this use case begins. |
| **Flow of events:** | 1. The system shows the list of units available for the current semester.  2. The actor selects the units from the list.  3. The system displays the list of students registered in the selected unit.  4. The actor checks all the students who have or have not paid the dues. |
| **Frequency of Use:** | This use case applies every time an actor wants to check student fee payment. |
| **Post-Condition:** | The actor gets the list of students who has not paid their dues and can notify the students to pay their fee. |
| **Alternative Flow:** | If no students are enrolled in the selected unit, the use case ends. |

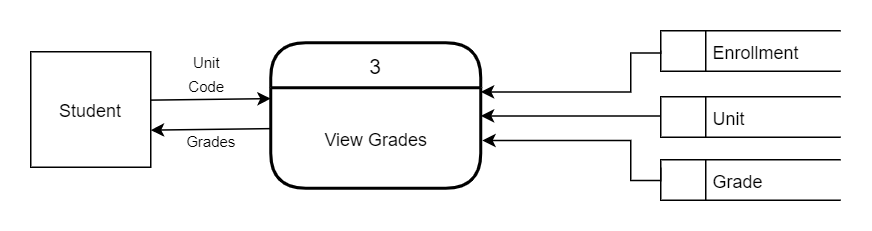
**Tutorial 5**

**I.**

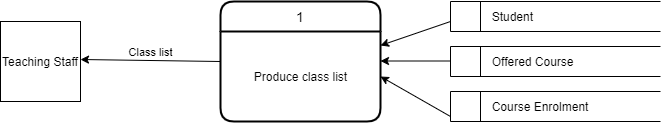
**a) student**

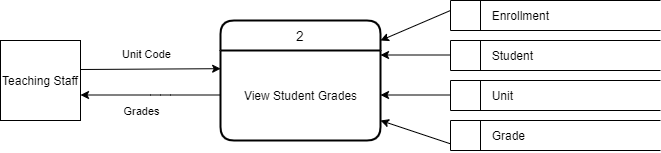


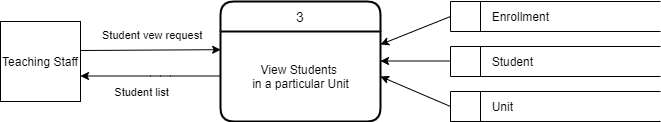




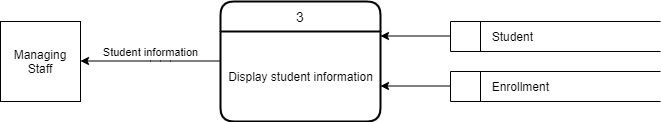
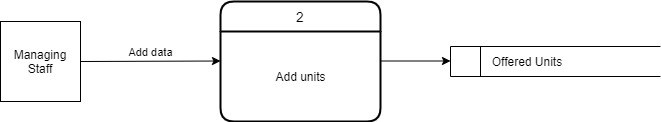
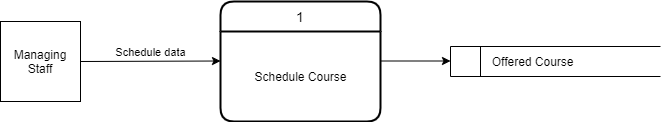
**b) teaching staff**







**c) managing staff**



**Tutorial 6**

**I.**

1. **Summary**

In order to develop a Database System for the Master Course of ICT, the team will create a centralized database that will store, track, maintain, and retrieve information of ICT students, grades, units being offered, and lecturers. The database system will be released in October 2020. This will help the managing team’s work as it will simplify their job. This project plan gives a brief statement of how the team aims to deliver the fully operational database system within the given time-frame.

This project plan includes the project scope that has been defined based on the objectives. Gantt Chart is used to show the project into six phases. There is a system analysis phase to identify the business rules and requirements. The system design phase shows the use cases of different users.

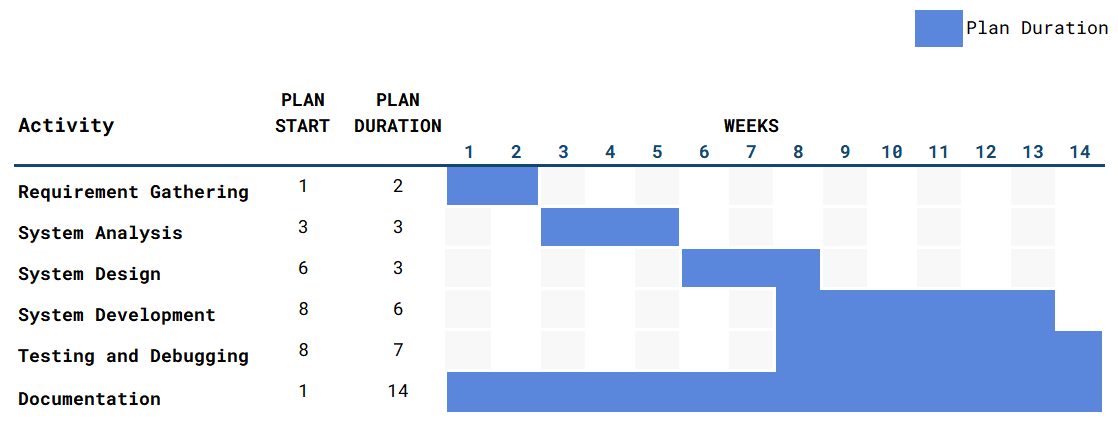
Due to time constraint, it is critical to address and eliminate foreseeable risks from the start to ensure that the database system project will be completed before the deadline.

