

# **Campus resource management system**

## **Phase 3**

Subject: System Analysis and Design

Code: SECD2613

Section 07

Prepared by

Abdalla Ali Abdalla Ali A23CS3022

EYAD AIMEN ELSHEIKH KHALIL A23CS3024

Othman Hassan Othman A23CS3026

Ali Isameldin Ali A23CS3001

## **1.0 Overview of the project:**

This project is designed to solve this university systems issues. The main focus of the project is to deliver IS that handles large amounts of data and automates most of the processes to reduce errors. This project covers four important areas to enhance its processes: Facility and booking management, Event management, student management and communication and notification.

## **2.0 Problem statement:**

The current system suffers from frequent input errors and delaying of processes and the system output is usually similar or close to its input due to lack of critical processes. The system also lacks some essential technical hardware equipment, while depending mainly on human workforce without much computerized intervention.

Not to mention that the system is difficult to use for new users.

## **3.0 Proposed solutions:**

To develop more advanced subsystems that can function together to maximize the output of the program, reduce the input errors, speed up the processes and can ease the management processes.

## **4.0 Current business Process and workflow:**

### **4.1 Facility booking and reservation management subsystem:**

In this subsystem we have 2 main entities student and booking management officer, so the processes are as follows:

1. Student: can book campus resources such as sport fields, libraries, restaurants, entertainment facilities, classrooms, etc..., they also can make all types of facility reservations as an example rooms reservations, suites reservations and facility reservations.
  
2. Booking management officer: can check the availability of facilities, confirm or deny reservations, issuing reservation slip containing information about the users, the reserved facility and the date, and can be stored in reservation data store.

The workflow:

The student goes to the booking management office. Then the student provides his/her information including name, ID. Then the student provides the reservation details, including date and what facility wants to book. Then the officer checks for the availability of the requested facility. Then the officer provides information about if the facility is available on that date or not to the student. And then the officer asks the student if they would like to confirm the reservation. If the student confirms the officer will issue a slip containing the information about the reservation and make another copy to store it and change the availability of the facility.

Functional requirements:

1. Input:
  - a. Student information.
  - b. Reservation details.
2. Process:
  - a. Checking for facility availability.
  - b. Confirming reservation.
  - c. Issuing slip.
3. Output:
  - a. Slip.

Non-functional requirements:

1. Performance:

The performance is not good, since the officer must search for facility availability manually and the officer might make some mistakes when issuing the slip.

2. Control:

If the facility is not available, the reservation cannot be confirmed and the student cannot make a reservation on the specified date.

## **4.2 Event management subsystem:**

In this subsystem we have 2 main entities student and event management officer, so the processes are as follows:

1. Student: can register for the events.
2. Booking management: can make schedules for upcoming events, manage events for the students to register and can confirm registration for students and prints a paper with the attendance list.

The workflow:

The student goes to the event management office. Then the student provides his/her information including name, ID. Then the student provides the event details that wishes to go to. Then the officer checks the event information. Then the officer registers the student for the event. The officer confirms the registration for and student and the student's information in the attendance list. In the same day as the event the officer prints a list of the students that can enter the event.

Functional requirements:

Input:

- b. Student information.
- c. Event details.

1. Process:

- a. Checking for facility availability.
- b. Confirming event registration.
- c. Writing student name in the attendance list.

2. Output:

- a. Attendance list.

Non-functional requirements:

1. Performance:

The performance is decent, since there are not many events and a mistake can take place only if the officer wrote student information wrong.

2. Control:

After checking for event information, if there is no event with the received information the officer cannot confirm the registration and the student cannot register for the event.

### **4.3 Student management subsystem:**

In this subsystem we have 2 main entities student and administrator, so the processes are as follows:

1. Student: can enroll for the university, register for courses, access schedule and academic profile.
2. Booking management officer: can complete Enrollment of new students, complete course registration for students, manage student records (edit, store, delete, show it to the student) and manage student activities.

The workflow:

The student goes to the administrator's office. Then the student provides his/her information including name, passport, faculty and etc... (for enrollment). The administrator takes the information to complete the enrollment. Then the administrator issues student card with the student ID for the student after the enrollment confirmation, issues a slip for the student and creates a student record for the student that also includes the student academic profile. The administrator provides information about available courses that the student can register for and the limit for credit hours. Then the student chooses the courses he/she would like to register for. Then the administrator registers the courses for the students and stores it in the student record. Finally, the administrator gives the student the schedule for the registered courses.

Functional requirements:

1. Input:
  - b. Student information.
  - c. Courses details.
2. Process:
  - a. Enrolling the new student.
  - b. Issuing student card.
  - c. Registering courses.
  - d. Storing student information and registered records.
  - e. Providing student with the schedule
3. Output:
  - a. Enrollment slip.
  - b. Student card.
  - c. Schedule.

Non-functional requirements:

1. Performance:

The performance is not good, due to the large number of new students that want to enroll. The current system is not good in terms of organizing and storing the student records, so the administrator must search it manually and modify it.

2. Control:

If the student information is not completed or the student did not pay for the enrollment the enrollment cannot be completed. If the student exceeds his/her credit hour limit the student

cannot complete the registration unless he/she reduces the number of courses, he/she would like to register for

#### **4.4 Communication and notification management subsystem:**

In this subsystem we have 3 main entities student, management, and other stakeholders.

The Workflow:

The current system includes Communication Between:

Management-Stakeholders:

Management takes the hard-copied data and summarizes it to write a report which is submitted to the stakeholders via email, and it includes brief info of the benefits, complaints and materials used in each time.

Management-Students:

Students communicate with the management to get specific info about current availability of courses and important dates in the future, they communicate via email or phone calls.

Management-Management:

Management members communicate within themselves in case of complications in the registration process of students or when finding difficulties with the stakeholders.

Also, the students can communicate with each other for group projects or research or just for activities.

There are also some cases where the data delivery should be urgent and done now. The management uses a hot dialing number to deliver such information.

Functional requirements:

1. Input:
  - a. Data for summarization.
  - b. Student information (contact information, results).

2. Process:
  - a. Summarizing data.
  - b. Sending result to students or other management departments.
3. Output:
  - a. Summarized data.
  - b. Student results information.

Non-functional requirements:

1. Performance:

The current performance of the sub-system is slow due to data redundancy and the data is separated between the email, messaging applications and the database.

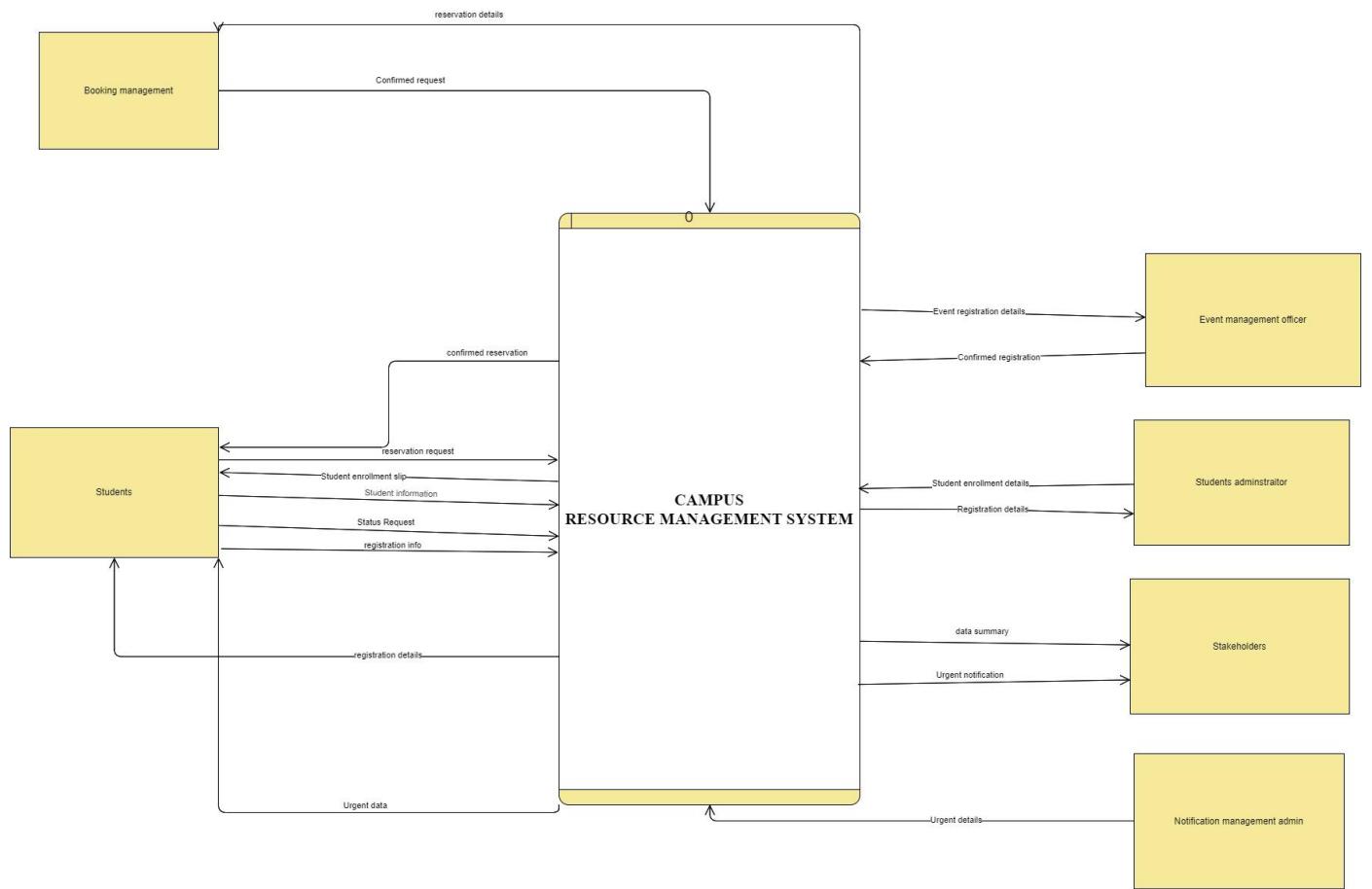
2. Control:

There are also some cases where the data delivery should be urgent and done now. The management uses a hot dialing number to deliver such information.

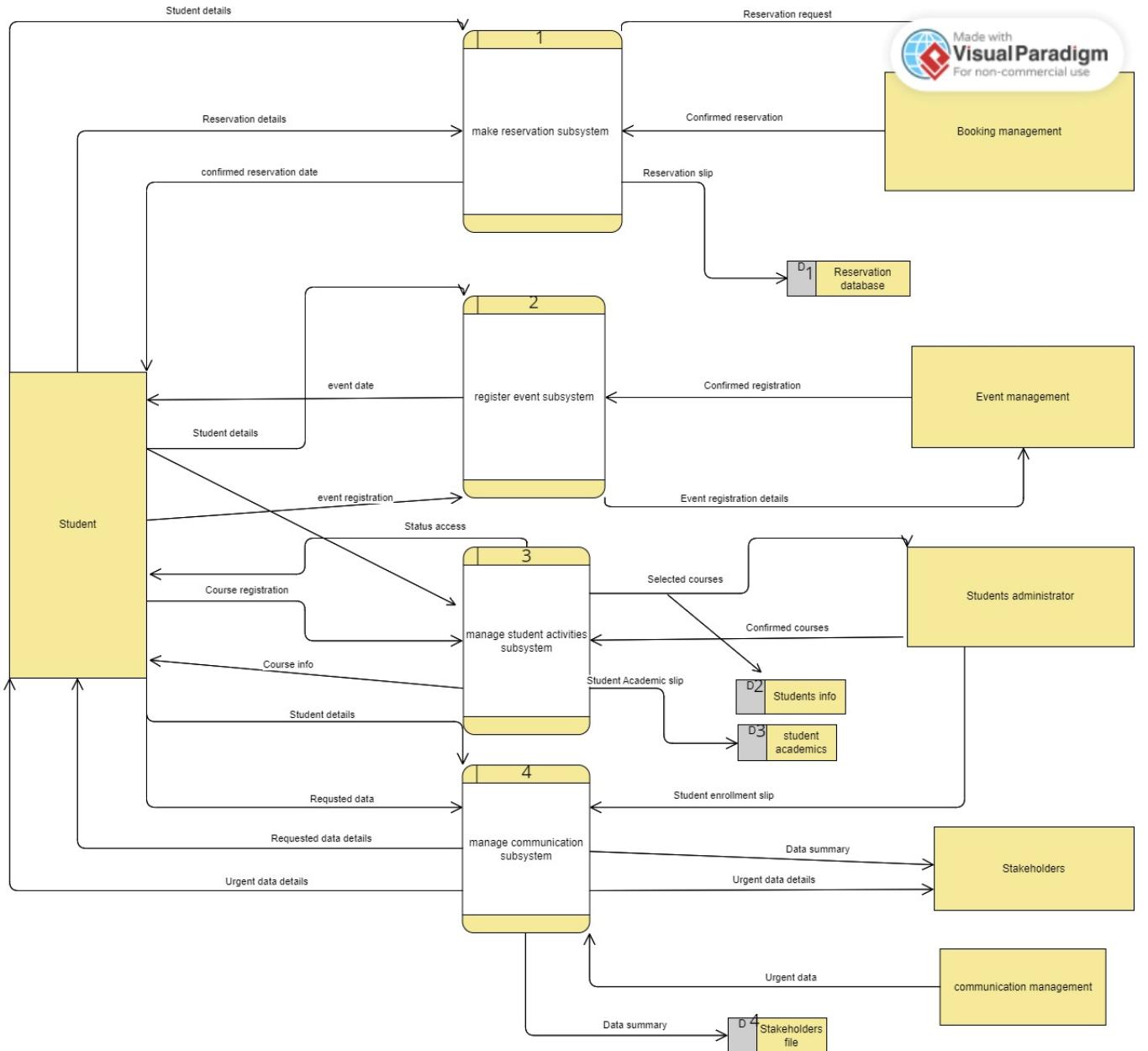
## **5.0 Logical DFD (AS-IS):**

We could not find the same shapes for entities and data stores stated in the slides, so we used these ones.