1. **System request**

This project involves the creation and development of a database system that will provide required information to the system users about the MICT course. The users of this system are students, teaching staff and managing staff. The administrator controls the database and can alter the contents of the database, while the users can read the data and information provided by the database.

This document has been created to outline how the team plans to design and develop the database system for the Master Course of ICT. This report consists of the Work Plan, Risk Analysis, Feasibility Analysis, Use Cases and Data Flow Diagrams.

The reason why we decided to create a database system for the Master Course of Information and Communications Technology is because database systems are crucial for educational organizations as it stores all the important records about the school including staff and students. This will not only help the students and lecturers but will simplify the work of the managing team.

1. **Work plan **
2. **Feasibility analysis**

This project is feasible because the likelihood of this project’s objective to be achieved in the following aspects are high:

## Technical Feasibility:

This project is technically feasible as developing a Database system using MSSQL and mylittleadmin tool which is provided by the university.

## Economic Feasibility:

This project is economically feasible since the development of the system requires a database management tool which is provided to us for free. And, we have plenty of time to complete this project as we are a team of two people.

## Organizational Feasibility:

## The services our system provides are designed to be flexible and expandable in the near future. Our proposed system will have its high usage among the students and university staff.

* Operational Feasibility:

The system will be developed using a database system which is a free and open-source software and there are many free tools available to interact with the database. So the economic and time resource will not be consumed much for the deployment and operation of the proposed system.

1. **Requirements definition**

**Functional requirements**

* Since the project is fully dependent on the database, the contents of the database should be very secure.
* The user is capable of accessing the data and the admin only can alter the database.
* Authentication of the user must be done every time they try to login to the system.
* The database system should be capable of generating reports as desired by the university staff and students.

**Non-functional requirements**

* The system should maintain efficiency in terms of processing speed. The system should be able to response to the request of the user as soon as possible
* It should not be of large size and the content in the database should be easily accessible.
* Maintenance process must be simple and easy if any error or problem occurs in future.

**Business Rules**

1. To use the system the user must be registered in the system.

2. Students cannot select the units they are already registered in.

**Business Requirements**

1. Validating the users while they log into the system.

2. For every unit a student selects there should be validation that checks if the student is already registered in that particular unit.

1. a) Student:

|  |  |
| --- | --- |
| **Title:** | Register for units |
| **Description:** | Actor wants to register for a unit provided for the current semester. |
| **Actor:** | Student |
| **Pre-Condition:** | Actor must log in to the system before this use case begins. |
| **Flow of events:** | 1. The System shows the list of units available for the current semester.  **Business rule: Students cannot select the units they are already registered in.**  2. The actor selects the units from the list of available offerings.  3. The system displays the unit schedule available for the selected units.  4. The actor selects the schedule from the list of available schedules. The actor can also change the schedules later up to for a fixed duration.  5. The units are saved in the system. |
| **Frequency of Use:** | This use case applies every time an actor wants to register for a unit. |
| **Post Condition:** | The student is registered in a unit and can view details and the schedule. |
| **Alternative Flow:** | If all the units are full, the actor cannot register for units and the use case ends. |

|  |  |
| --- | --- |
| **Title:** | Submit Assignment |
| **Description:** | Actor wants to submit an assignment for a unit. |
| **Actor:** | Student |
| **Pre-Condition:** | Actor must log in to the system before this use case begins. |
| **Flow of events:** | 1. The system shows the list of units that the actor is enrolled in.  2. The actor selects the units for which he/she wants to submit the assignment.  3. The system provides the list of options to the user.  4. The actor selects the option to submit the assignment.  5. The system displays the field to upload the files.  6. The actor uploads the files in the system.  7. The submitted assignments are saved in the system. |
| **Frequency of Use:** | This use case applies every time an actor wants to submit an assignment. |
| **Alternative Flow:** | If the assignment submission is closed, the actor cannot submit the assignment and the use case ends. |

b) Teaching Staff:

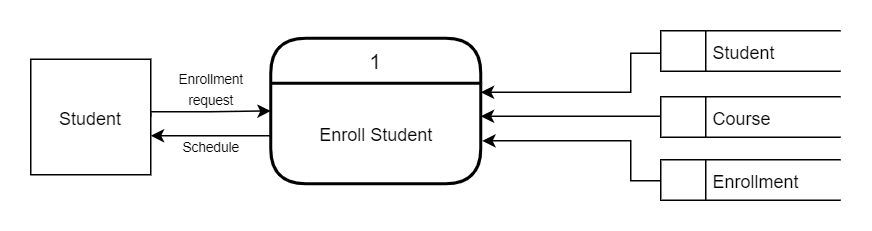
|  |  |
| --- | --- |
| **Title:** | Submit Grades |
| **Description:** | Actor wants to submit grades to the students |
| **Actor:** | Teaching Staff |
| **Preconditions:** | The actor must log in to the system. |
| **Flow of events:** | 1. The system displays the list of units taught by the actor.  2. The actor selects the unit.  3. The system displays the list of all the students enrolled in the unit.  4. The actor enters the grade for each student in the list. The actor can also change the grades later.  5. The grades are saved in the system. |
| **Frequency of Use:** | This use case applies every time an actor wants to submit grades. |
| **Alternative Flow:** | If the actor does not teach any units, the use case ends. |

c) Managing Staff:

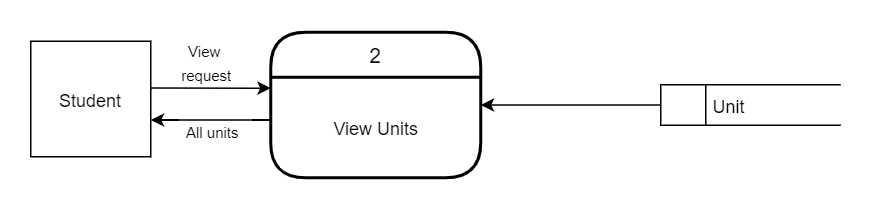
|  |  |
| --- | --- |
| **Title:** | Verify Student fee payment. |
| **Description:** | Actor wants to check if the student has paid the fee. |
| **Actor:** | Managing Staff |
| **Pre-Condition:** | Actor must log in to the system before this use case begins. |
| **Flow of events:** | 1. The system shows the list of units available for the current semester.  2. The actor selects the units from the list.  3. The system displays the list of students registered in the selected unit.  4. The actor checks all the students who have or have not paid the dues. |
| **Frequency of Use:** | This use case applies every time an actor wants to check student fee payment. |
| **Post-Condition:** | The actor gets the list of students who has not paid their dues and can notify the students to pay their fee. |
| **Alternative Flow:** | If no students are enrolled in the selected unit, the use case ends. |

**a. Student**

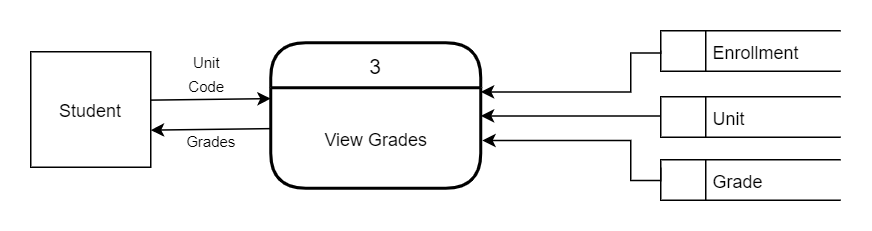
1. Enroll in a unit



2. View Unit details

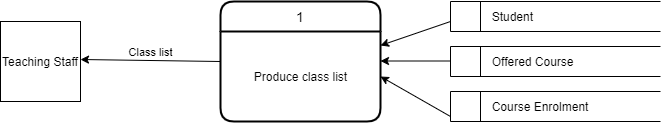


3. View grades

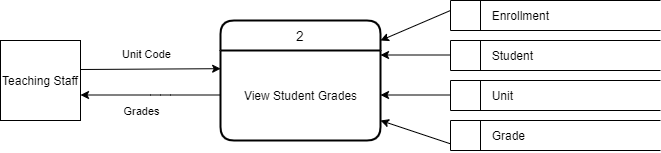


**b. teaching staff**

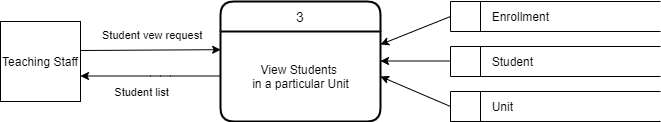
1. Produce class list



2. View student grades

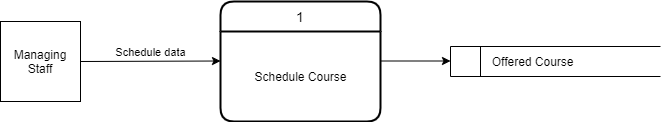


3. View students in a particular unit

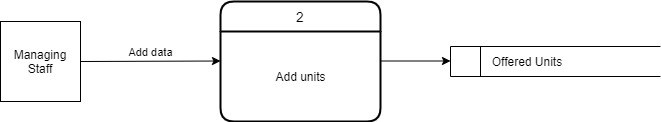


**c. managing staff**

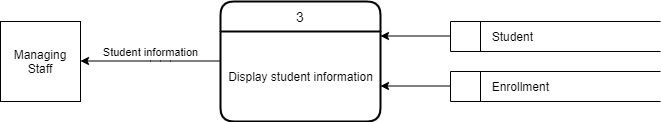
1. Schedule Course



2. Add units



3. Display student information



**Tutorial 7**

**SAD Entity Relationship Diagram**

Diagram, schematic

Description automatically generated