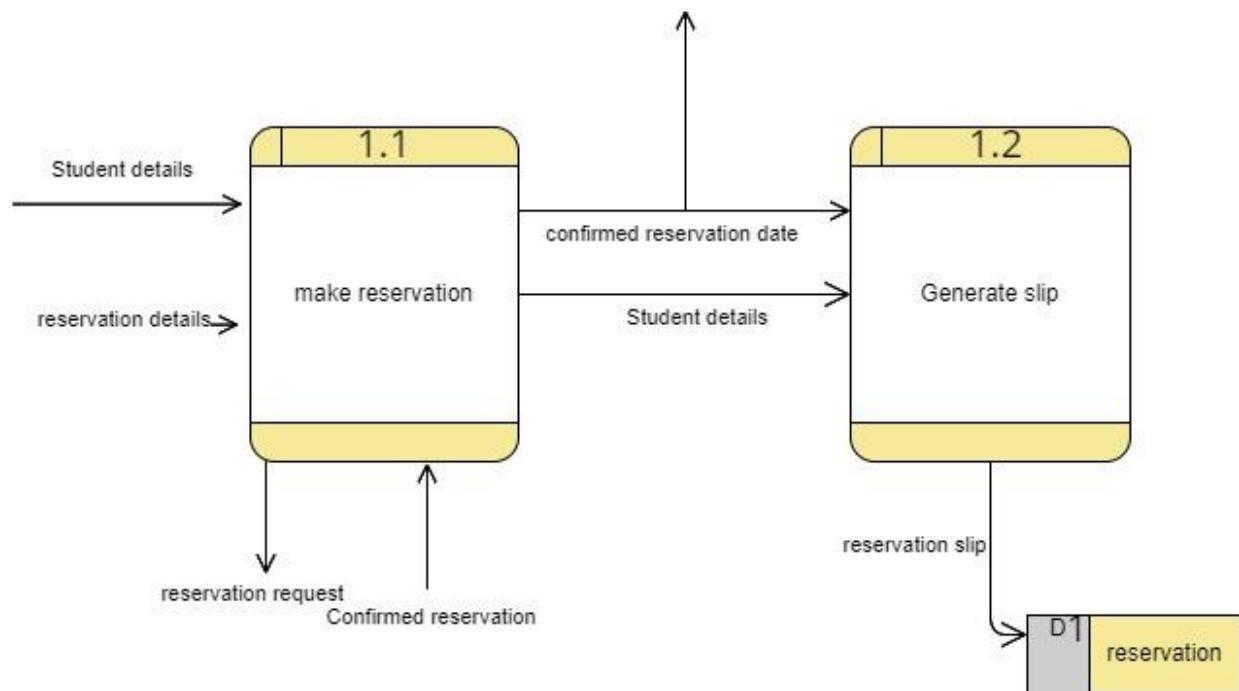
## 5.1 DFD CONTEXT DIAGRAM



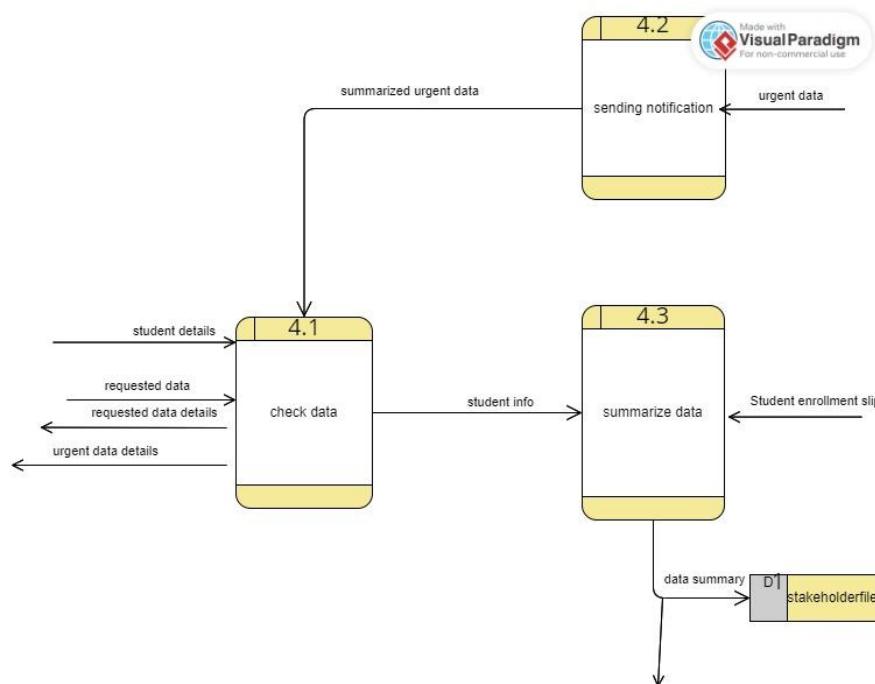
## 5.2 DFD ZERO DIAGRAM FOR THE SYSTEM



### 5.3 Facility booking and reservation management subsystem child diagram



### 5.4 Communication and notification management subsystem Child diagram



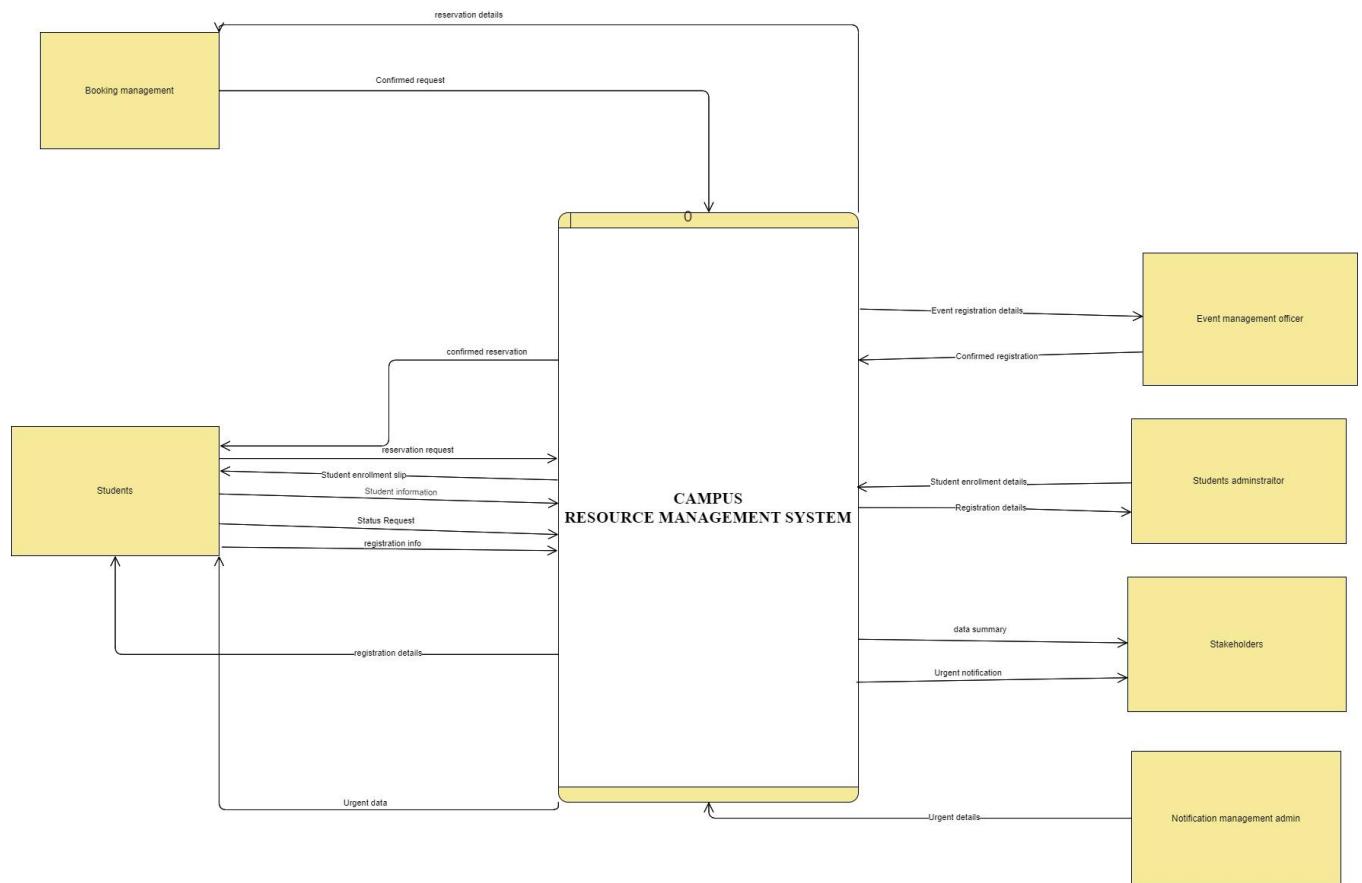
## 6.0 System Analysis and Specification:

We could not find the same shapes for entities and data stores stated in the slides, so we used these ones.

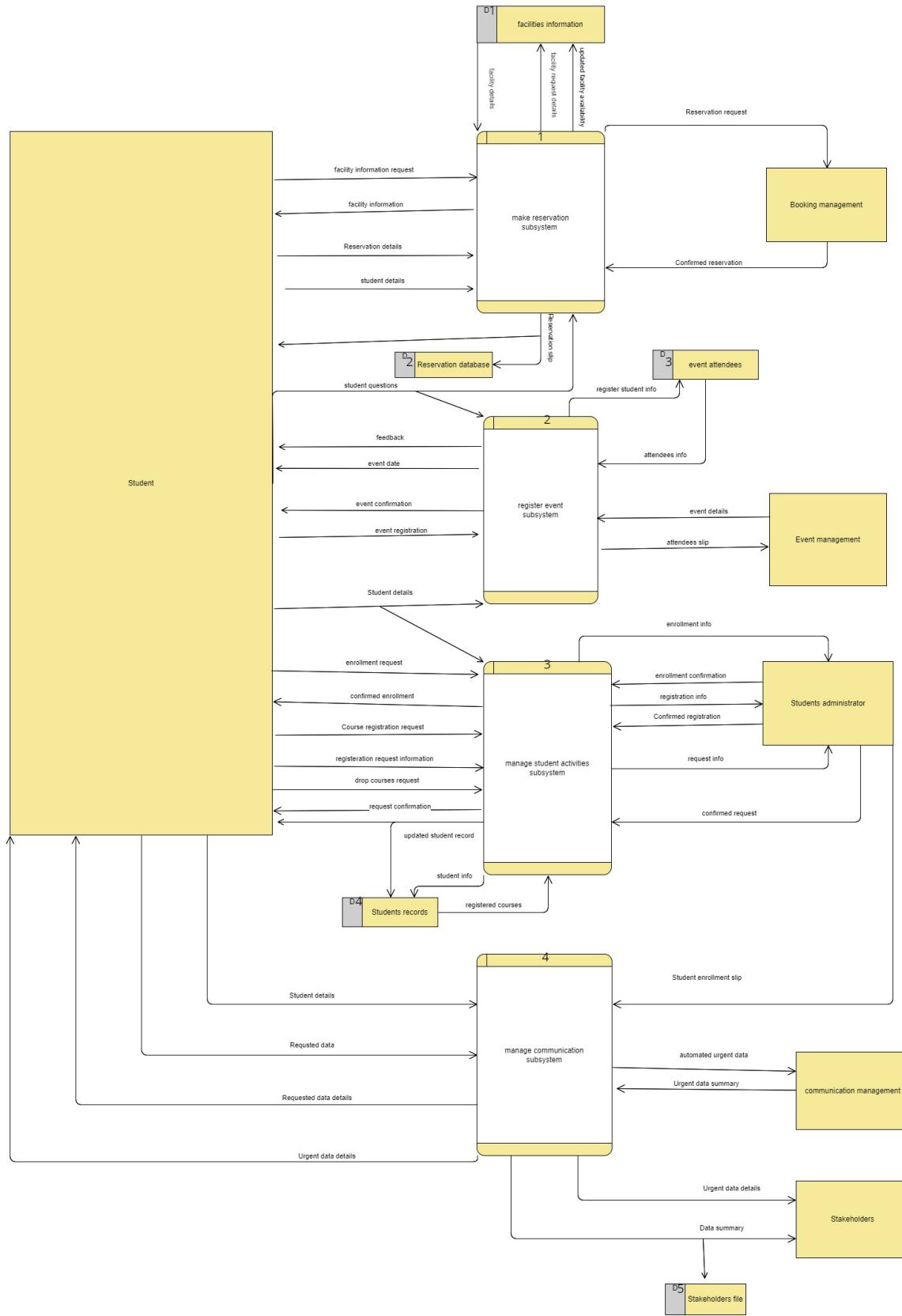
### 6.1 Logical DFD (TO-BE):

#### 6.1.1 DFD Context diagram

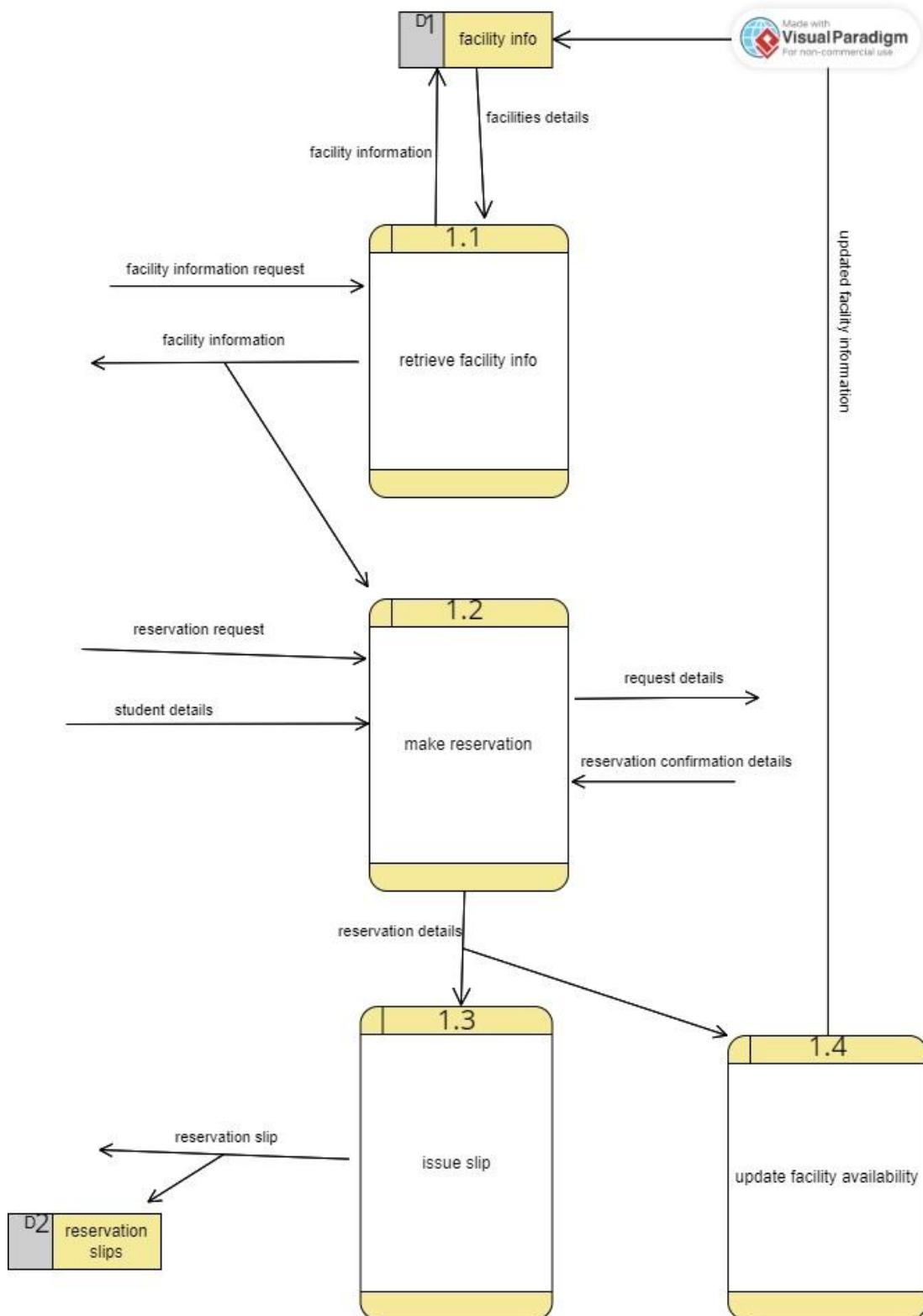
The new system has the same context diagram as the AS-IS system, since the new system will do the same processes but are more automated and efficient



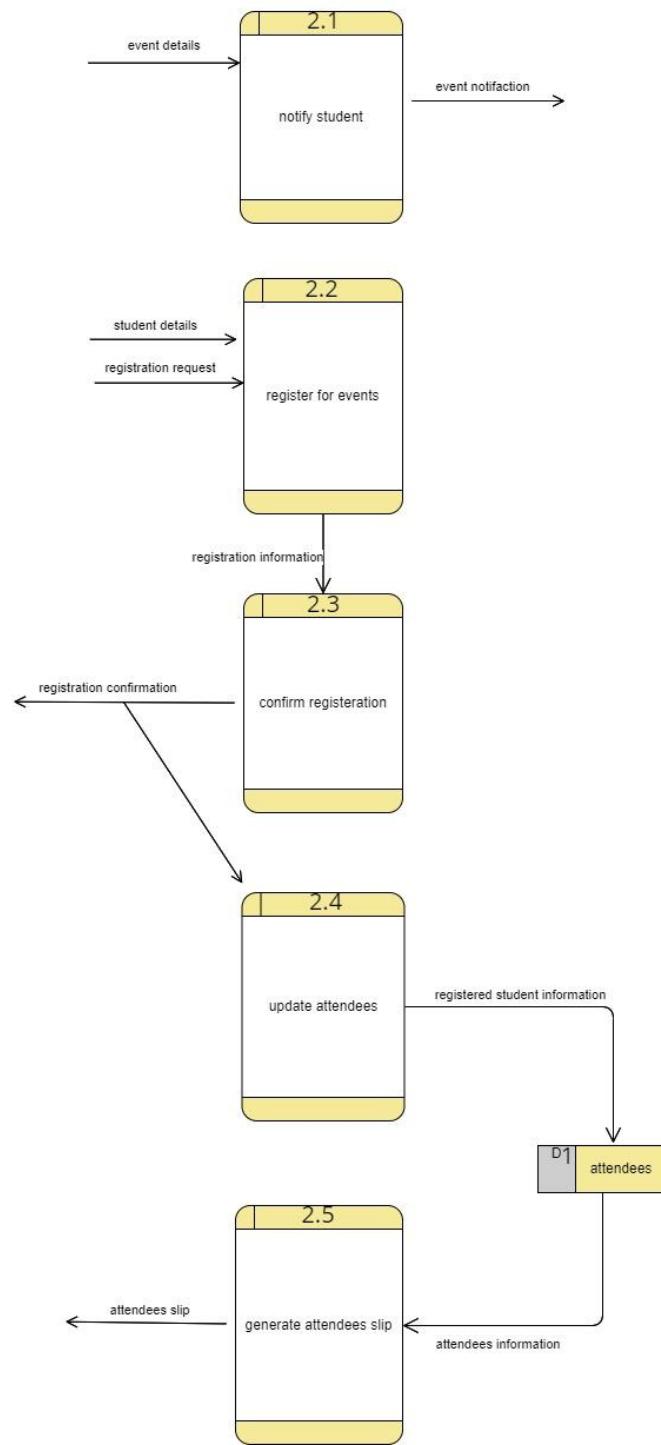
## 6.1.2 DFD Diagram 0



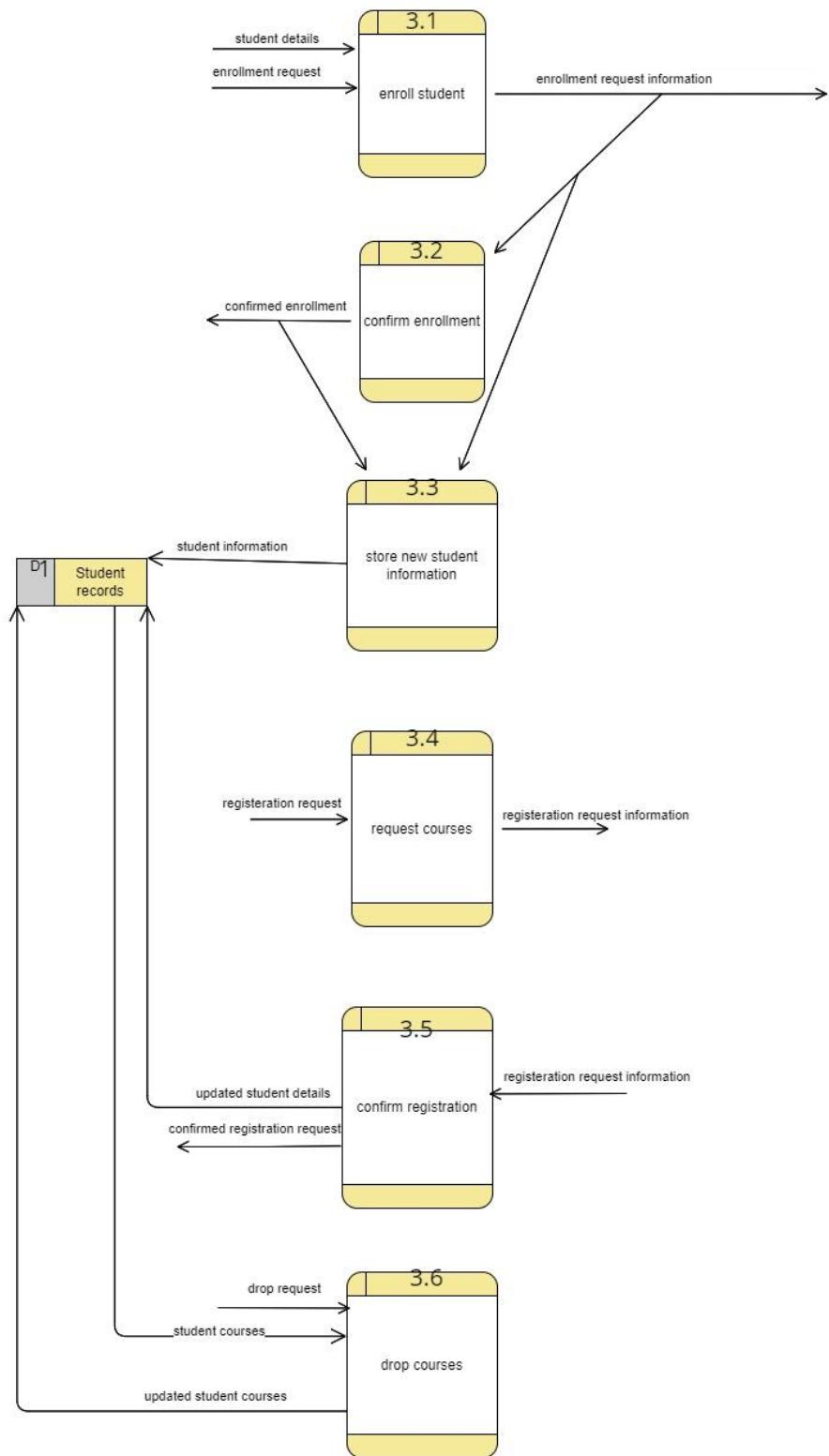
### 6.1.3 Make reservation subsystem child diagram



## 6.1.4 register events subsystem child diagram



### 6.1.5 Manage student activities subsystem child diagram



## 6.2 Process Specification:

We could not extract a process specification for each subsystem in diagram 0, because it is big and contains so many processes we just included their child diagrams.

### 6.2.1 Make reservation subsystem process specifications:

Number: 1

Name: [make reservation subsystem](#)

Description: The student can view information about the facilities and book facilities.

---

Input Data Flow:

Student details

Facility information request

Reservation details

Facility details

[Confirmed reservation](#)

---

Output Data Flow:

Facility details request

Facility information

Reservation request

updated facility availability

reservation slip

---

Type of Process:

Subprogram/Function Name:

Online

Manual

Batch

---

Process Logic:

Conditions:	1	2	2
Student info Valid	Y	Y	--
Facility is available for booking	Y	--	--

Send facility information			X
Receive reservation request		X	
Forward reservation details	X		
Send reservation slip	X		

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

If the student wants to make a reservation for an unavailable facility he has to wait, but he'll not be informed of the time of when the facility is available.

Number: 1.1

Name: **retrieve facility info.**

Description: retrieves important information from the facility

---

Input Data Flow:

**Facility information request**

---

Output Data Flow:

**Facility information**

---

Type of Process:	Subprogram/Function Name:
<input checked="" type="checkbox"/> Online <input type="checkbox"/> Manual <input type="checkbox"/> Batch	

Process Logic:

BEGIN

RECEIVE Facility information request.

RETREIVE Facility Details from facility info database.

SEND Facility information file TO user.

STORE Facility information in facility info database.

END

---

Refer to Name:

<input checked="" type="checkbox"/> Structured English	<input type="checkbox"/> Decision Table	<input type="checkbox"/> Decision Tree
--	---	--

Unresolved Issues:

There is a case where the required data cannot be found.

Number: 1.2

Name: **make reservation.**

Description: **does the required reservation.**

---

Input Data Flow:

Reservation Request.

Student details.

---

**Reservation confirmation details.**

Output Data Flow:

Request details

---

Reservation details

---

Type of Process:

---

Subprogram/Function Name:

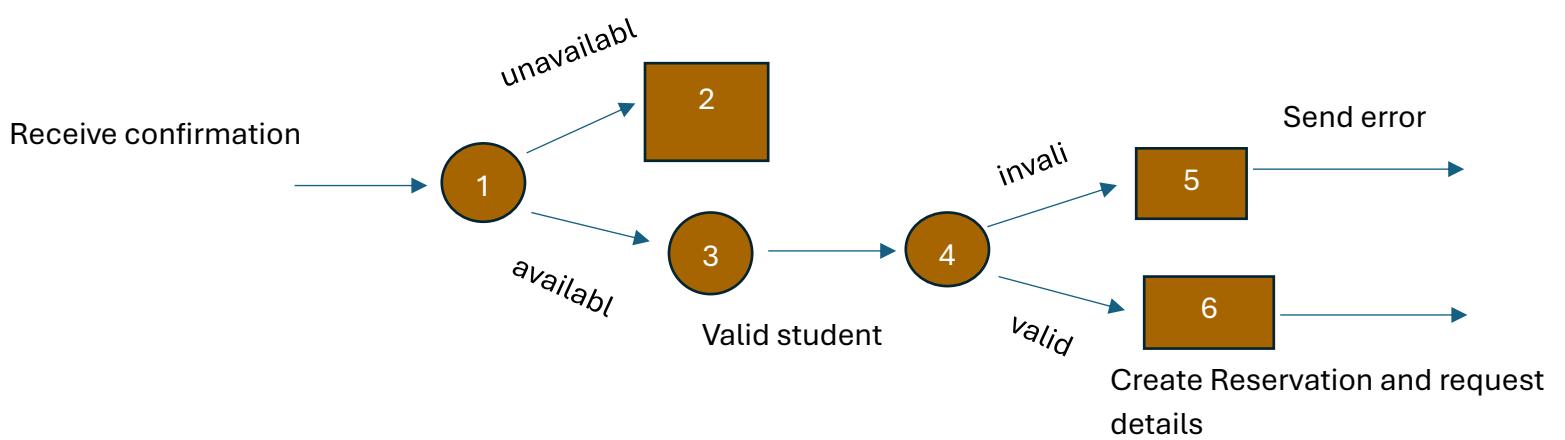
Online

Manual

Batch

---

Process Logic:



Refer to Name:

---

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

There is a case where the reservation details or the Student details Are invalid.

Number: 1.3

Name: **issue slip.**

Description: generates the slip for reservation.

---

Input Data Flow:

**Reservation details.**

---

Output Data Flow:

**Reservation Slip.**

---

Type of Process:	Subprogram/Function Name:
------------------	---------------------------

Online

Manual

Batch

Process Logic:

BEGIN

RECEIVE Reservation details.

Generate Reservation slip.

SEND Reservation slip To user.

STORE Reservation slip in reservation slip.

END

---

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

There is a case where the required data cannot be found.

Number: 1.4

Name: [update facility availability.](#)

Description: [updates the availability of the facility.](#)

---

Input Data Flow:

[Reservation details.](#)

---

Output Data Flow:

[Updated facility information.](#)

---

Type of Process:

Subprogram/Function Name:

Online

Manual

Batch

---

Process Logic:

BEGIN

RECEIVE Reservation details.

Generate Updated facility information.

STORE Updated facility information in facility info database.

END

---

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

There is a case where the required data cannot be stored.

## 6.2.2 Register event subsystem process specifications:

Number: 1

Name: make reservation subsystem

Description: the student can view information about the facilities and book facilities.

---

Input Data Flow:

Student details

Facility information request

Reservation details

Facility details

Confirmed reservation

---

Output Data Flow:

Facility details request

Facility information

Reservation request

updated facility availability

reservation slip

---

Type of Process:

Subprogram/Function Name:

Online

Manual

Batch

---

Process Logic:

Conditions:	1	2	2
Student info Valid	Y	Y	--
Facility is available for booking	Y	--	--
Send facility information			X
Receive reservation request			X

Forward reservation details	X	
Send reservation slip	X	

Refer to Name:

Structured English  
Tree

Decision Table

Decision

Unresolved Issues:

If the student wants to make a reservation for an unavailable facility he has to wait, but he'll not be informed of the time of when the facility is available.

Number: 2

Name: Register Event subsystem

Description: student can be notified for events, and register for events

Input Data Flow:

Event details

Event registration

Attendees' info

student details

Output Data Flow:

Event date

Event confirmation

Attendees slip

Register student info

Type of Process:

Online

Manual

Batch

Subprogram/Function Name:

Process Logic:

BEGIN

READ event details

SEND event notification to student  
READ student details  
READ registration request from student  
CONFIRM student registration  
UPDATE the number of attendees in attendee's database  
CREATE attendees slip  
SEND attendees slip to Event Management  
END

---

Refer to Name:

Structured English       Decision Table       Decision Tree

---

Unresolved Issues:

If there is no available seat in the event the system can still take registration request, The number of requests while the event is closed might crash the system.

Number: 2.1

Name: [notify student](#).

Description: notifies the student of important information.

---

Input Data Flow:

[Event details](#)

---

Output Data Flow:

[Event notification](#).

---

Type of Process:	Subprogram/Function Name:	
<input checked="" type="checkbox"/> Online <input type="checkbox"/> Manual <input type="checkbox"/> Batch		

---

Process Logic:

BEGIN

ACCEPT Event details.

Send Event notification to student.

END

---

Refer to Name:

<input checked="" type="checkbox"/> Structured English	<input type="checkbox"/> Decision Table	<input type="checkbox"/> Decision Tree
--	---	--

---

Unresolved Issues:

There is a case where the student contact info cannot be found.

Number: 2.2

Name: [register for events.](#)

Description: [does the required reservation.](#)

---

Input Data Flow:

Student details.

---

Registration request.

Output Data Flow:

---

[Registration information](#)

---

Type of Process:	Subprogram/Function Name:
------------------	---------------------------

Online       Manual       Batch

Process Logic:

BEGIN

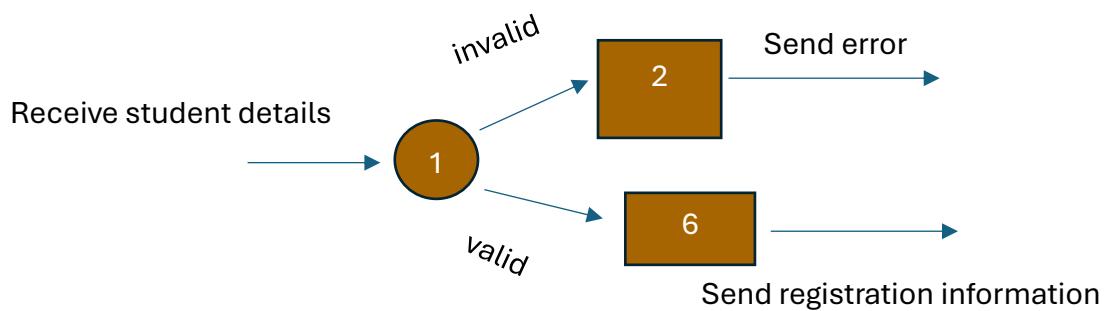
ACCEPT student details.

CREATE registration information.

Send registration information to student.

END

Refer to Name:



---

Structured English       Decision Table       Decision Tree

---

---

Unresolved Issues:

There is a case where the reservation details or the Student details Are invalid.

Number: 2.3

Name: [confirm registration.](#)

Description: [confirms registration for the student.](#)

---

Input Data Flow:

[Registration information.](#)

---

Output Data Flow:

[Registration confirmation.](#)

---

Type of Process:

Online

Manual

Batch

Subprogram/Function Name:

Process Logic:

BEGIN

RECEIVE Registration information.

CONFIRM Registration.

SEND Registration confirmation To student.

END

---

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

There is a case where the required info is invalid.

Number: 2.4

Name: **update attendees.**

Description: **updates the list of attendees.**

---

Input Data Flow:

**Registration confirmation**

---

Output Data Flow:

**registered student information**

---

Type of Process:	Subprogram/Function Name:
------------------	---------------------------

Online

Manual

Batch

Process Logic:

BEGIN

RECEIVE Registration confirmation.

CREATE registered student information.

STORE registered student information in attendees database.

END

---

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

There is a case where the selected student is missing from the attendees list.

Number: 2.5

Name: generate attendees slip.

Description: generates the slip for attendees.

---

Input Data Flow:

Attendees information.

---

Output Data Flow:

Attendees slip.

---

Type of Process:	Subprogram/Function Name:
------------------	---------------------------

Online

Manual

Batch

Process Logic:

BEGIN

RECEIVE attendees information.

GENERATE attendees slip.

SEND attendees slip to EVENT MANAGEMENT.

END

---

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

There is a case where the management contact info is missing.

### **6.2.3 Manage student activities subsystem process specifications:**

Number: 3.0

Name: Manage student activities subsystem

Description: students can enroll, can register for courses and can drop courses, and administration can manage their curriculum activities.

---

Input Data Flow:

student details

enrollment request

enrollment confirmation

course registration request

course registration confirmation

course drop request

confirmed drop course request

registered courses

---

Output Data Flow:

student info

enrollment information

enrollment confirmation

request info

registration info

registration confirmation

course schedule

drop course request confirmation

updated student record

---

Type of Process:

Online

Manual

Batch

Subprogram/Function Name:

Process Logic:

BEGIN

READ student details

READ enrollment request

SEND enrollment information to student administrator.

READ enrollment confirmation

SEND enrollment confirmation to student

READ course registration request

SEND course registration info to student administration

READ course registration confirmation

SEND registration confirmation and course schedule

READ course drop request

CONFIRM drop course request

SEND drop course confirmation to student

END

---

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

The enrollment request confirmation processes are manual, the time to process enrollment for many students is relatively a lot, the student will think he had been declined and enroll in another faculty.

Number: 3.1

Name: Enroll student

Description: Receive an enrollment request and send it to Administrator

---

Input Data Flow:

Student details.

Enrollment request.

---

Output Data Flow:

Enrollment request information.

---

Type of Process:	Subprogram/Function Name:
<input checked="" type="checkbox"/> Online <input type="checkbox"/> Manual <input type="checkbox"/> Batch	

---

Process Logic:

BEGIN

READ student details

READ enrollment request

SEND enrollment information to student administrator.

END

---

Refer to Name:

<input type="checkbox"/> Structured English	<input type="checkbox"/> Decision Table	<input type="checkbox"/> Decision Tree
---	---	--

---

Unresolved Issues:

If the university receives a lot of requests at the same time, what will Happen?

Number: 3.2

Name: Confirm enrollment

Description: The student administrator confirms the enrollment request from the student

---

Input Data Flow:

Enrollment request information.

---

Output Data Flow:

Confirmed enrollment.

---

Type of Process:

Online

Manual

Batch

Subprogram/Function Name:

Process Logic:

BEGIN

READ enrollment request information

BEGIN IF

IF student request is confirmed

    SEND the confirmed enrollment request to Store new student information process

    NOTIFY student that he/she is accepted

ELSE

    NOTIFY student that he/she is not accepted

END IF

END

---

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

After the student is confirmed and notified, what will happen to the student record if the student changed his/her mind and doesn't want to continue with the university.

Number: 3.3

Name: **Store new student information**

Description: This process stores student information (from enrollment request information because it has the student information in it, and the student information includes faculty, fees and other related data) in student records data store

---

Input Data Flow:

**Confirmed enrollment.**

**Enrollment request information**

---

Output Data Flow:

**Student information.**

---

Type of Process:	Subprogram/Function Name:
<input checked="" type="checkbox"/> Online <input type="checkbox"/> Manual <input type="checkbox"/> Batch	

Process Logic:

BEGIN

READ confirmed enrollment

READ enrollment request information

EXTRACT Student information from enrollment request information

MOVE student information to Student records data store

END

---

Refer to Name:

Structured English       Decision Table       Decision Tree

---

Unresolved Issues:

How will the system check if the enrollment confirmation and enrollment request information belong to the same student?

Number: 3.4

Name: Register for courses

Description: Student sends his/her courses to register for and it will be sent to the student administrator

---

Input Data Flow:

Registration request.

---

Output Data Flow:

registration request information

---

Type of Process:

Subprogram/Function Name:

Online

Manual

Batch

---

Process Logic:

BEGIN

READ registration request

SEND registration request information to student administrator

END

---

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

How will the system manage the situation that the student sends a registration request multiple times?

Number: 3.5

Name: Confirm registration

Description: The student administrator confirms or rejects the registration

---

Input Data Flow:

Registration request information

---

Output Data Flow:

Updated student details

Confirmed registration request

---

Type of Process:

Subprogram/Function Name:

Online

Manual

Batch

---

Process Logic:

BEGIN

READ Registration request information

BEGIN IF

IF Registration request information is confirmed

    SEND the updated student details to student records data store

    NOTIFY student that his/her registration request is accepted

ELSE

    NOTIFY student that his/her registration request is not accepted

END IF

END

---

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

How will the student administrator retrieve the student record to check the registration request information with already registered courses.

Number: 3.6

Name: Drop courses

Description: Student sends his/her courses to drop and be removed from the registered courses

---

Input Data Flow:

Drop request

Student courses

---

Output Data Flow:

updated student courses

---

Type of Process:	Subprogram/Function Name:
------------------	---------------------------

Online       Manual       Batch

Process Logic:

BEGIN

READ drop request

RETRIEVE student courses from student records data store

BEGIN IF

IF student courses are greater than 2

    REMOVE the course specified from drop request from student courses

    SEND updated student courses to student records data store.

END IF

END

---

Refer to Name:

Structured English       Decision Table       Decision Tree

---

Unresolved Issues:

How will the system notify the student if the drop request is confirmed or declined?

### **6.2.3 Manage communication subsystem process specifications:**

Number: **4.0**

Name: **manage communication subsystem**

Description: **send the requested data to students upon requesting, summarize the overall data and send it to**

**Stakeholders, notify the entities if there is an emergency.**

---

**Input Data Flow:**

**Student details**

**Student enrollment slip**

**Requested data**

**Urgent data summary**

---

**Output Data Flow:**

**Requested data details**

**Data summary**

**Automated urgent data**

**Urgent data details**

---

Type of Process:

Subprogram/Function Name:

Online

Manual

Batch

---

Process Logic:

BEGIN

READ data request

READ student details

BEGIN IF

IF student details are valid

SEND the requested data to student

END IF

SEND automated urgent data to communication management

READ the urgent data summary

SEND urgent data details to student and stakeholders

SEND the data summary to stakeholders and database file

END

---

Refer to Name:

Structured English

Decision Table

Decision Tree

---

Unresolved Issues:

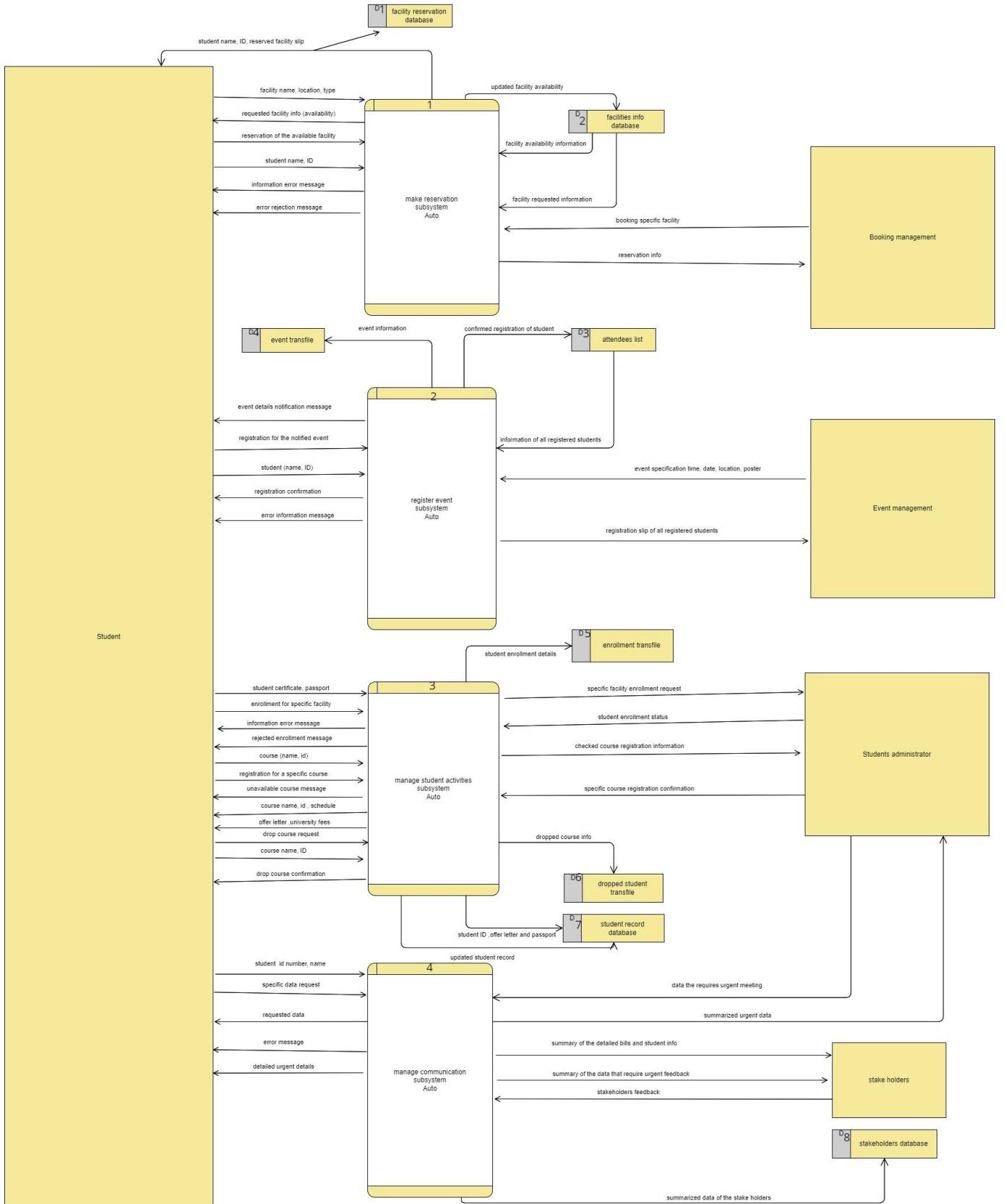
Receiving many data requests, while sending multiple emergency notifications, these two might cause a clash in the system, and causing the system to send the wrong information.

## **7.0 physical system design:**

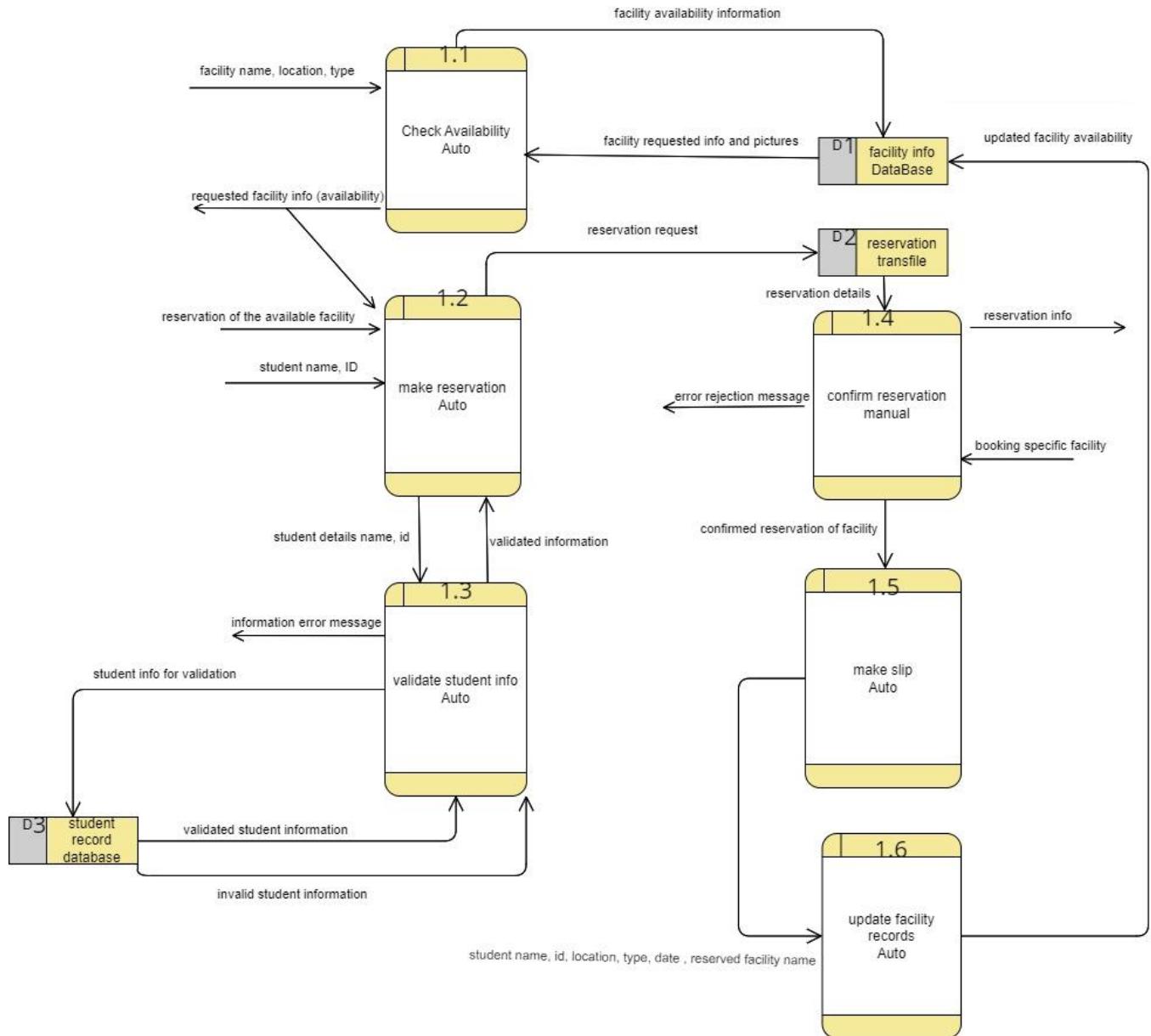
### **7.1 Physical DFD (TO-BE):**

#### **7.1.1 Physical DFD Diagram 0**

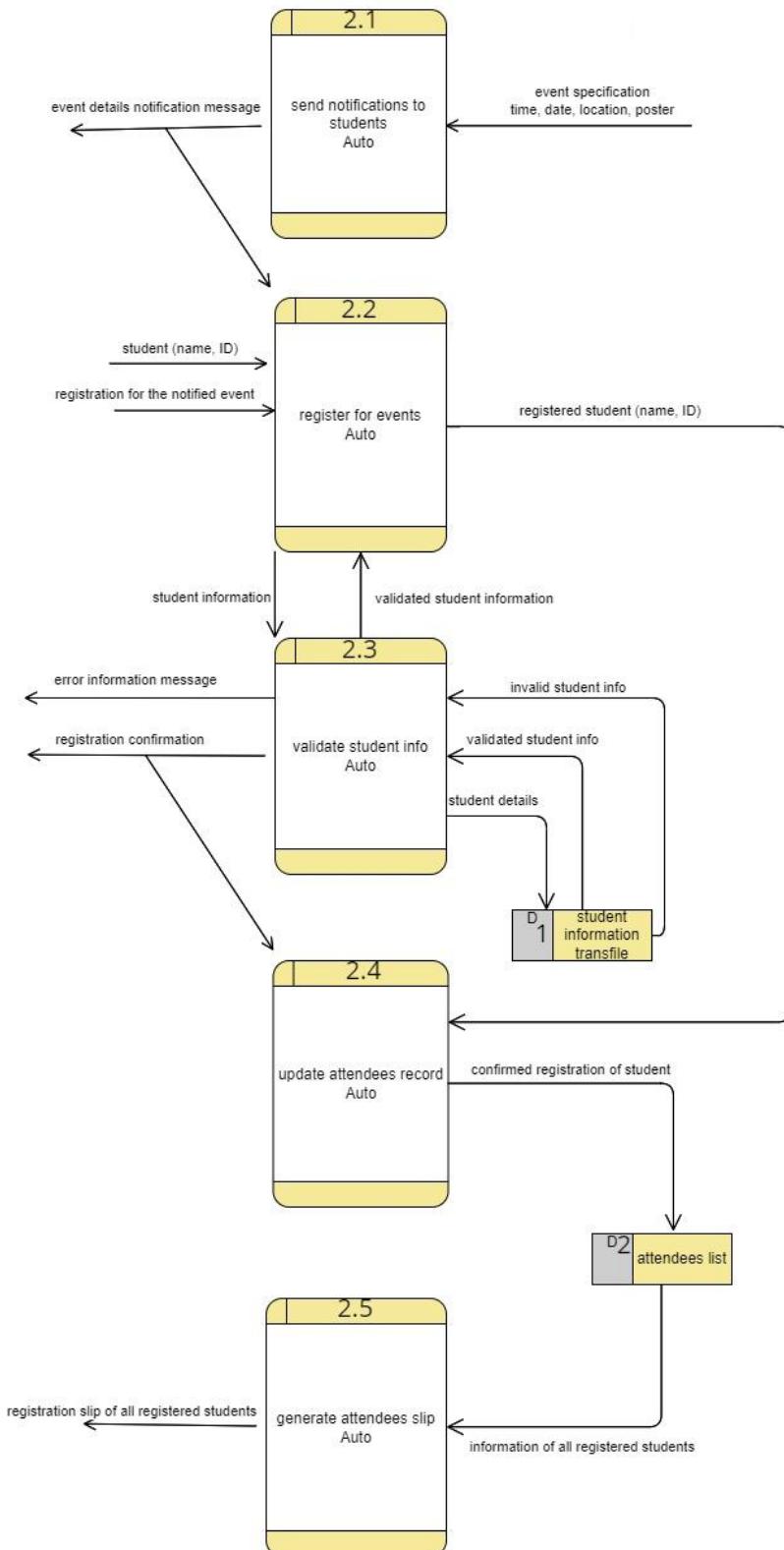
**(next page)**



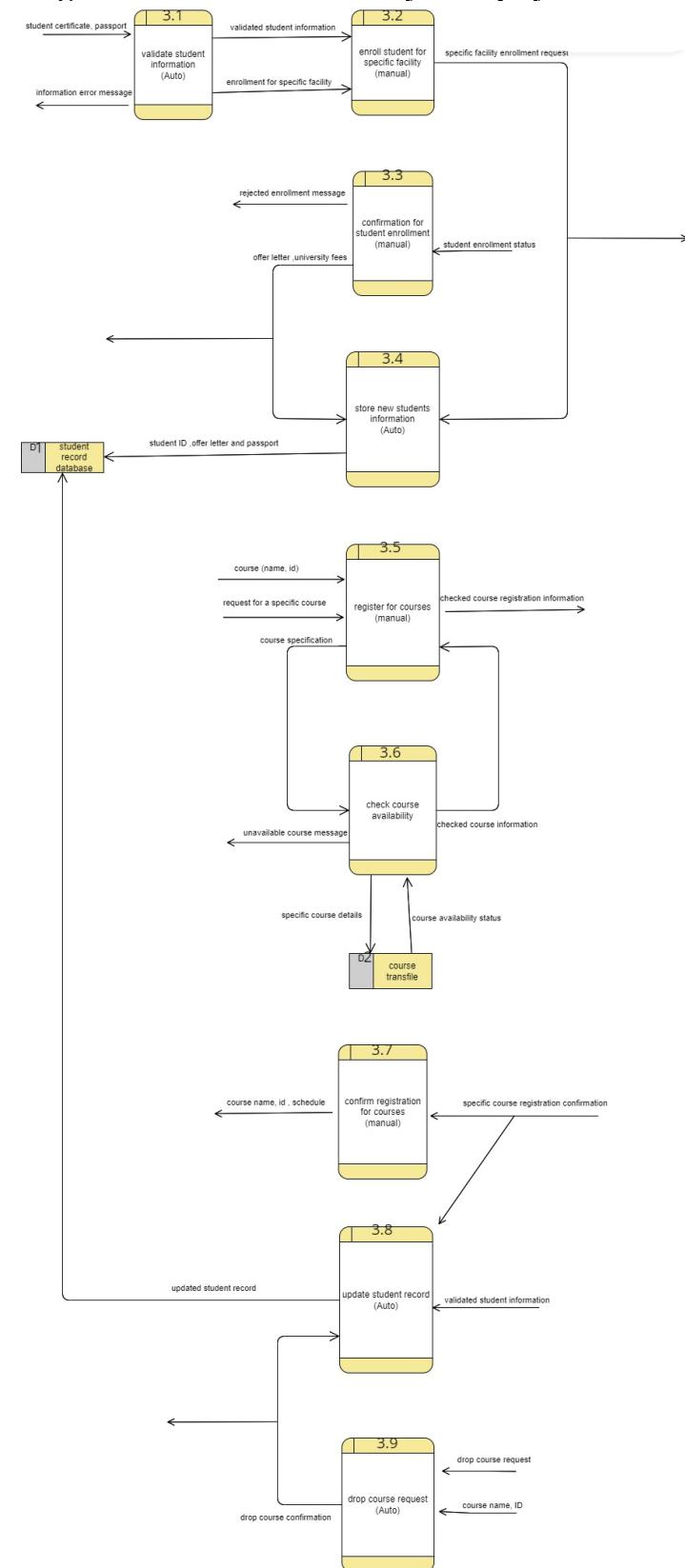
### 7.1.2 Make reservation subsystem Physical child diagram



### 7.1.3 register events subsystem physical child diagram

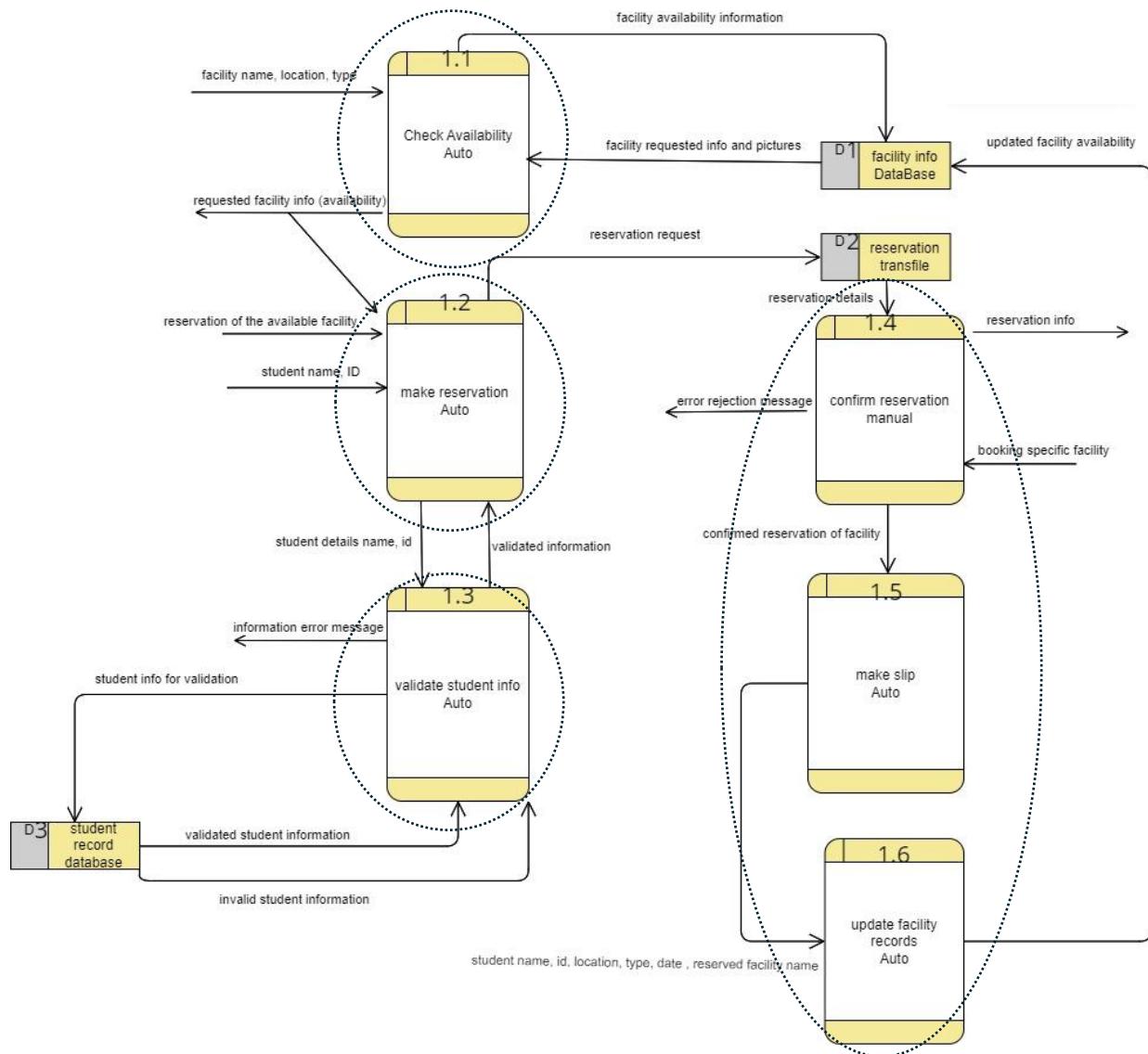


### 7.1.4 Manage student activities subsystem physical child diagram



## 7.1.5 Partitioning:

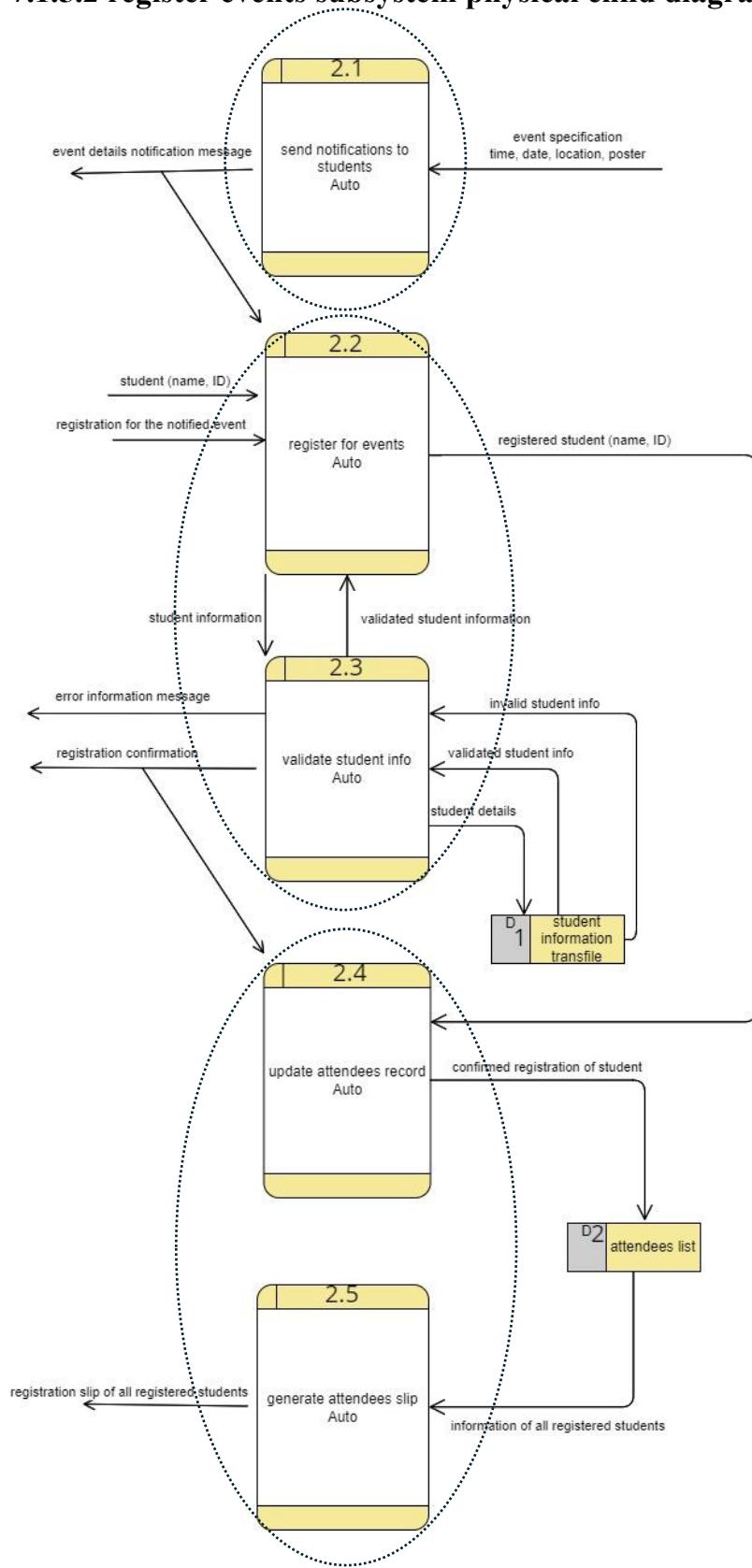
### 7.1.5.1 Make reservation subsystem Physical child diagram Partitioning:



Process 1.4, 1.5 and 1.6 can be done simultaneously, once the facility confirms the registration it can update the records as soon as the slip is generated.

The rest of the processes can't be grouped together.

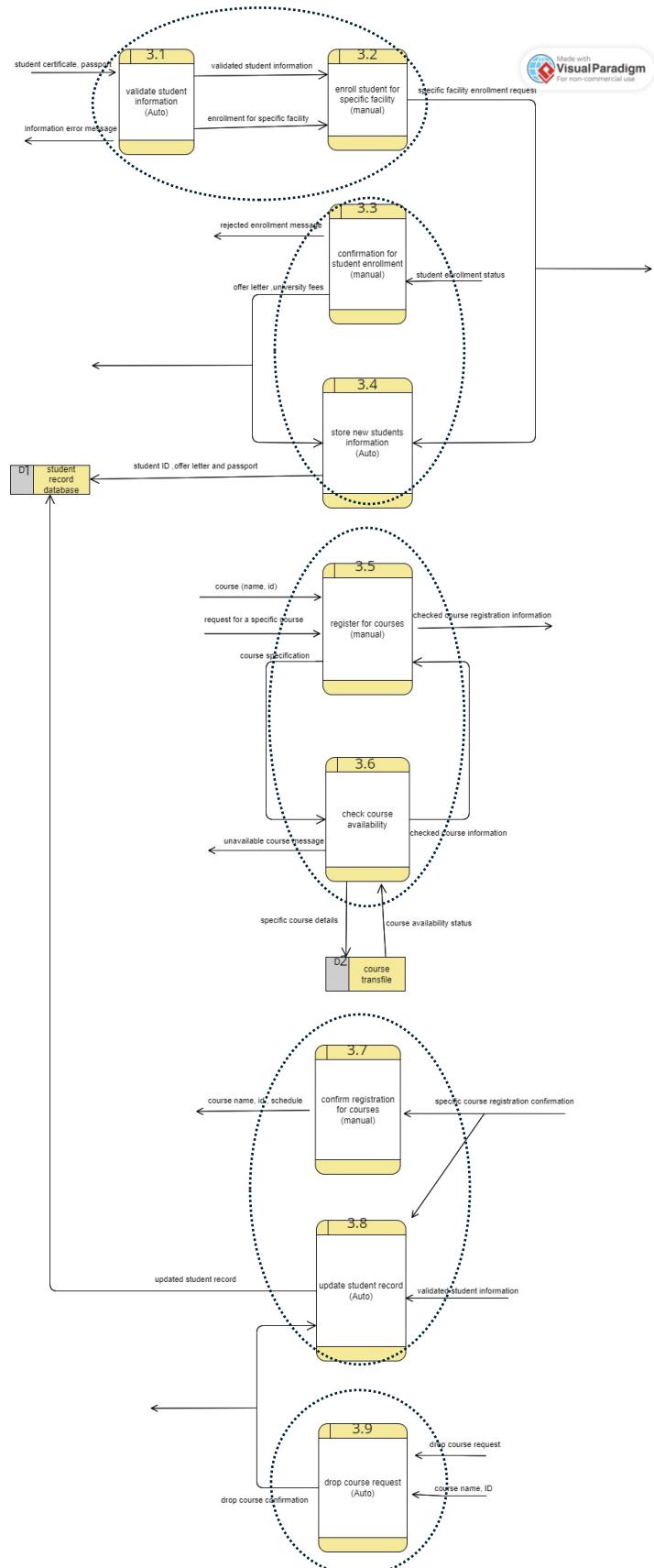
### 7.1.5.2 register events subsystem physical child diagram Partitioning:



Process 2.2 and 2.3 can be partitioned together since we can validate student details at the moment of registration and the student data can be stored in the same moment of registration.

Process 2.4 and 2.5 can be done simultaneously, the facility updates the attendees records as soon as the slip is generated.

Process 2.1 is left alone.



Process 3.1 and 3.2 can be partitioned together since we can validate student details now of registration and the student be enrolled immediately.

Process 3.3 and 3.4 can be partitioned together, we can store the student details now of confirmation.

Process 3.5 and 3.6 can be partitioned together since we can't register for a new course without checking for available courses. Thus, the process must be done simultaneously.

Process 3.7 and 3.8 can be partitioned together since we can validate student details now of courses registration and the student data can be stored in the same moment of registration.

Process 3.9 is left alone.

#### **7.1.6 CRUD matrix:**

There will be 3 CRUD matrix table for 3 subsystems, because we didn't do a child diagram for the notification subsystem that contains the last master file(the CRUD will be applied for only one process)

##### **7.1.6.1 Make reservation subsystem:**

Activity	Facility information database	Student record database
Check availability	R	
Make reservation		
Validate student information		R
Confirm reservation		
Make slip		
Update facility records	CUD	

#### **7.1.6.2 Register event subsystem:**

<b>Activity</b>	<b>Attendees list</b>
Send notification to students	
Register for events	
Validate student information	
Update attendees record	CRUD
Generate attendees slip	

### **7.1.6.3 Manage student activities subsystem:**

The R (read) is in the first CRUD matrix

<b>Activity</b>	<b>Student record database</b>
Validate student information	
Enroll student for a specific faculty	
Confirm student enrollment	
Store new student information	C
Register for courses	
Check course availability	
Confirm registration for courses	
Update student record	UD
Drop course request	

### 7.1.7 Event response table:

#### 7.1.7.1 Make reservation subsystem Event response table

Event	Source	Trigger	Activity	Response	Destination
Student checks facility availability	Student	Faculty name, location and type	Find the facility information and show it to the student	Facility information	Student
Student makes a reservation (reserve a facility)	Student	Reservation of the available facility, Student name	Send the reservation request to a temporary data store	Reservation request	Reservation transaction file
System validates student information	Make reservation process (indirectly student)	Student details (name and ID)	Search for student details and validating it	1. Validated information, 2. Information error message	1. Make reservation 2. process, Student
Booking management confirms the facility reservation by the student	Reservation transaction file	Reservation details	The booking management confirms or deny the reservation	1. Confirmed reservation of faculty, 2. Error rejection message	1. Make slip process, 2. Student
Make a slip of the re-served facility	Confirm reservation process	Confirmed reservation of facility	Making a slip of the reservation and send it to be updated in the database	Student name ,ID and re-served facility name, type, location and date	Update facility records process

Update the reserved facility information in the database	Make slip process	Student name ,ID and reserved facility name, type, location and date	Send the received data to the database and update the record of the reserved facility	Updated facility availability	Facility information database
--	-------------------	--	---	-------------------------------	-------------------------------

#### 7.1.7.2 register events subsystem Event response table

Event	Source	Trigger	Activity	Response	Destination
Event management send an event details to the students	Event management	Event specification Time, date, location and poster	Send the event information to the students	Event details notification message	Student
Student register for events	Student	Student name and ID, and registration of the notified event	Do the registration for events	Registered student (name and ID)	Update attendees record process
System validates student information	Register for events process (indirectly student)	Student Information (name and ID)	Search for student details and validating it	1. Validated student information, 2. Information error message 3. registration confirmation	1. Make reservation process, 2. Student
System updates attendees record	Register for events process	Registered student (name and ID)	The system sends the student information to attendees list	Confirmed registration of student	Attendees list

System produces slip for the event	Attendees list	Information of all registered students	The system produces a list for a specific events including the attendees information	Registration slip of all registered students	Event management
------------------------------------	----------------	--	--	--	------------------

#### 7.1.7.4 manage student activities subsystem Event response table

Event	Source	Trigger	Activity	Response	Destination
System validates the student information	Student	Student certificate, passport	The system checks if the student applicable to proceed to the enrollment information	1.Information error message, 2.Validated student information 3.enrollment for specific facility	Enroll student for specific facility process
The system sends the enrollment request to the administrator	Validate student information (indirectly student)	1.Validated student information 2.enrollment for specific facility	The system sends the enrollment request to the administrator  Containing student information	Specific facility enrollment request	Administrator
Students administrator sends enrollment status to student	Students administrator	Specific facility enrollment request	The administrator confirms or deny the enrollment request and send the status to the student	1.Rejected enrollment message, 2.offer letter, university fees	1.Student, 2.Store new students information process
System stores new student information	Confirm student enrollment process, 2.Enroll student for a specific facility process	1.Specific facility enrollment request 2.offer letter, university fees	The system takes the student information and enrollment request with other information and creates a new	Student ID, offer letter, passport	Student record database

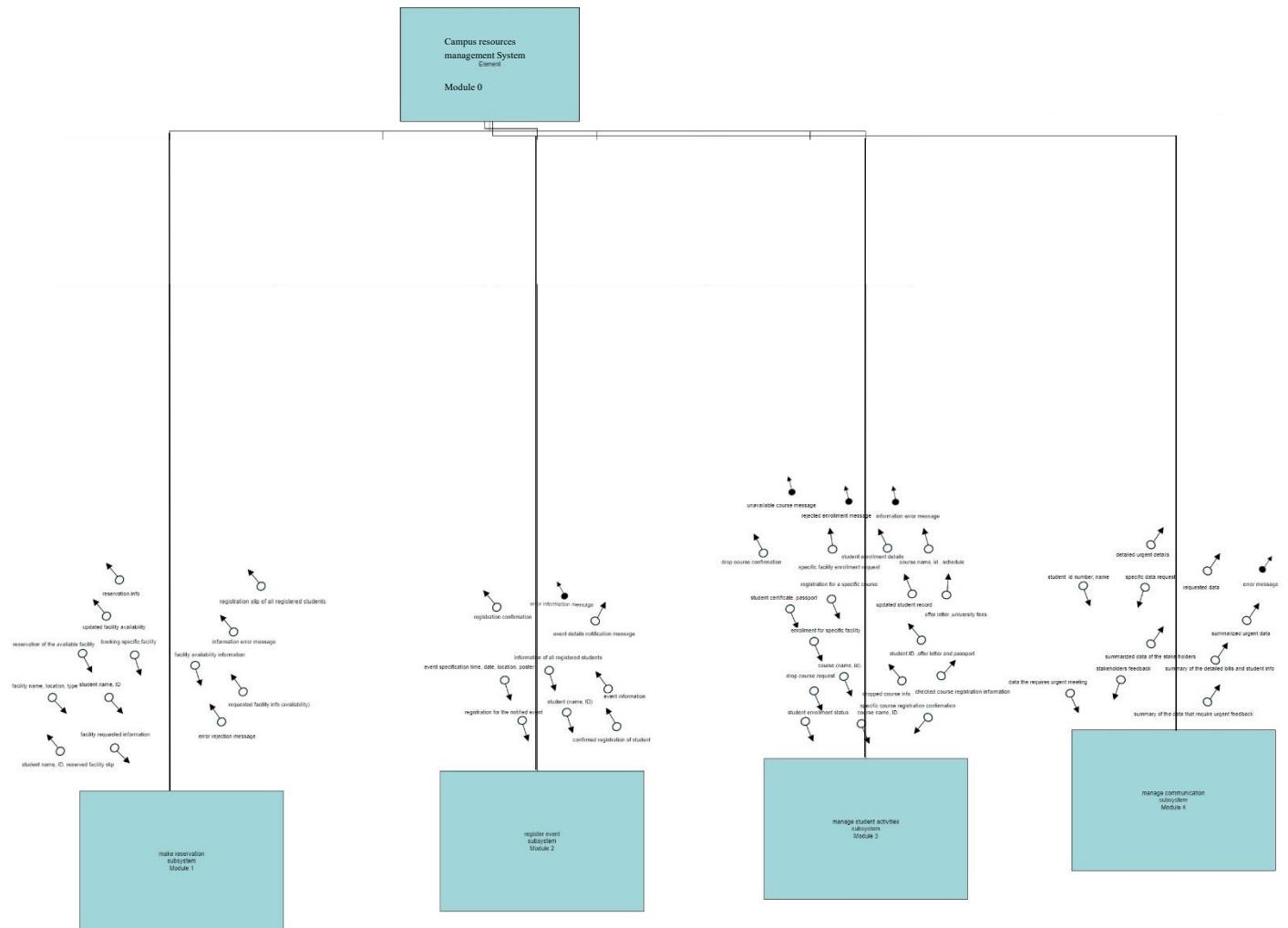
			student record in student record database		
Student register for courses	Student	1.Course (name, ID), 2.register for specific course request	The student sends the course data, and the system will first send request to check the course availability and then send it to the Students administrator	Checked course registration information	1. Students administrator, 2.Check course availability process
System checks for course availability	Register for courses process	Course specification	The system checks for the requested course with the courses in the course transaction file and send a message back	1.Unavailable course message, 2. checked course information	1.Student, 2.Check course availability process
Administrator confirms the course registration	Students administrator	Specific course registration confirmation	The students administrator confirms the registration and send the course information to the student	Course name, id, schedule	Student

System updates student record	1.Students administrator, 2.Validate student information process 3.drop course request process	1. Specific course registration confirmation 2.validated student information 3.drop course confirmation	The system sends the course information along with student information to update the student record	Updated student record	Student record database
Student drops a course	Student	1.Drop course request 2.Course name, ID	The system checks internally the number of courses that the student register for and then completes the drop and send the data to update student record process, also the status to the student	Drop course confirmation	1.Student 2.Update student record process

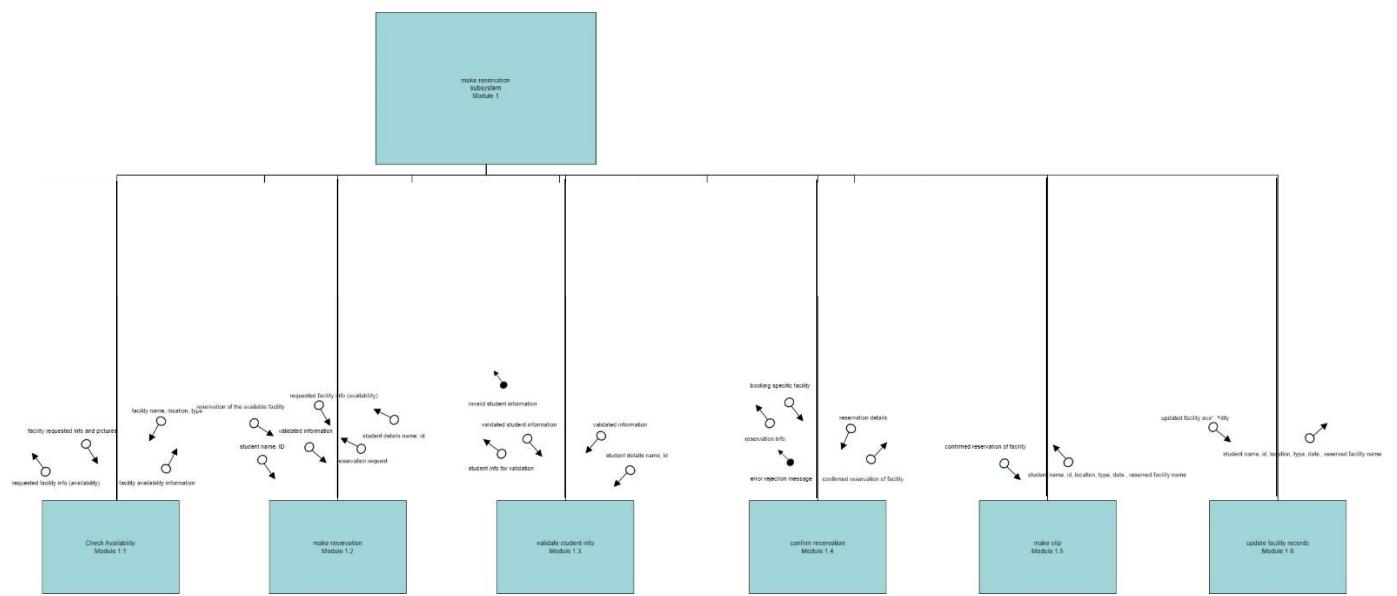
## 7.1.8 Structure chart:

We have a lot of data, so the diagram looks so complicated, so we decided to divide it into parts so it can be more readable.

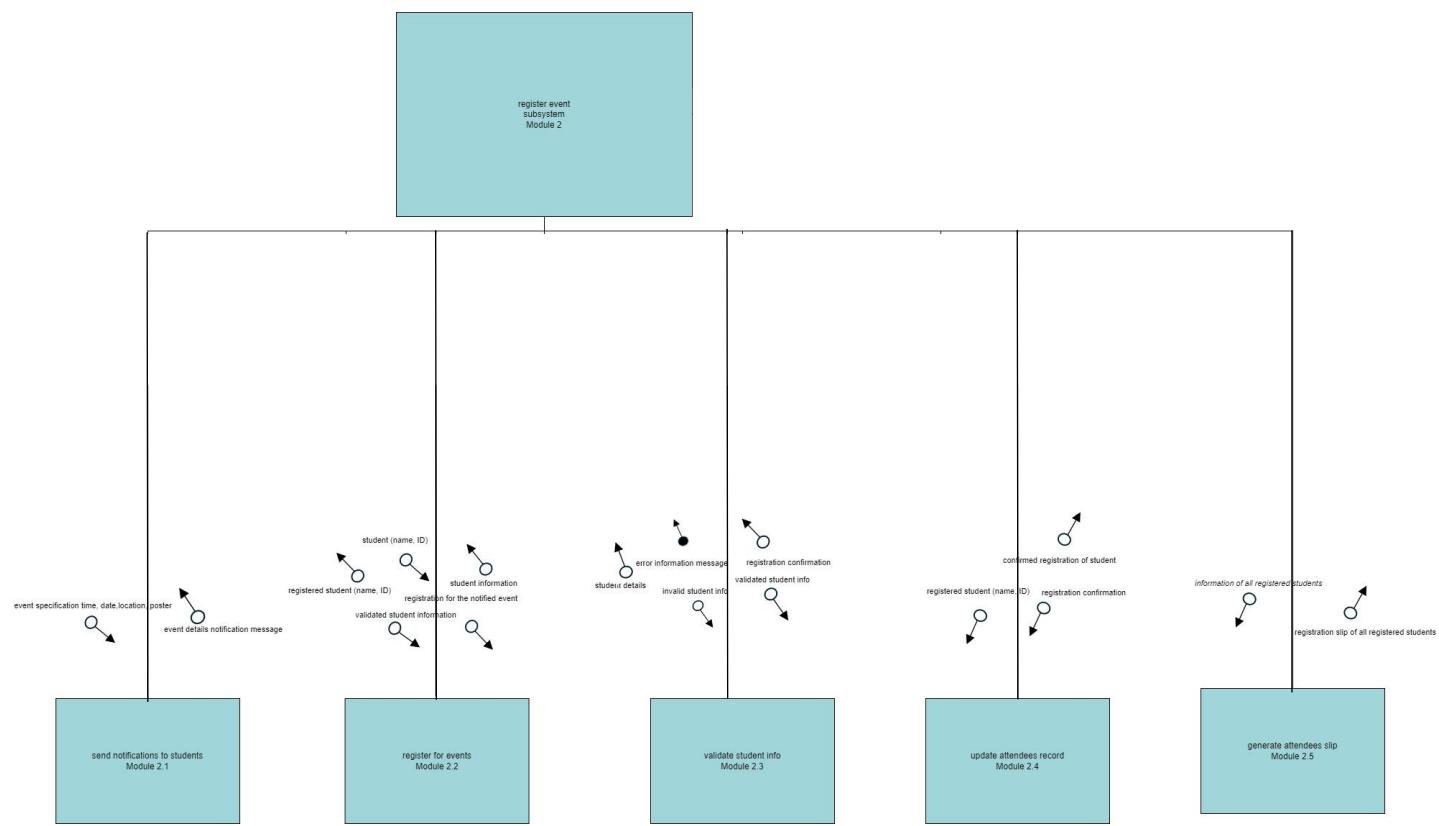
### 7.1.8.1 Diagram 0 Structure chart:



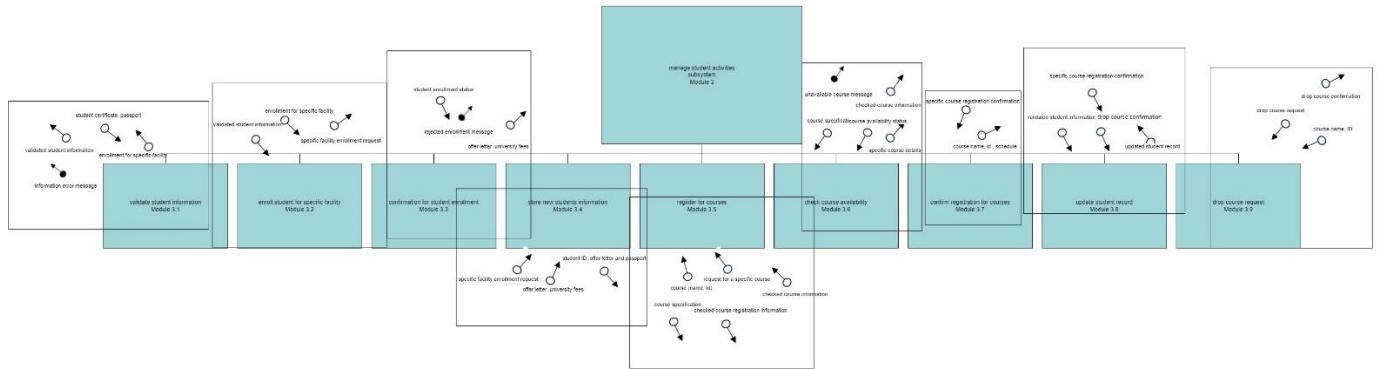
### 7.1.8.2 Module 1 Structure chart:



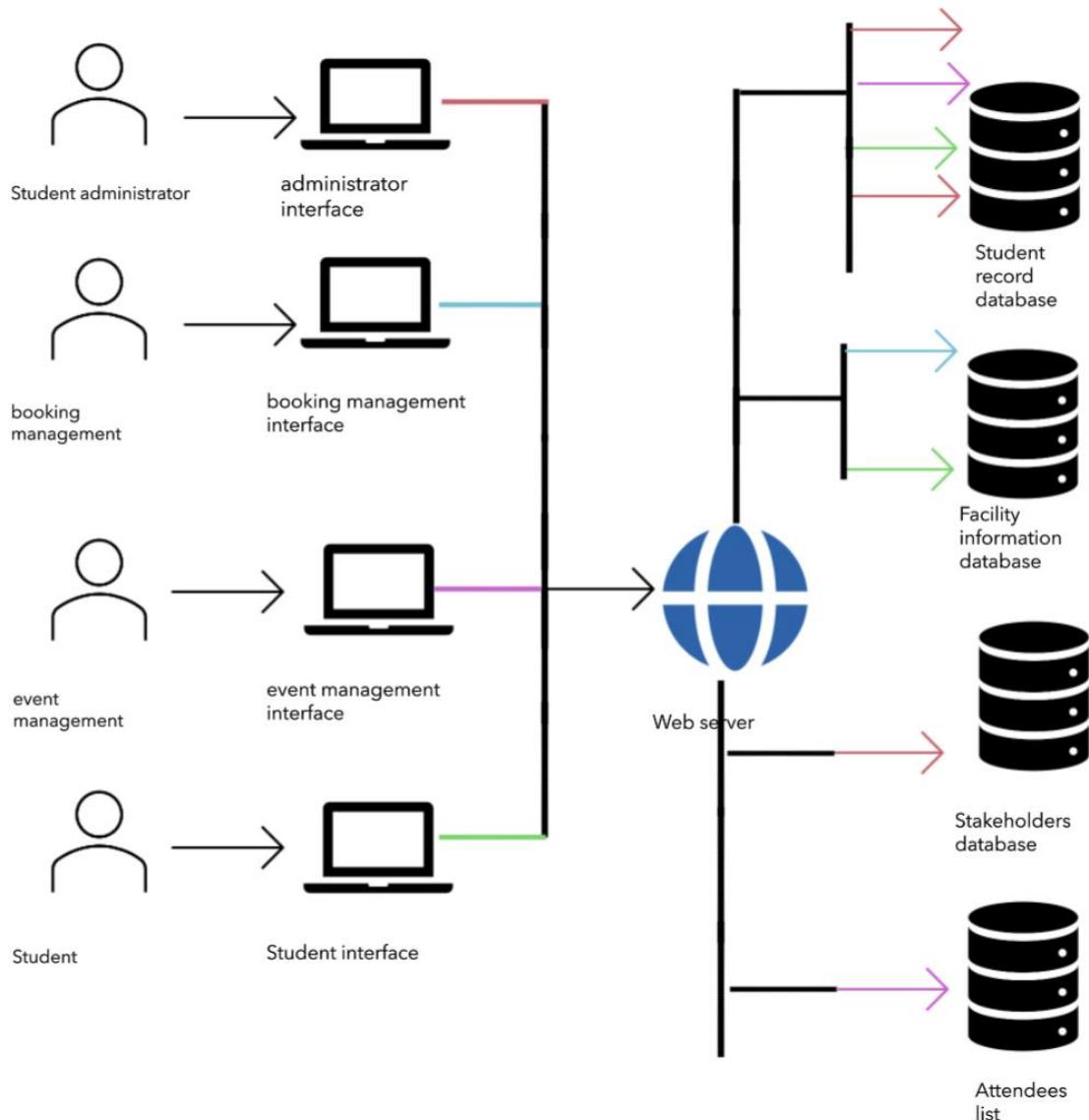
### 7.1.8.3 Module 2 Structure chart:



### 7.1.8.4 Module 3 Structure chart:



### 7.1.9 System architecture:



As you can see there are 4 user interface for each 4 main users, and all the users will access the data using a web server that will routes all the requests or access to the databases, we also have 4 databases each one contains information that regards each management with all the managements can access student records database for validation and some for other purposes

## 8.0 System wireframe:

This is the landing page where the user will choose if is a staff member or a student. You will see the consistency in the design with the colors and the header part also the navigation buttons.

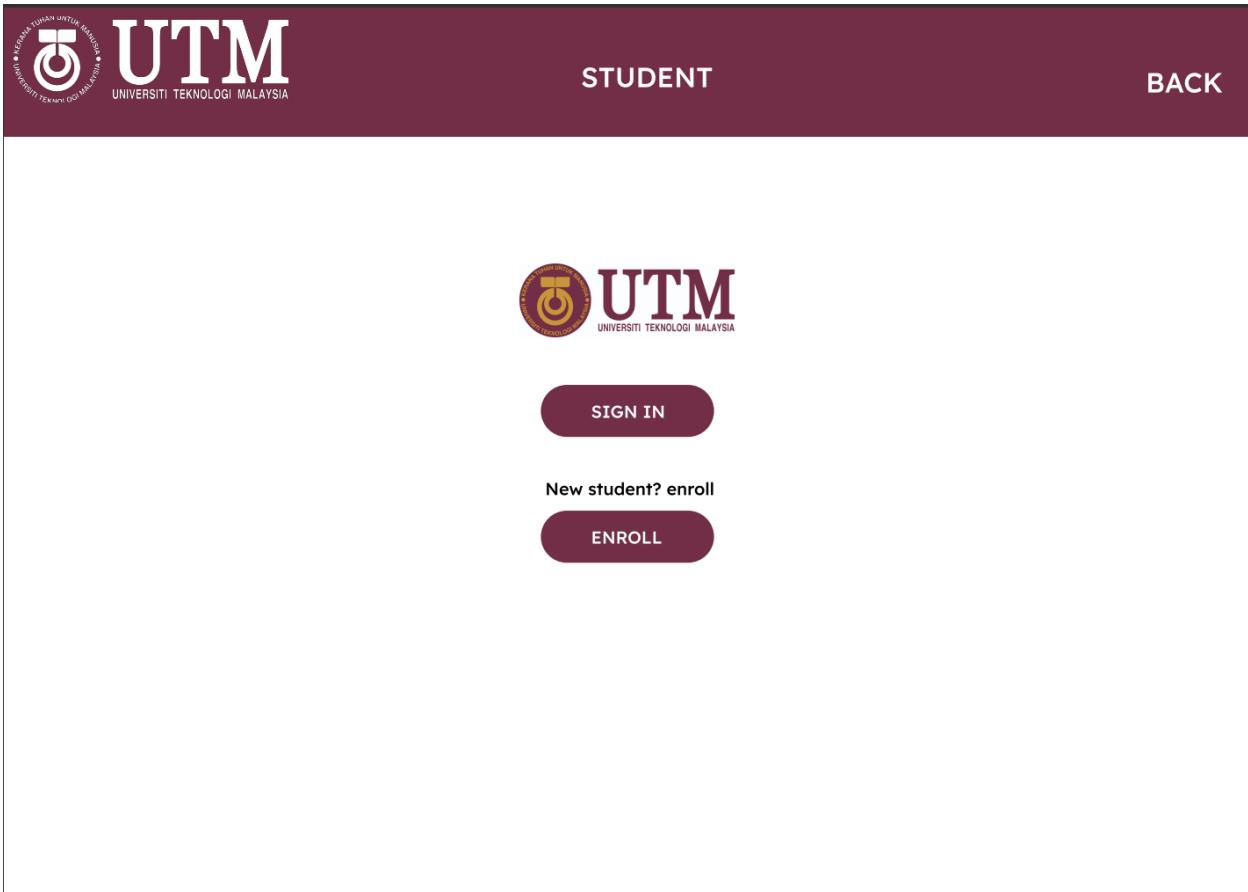


Student

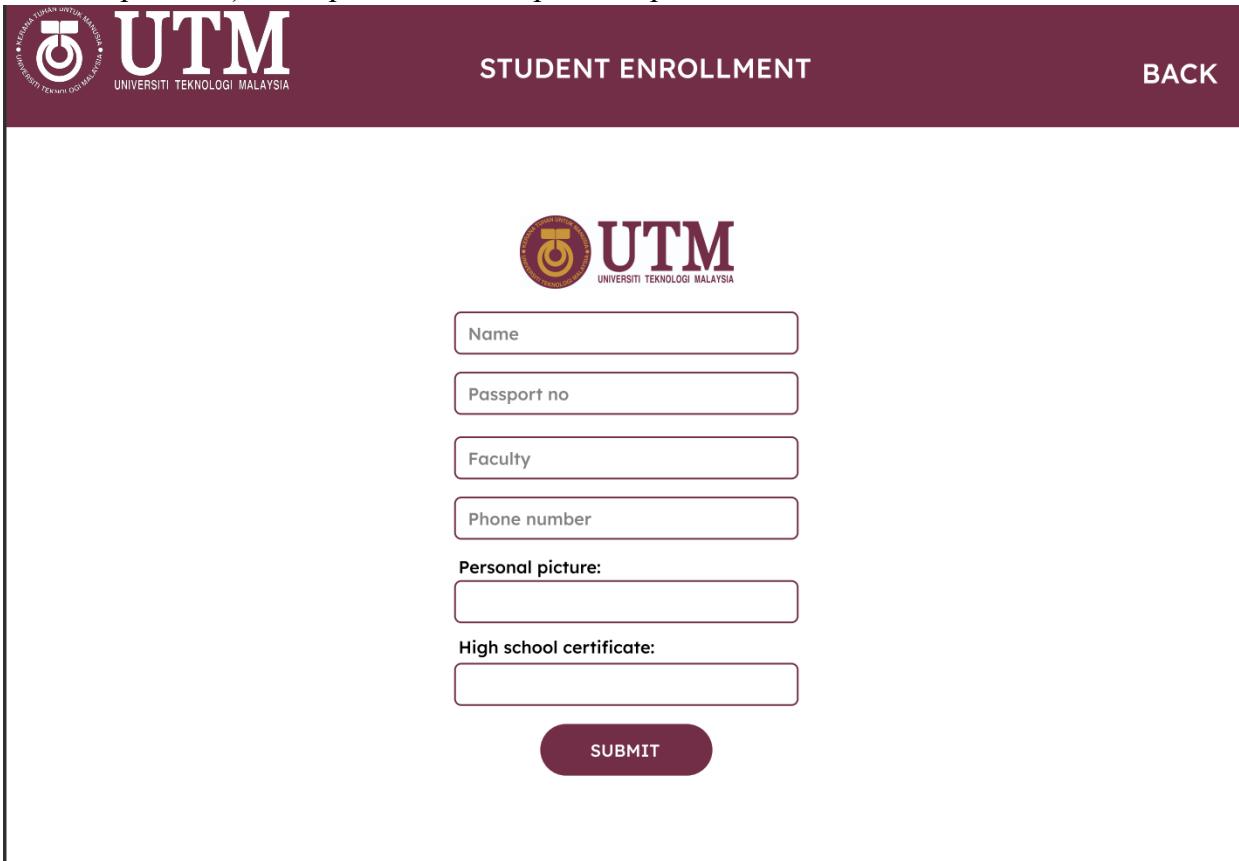
Staff

## 8.1 Student screens:

After the user click on “Student” button the system will navigate to this page, either sign in if the user is registered or enroll if the user is not registered, also there is a “BACK” button to return to the home page.



If the user clicks “ENROLL” button it will navigate to this page, where there are text boxes to input the student name and other information, with a drop down menu to choose the faculty (to reduce input error) and 2 places to attach personal picture and certificate,



The image shows a mobile application interface for "STUDENT ENROLLMENT". At the top, the UTM logo is displayed, followed by the text "STUDENT ENROLLMENT" and a "BACK" button. The form consists of several input fields: "Name", "Passport no", "Faculty", and "Phone number". Below these are two file upload fields labeled "Personal picture:" and "High school certificate:". A large "SUBMIT" button is located at the bottom of the form area.

After clicking “SUBMIT” button it shows this pop up to tell the student that the request is sent to the administrator.



Your Enrollment request is sent to the  
Student administrator

BACK

SUBMIT

IF the user clicks on “SIGN IN” button it navigates the student to this page, where there are 2 text boxes to enter the email and password to access the student account, after clicking “SIGN IN” button if the information is valid it will navigate the student to the home page.



SIGN IN

BACK

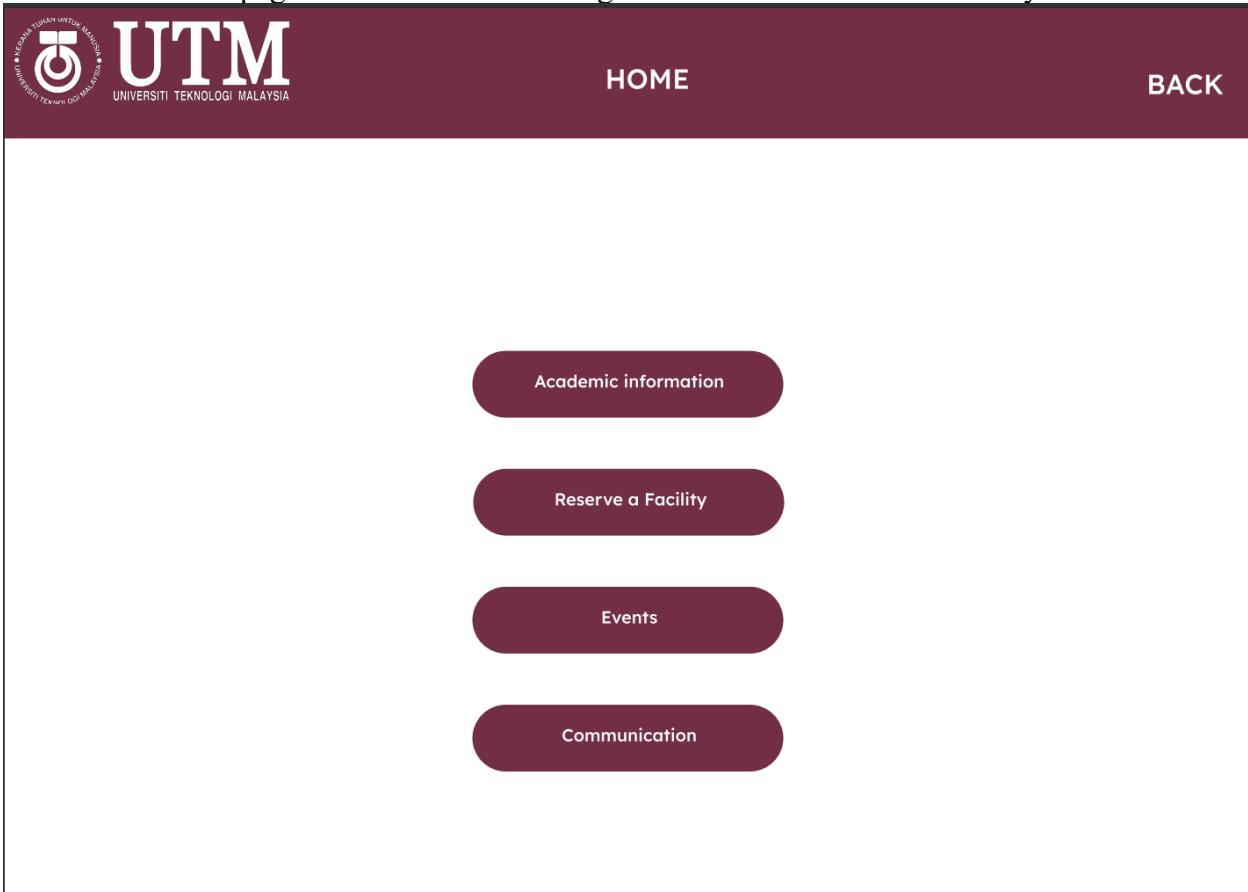


Email

Password

SIGN IN

The student home page has 4 buttons indicating what the student can do in the system.



A screenshot of the UTM Student Home Page. At the top left is the UTM logo with the text "UTM" and "UNIVERSITI TEKNOLOGI MALAYSIA". At the top right are two buttons: "HOME" and "BACK". Below the header, there are four dark red rounded rectangular buttons, each containing a white text label: "Academic information", "Reserve a Facility", "Events", and "Communication".

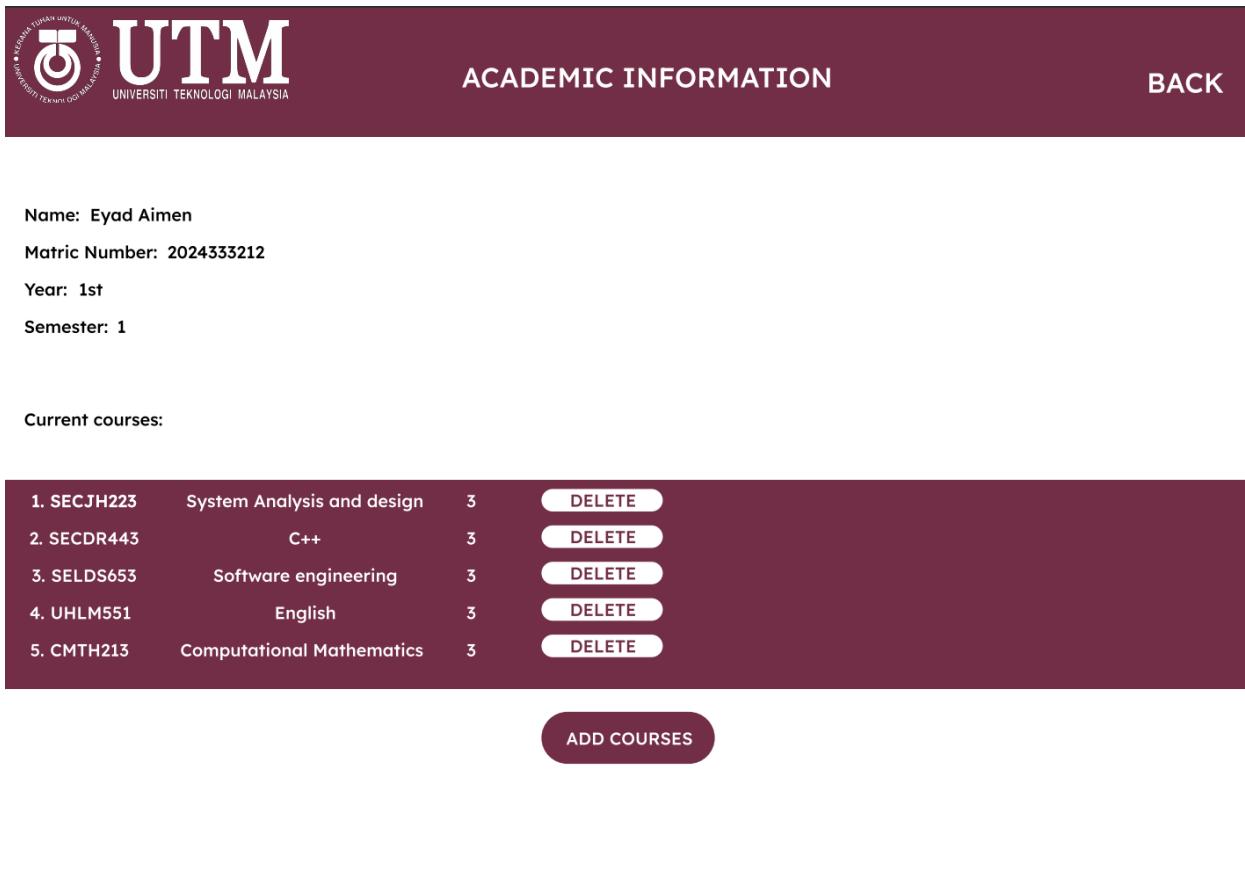
Academic information

Reserve a Facility

Events

Communication

If the student clicks on “Academic information” button it will navigate the student to this page, you can see the student information is displayed along side the registered courses, and 2 buttons one for adding a course and the other for dropping a course



The screenshot shows a mobile application interface for academic information. At the top, there is a dark header bar with the UTM logo on the left, the text "ACADEMIC INFORMATION" in the center, and a "BACK" button on the right. Below the header, the student's details are listed: Name: Eyad Aimen, Matric Number: 2024333212, Year: 1st, and Semester: 1. A section titled "Current courses:" lists five courses with their respective codes, names, credits, and "DELETE" buttons. At the bottom of the list is a "ADD COURSES" button.

1. SECJH223	System Analysis and design	3	DELETE
2. SECDR443	C++	3	DELETE
3. SELDS653	Software engineering	3	DELETE
4. UHLM551	English	3	DELETE
5. CMTH213	Computational Mathematics	3	DELETE

**ADD COURSES**

If the student clicks on ADD COURSES” button it will navigate the student to this page, where there is a textbox to enter the course code the student wishes to add and then clicks “SUBMIT” button.

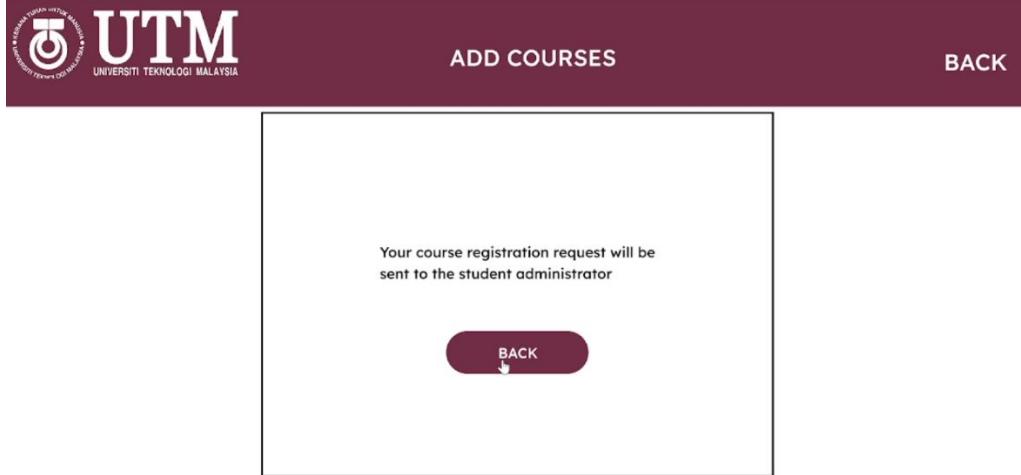


Enter course code:

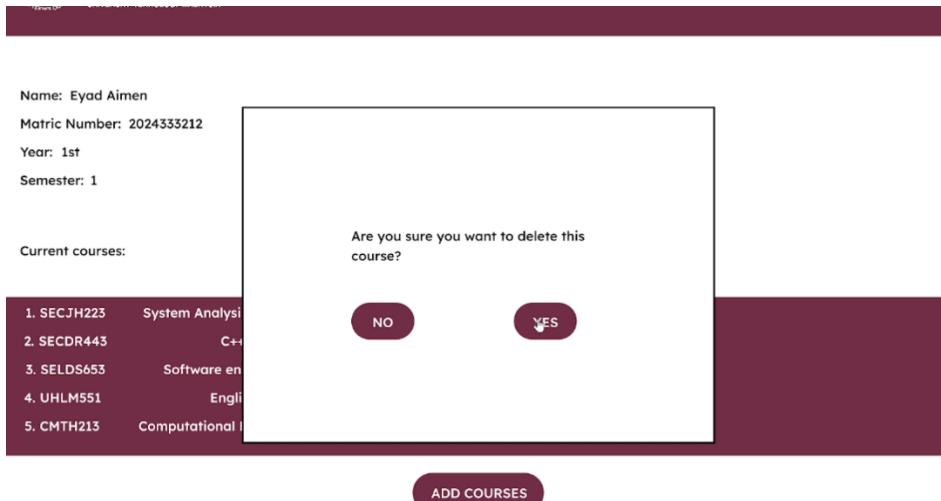
  
  

---

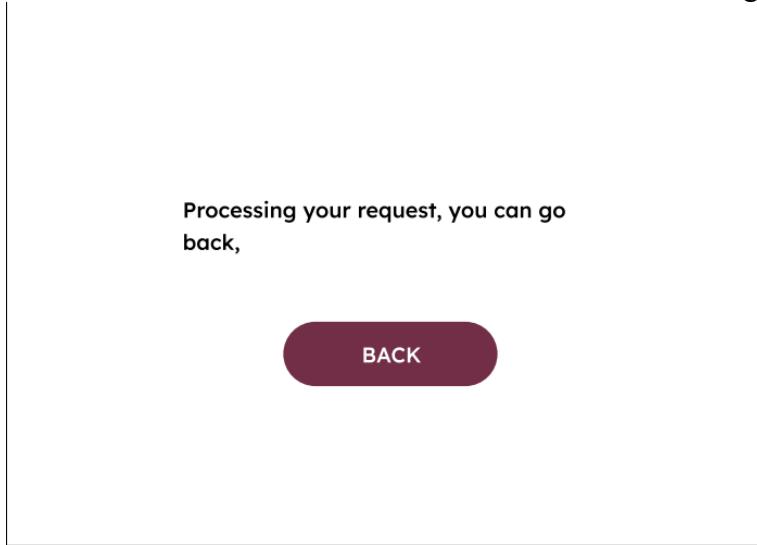
After clicking the “SUBMIT” button it will show this message



If the student clicks on “DELETE” button it will show this pop up.



If the student clicks “YES” button it will show this message



If the student presses “NO” button nothing will happen.

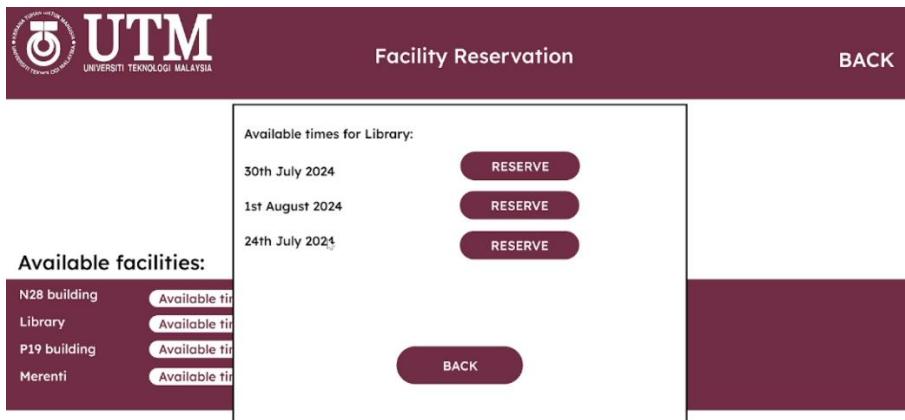
We go back to the home page of the student and see if the student clicks “Reserve a facility” button, it navigates the student to this page.



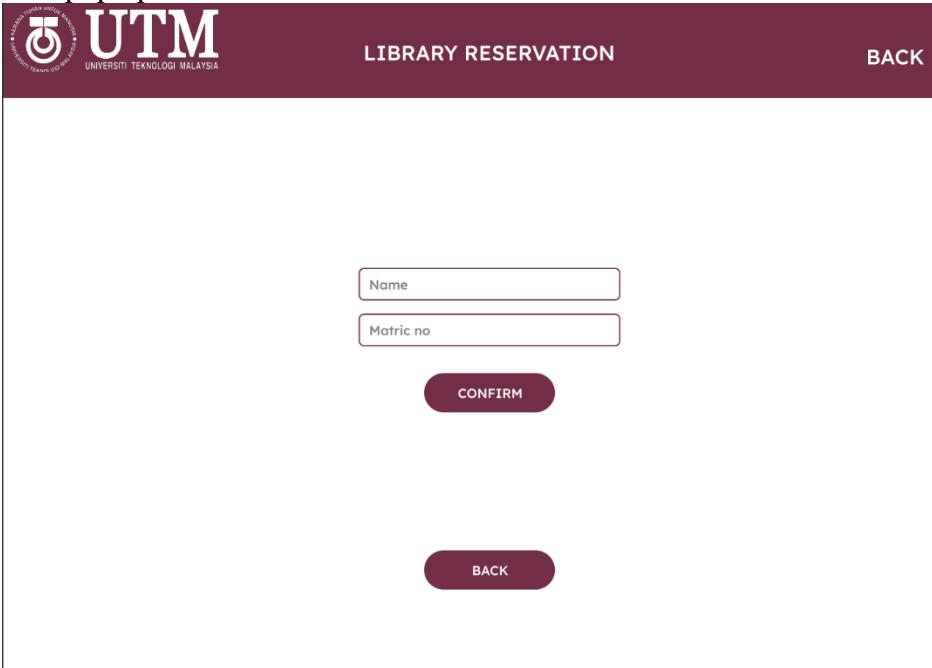
**Available facilities:**

N28 building	<a href="#">Available times</a>
Library	<a href="#">Available times</a>
P19 building	<a href="#">Available times</a>
Merenti	<a href="#">Available times</a>

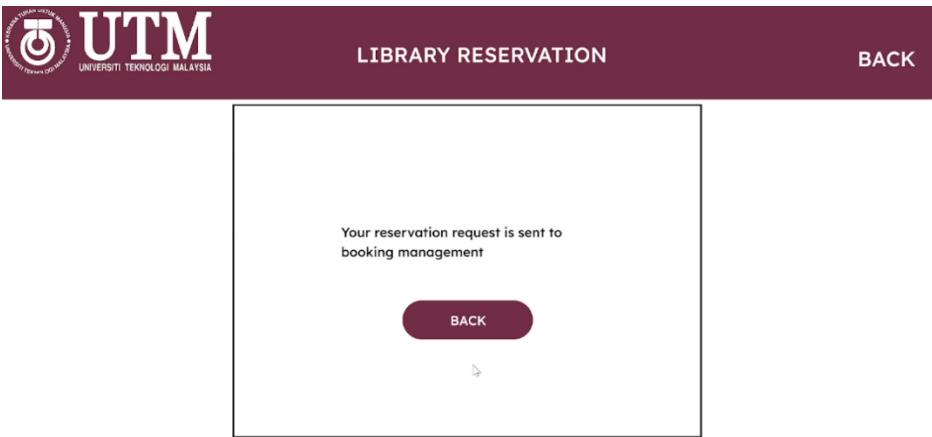
If the student clicks on “Available times” button, it will show a pop up that contains information of the selected facility, here is an example if the student clicks on the button for “Library”.



If the student clicks on “RESERVE” button it will show the next page, it has 2 textboxes to enter the student name and ID, and then the confirm button to complete the reservation request and the next pop up will be shown



Pop up:



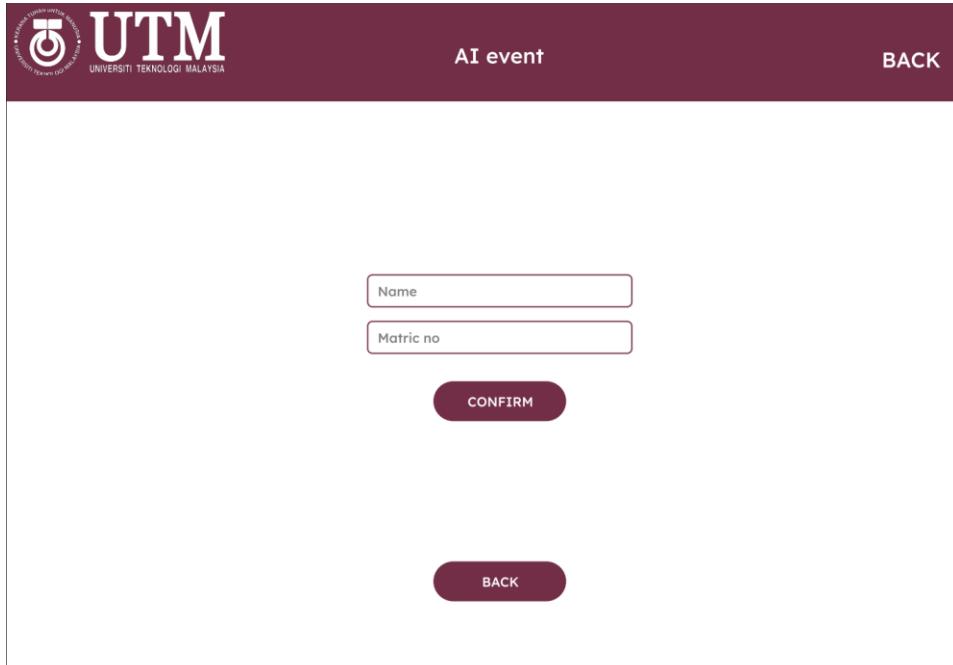
We go back again to the student home page, if the student clicks “Events” button it will show this page, it will show the upcoming events with their information, if the student clicks “REGISTER” it will navigate the student to a page regarding completing the registration for the event,



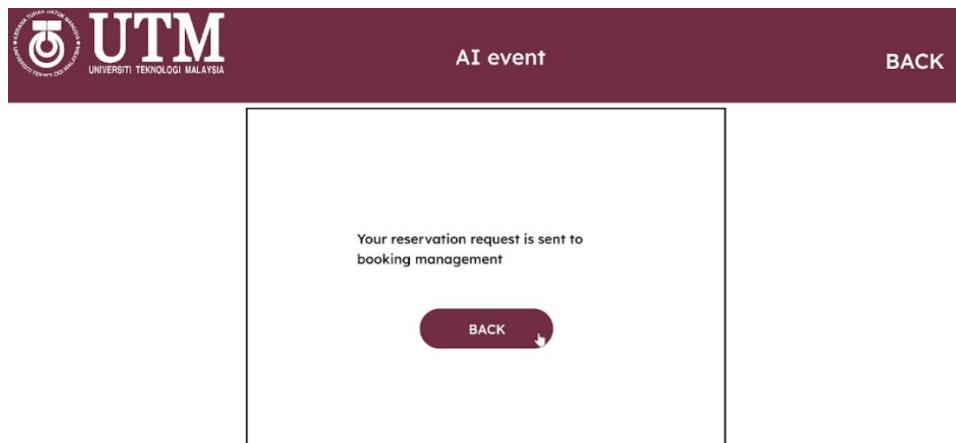
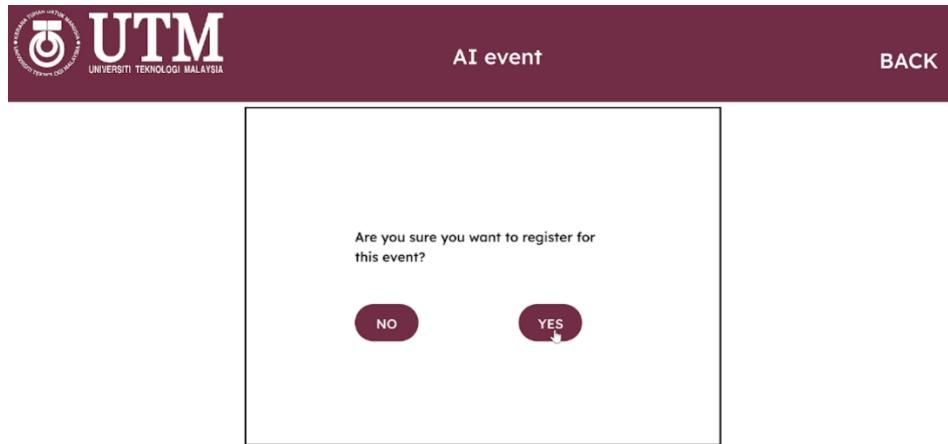
**Available facilities:**

AI event	15th-July-12PM to 2PM	N28A	<b>REGISTER</b>
Chess tournament	20th-July-12PM to 5PM	N28 Lounge	<b>REGISTER</b>
Problem solving competition	1st-August-10AM to 10PM	N28	<b>REGISTER</b>
New students welcoming event	10th-October-12PM to 2PM	N28A	<b>REGISTER</b>

Here is an example if the student clicks on “REGISTER” button for AI event.



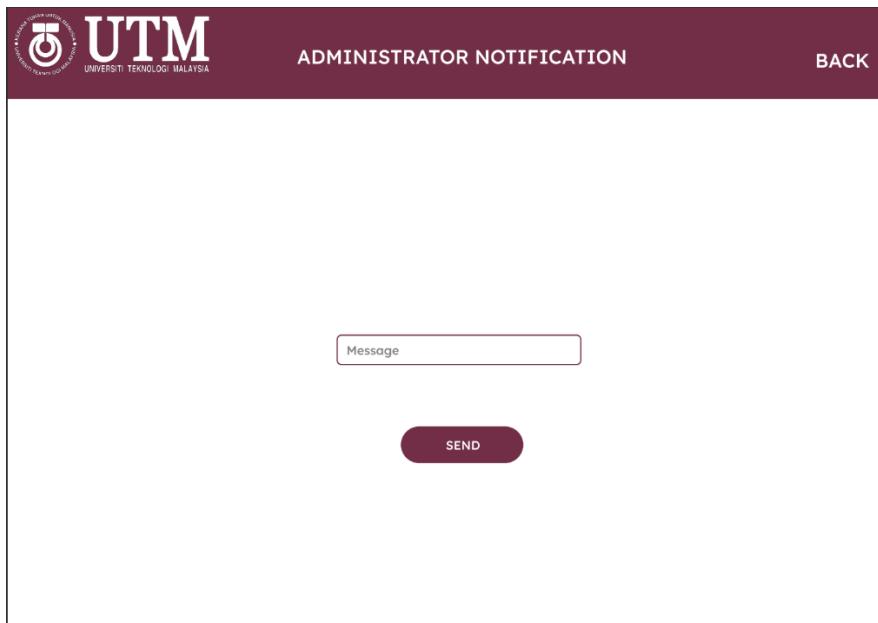
If the student clicks “CONFIRM” button it will show this pop up, if the student clicks “NO” nothing will happen, but if “YES” is clicked another pop up will be shown containing a message



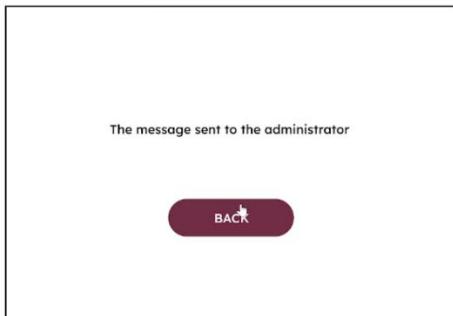
We go back again to the student home page and see if the student clicks on “Communication” button it will navigate to this page. It has 2 buttons the first one is to send a message to the administrator and the second one is to see the received messages.



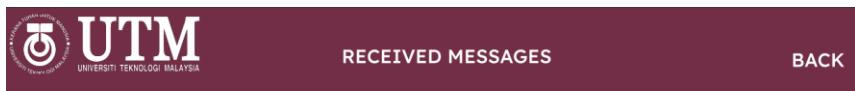
If the student clicks on “Notify administrator” button it will navigate to the next page, it has one textbox for the message and a button to sent the message.



If the student clicks “SEND” button the next pop up will be shown containing a message



Returning to the communication page, if the student clicks on “Received messages button” it will navigate to the next page. It contains all the received messages and the information about the senders, name type and etc...



Name	Matric	Type	
Othman Hassan	2024335212	Student	<a href="#">VIEW</a>
Ali	-	Administrator	<a href="#">VIEW</a>
Abdalla Ali	-	CS faculty dean	<a href="#">VIEW</a>

If the student clicks on “VIEW” button the next pop up will be shown containing the sent message. For example if the student clicks on the button for ALI this will be displayed.

The screenshot shows a mobile application interface for managing messages. At the top, there is a header bar with the UTM logo and the text "RECEIVED MESSAGES". On the right side of the header is a "BACK" button. Below the header, there is a table listing received messages. The table has two columns: "Name" and "Matric". The first message listed is from "Othman Hassan" with Matric number "202433321". The message content is: "Ali:  
Dear Eyad ,  
I would like to let you know that  
your courses registration has been  
completed". Below this message is a "BACK" button. The second message listed is from "Ali Abdalla Ali" with Matric number "-".

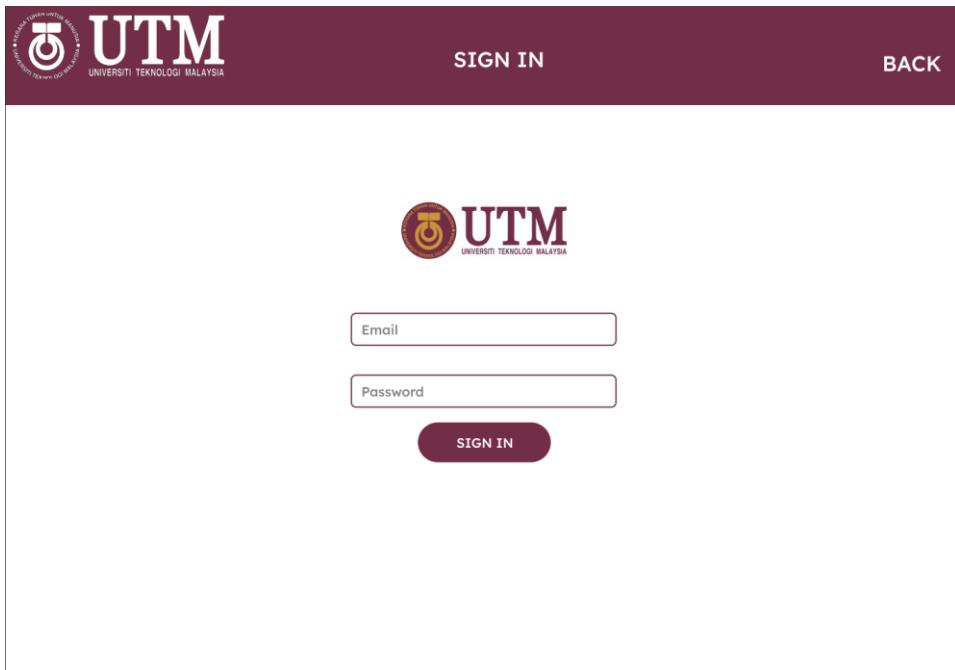
Name	Matric
Othman Hassan	202433321
Ali Abdalla Ali	-

Ali:  
Dear Eyad ,  
I would like to let you know that  
your courses registration has been  
completed

BACK

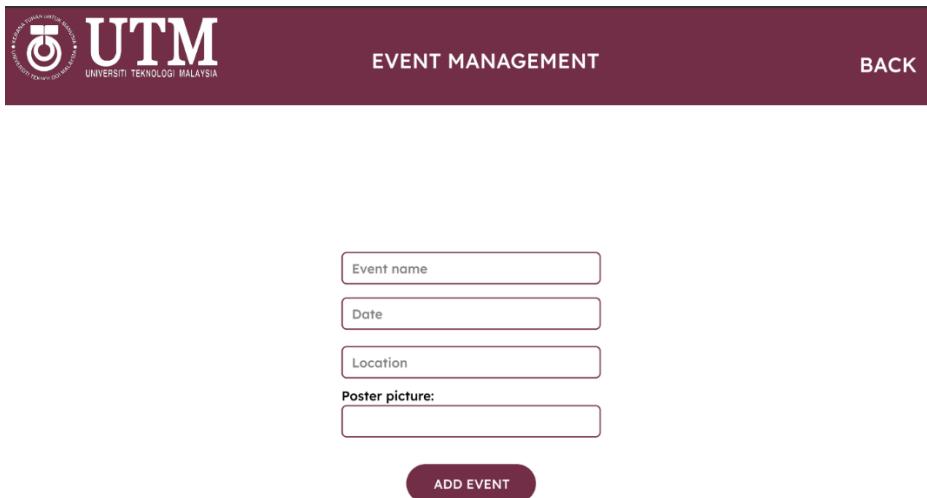
## 8.2 Staff screen:

When the user clicks “Staff” button it will navigate to the next screen

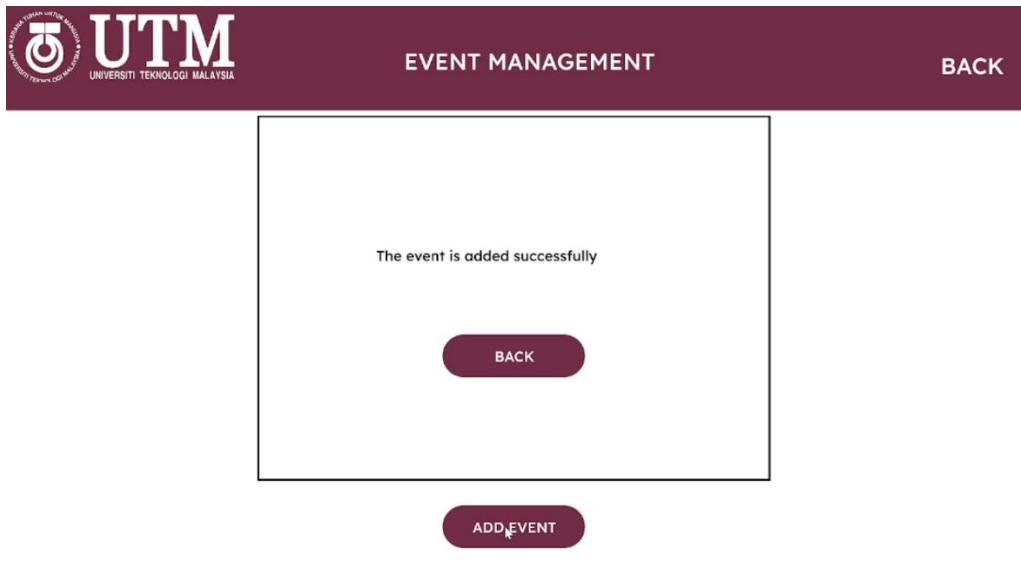


### 8.2.1 Event management screens:

If the user is from the Event management the next page will be displayed, it has 3 textboxes to enter event information and a place to attach a picture for the event.



When the user clicks “ADD EVENT” button a pop up with a message will be displayed.



### **8.2.2 Booking management screens:**

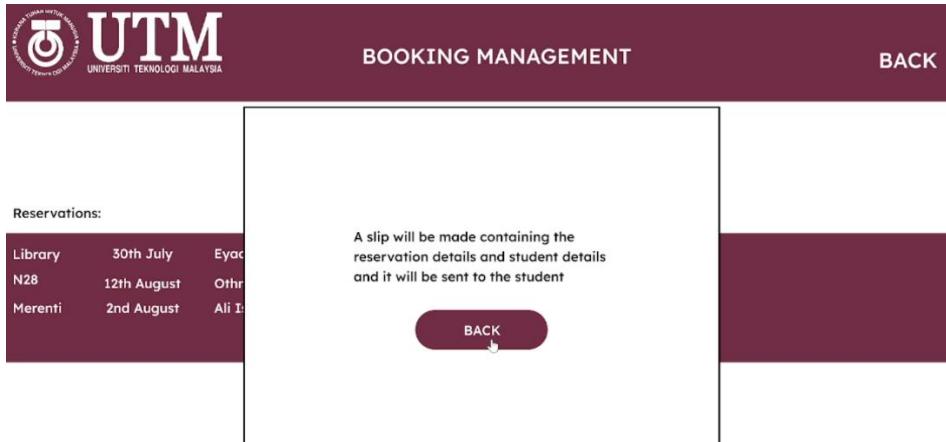
If the user is from the booking management the next page will be displayed. It has all the requests for reserving a facility with the student name and facility with the date. It has 2 buttons to confirm or reject the request



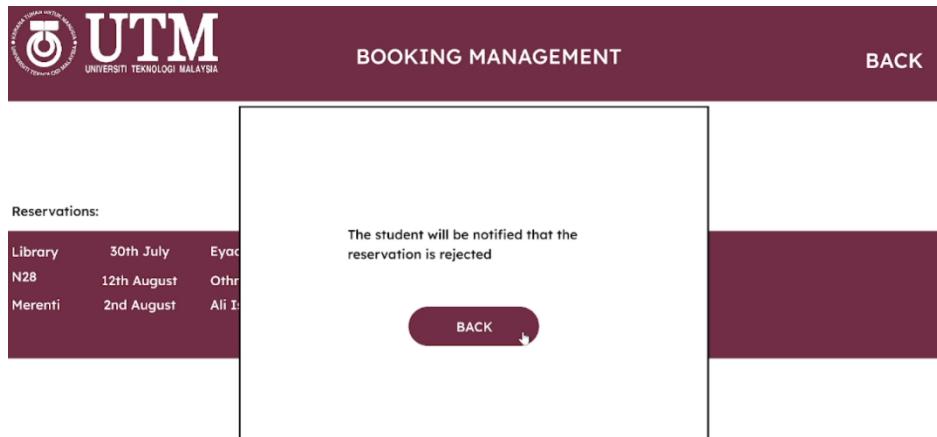
Reservations:

Library	30th July	Eyad Aimen	CONFIRM	REJECT
N28	12th August	Othman Hassan	CONFIRM	REJECT
Merenti	2nd August	Ali Isam	CONFIRM	REJECT

If the user clicks “CONFIRM” button the next pop up will be displayed.



If the user clicks “REJECT” button the next pop up will be displayed.



### 8.2.3 Student administrator screens:

If the user is from the student administration office this page will be displayed. It has buttons indicating the functions that the administrator can perform.

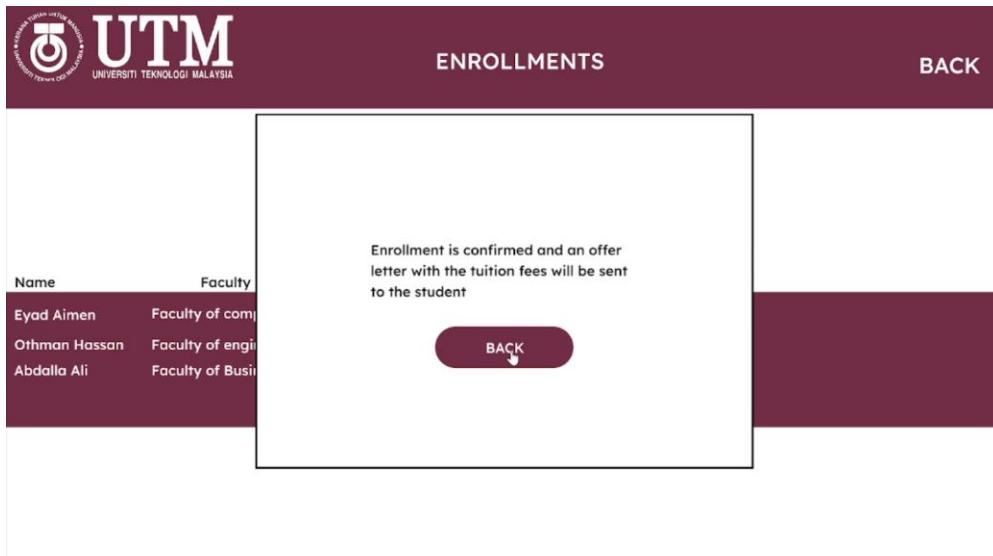


If the user clicks “Enrollment requests” this page will be displayed. It has the names and faculties of the students they would like to enroll for, and 2 buttons to either accept or reject the request

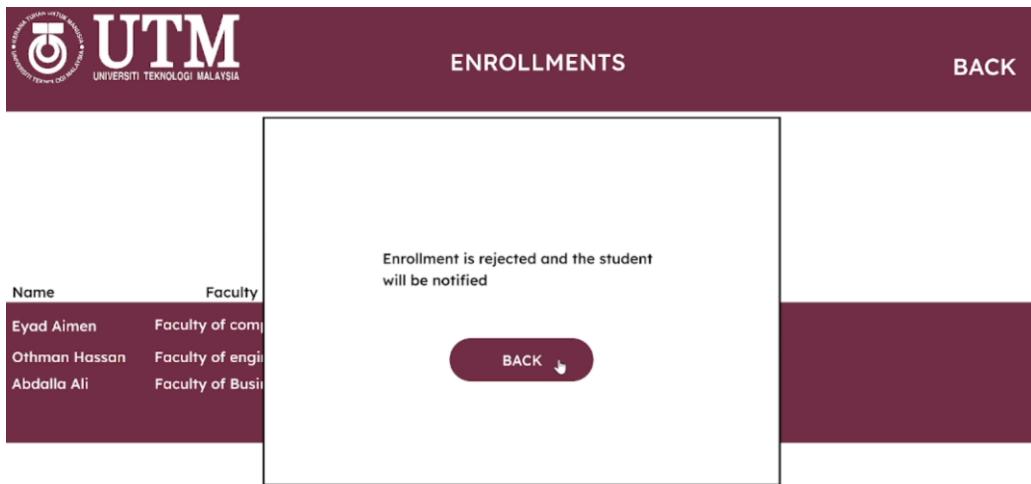


Name	Faculty	passport no		
Eyad Aimen	Faculty of computing	2487543	CONFIRM	REJECT
Othman Hassan	Faculty of engineering	5854943	CONFIRM	REJECT
Abdalla Ali	Faculty of Business	463454	CONFIRM	REJECT

If the user clicks “CONFIRM” this pop up will be displayed.



If the user clicks “REJECT” this pop up will be displayed.



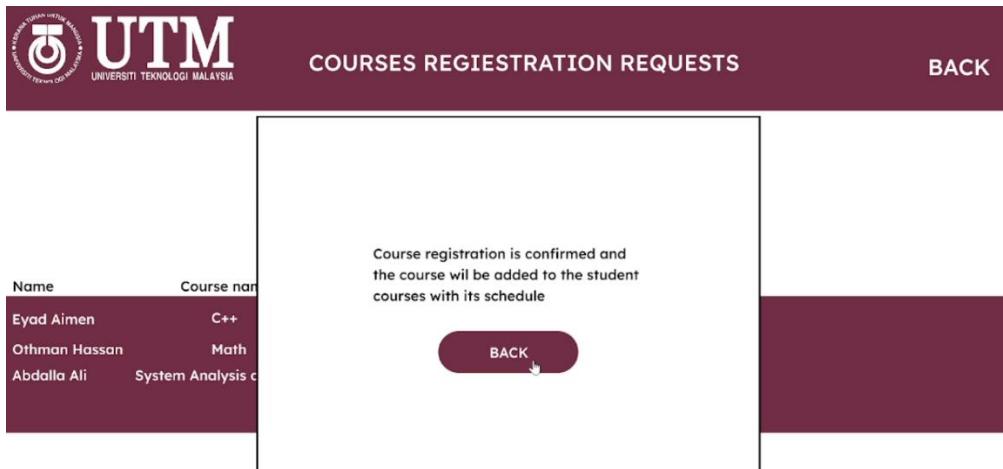
If the user clicks "Courses registration requests" button this page will be displayed. It has the student name, course name, course code and 2 buttons to reject or confirm the request.



The screenshot shows a dark purple header bar with the UTM logo on the left, the text "COURSES REGISTRATION REQUESTS" in the center, and a "BACK" button on the right. Below the header is a table listing three student entries with columns for Name, Course name, Course code, and two buttons each: "CONFIRM" and "REJECT".

Name	Course name	Course code	CONFIRM	REJECT
Eyad Aimen	C++	SECDR443	CONFIRM	REJECT
Othman Hassan	Math	CMTH213	CONFIRM	REJECT
Abdalla Ali	System Analysis and Design	SECJH223	CONFIRM	REJECT

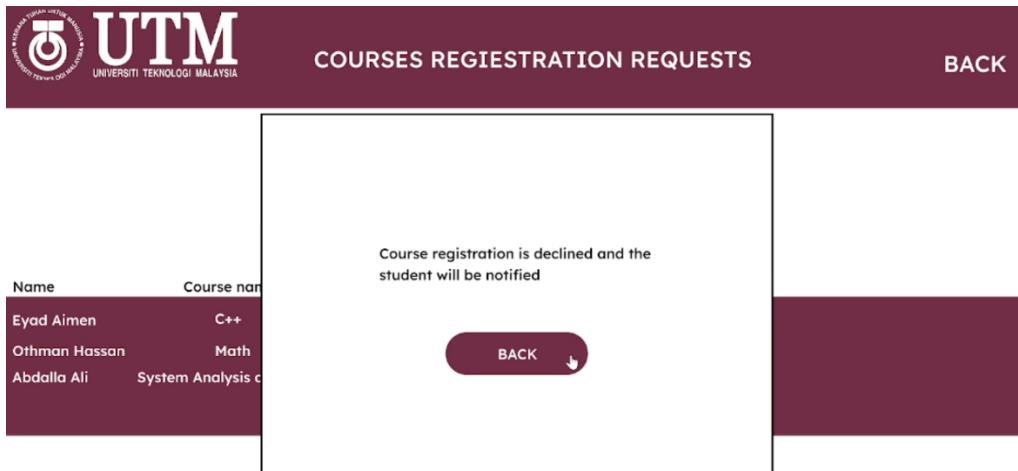
If the user clicks “CONFIRM” this pop up will be displayed.



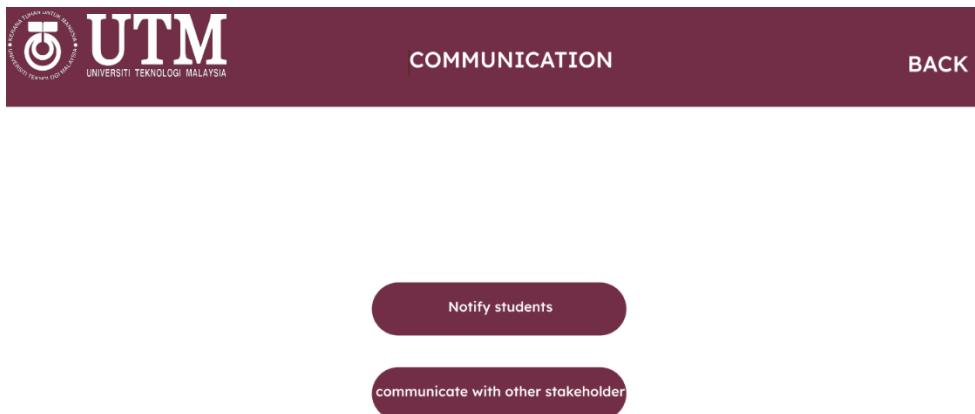
The screenshot shows a dark purple header bar with the UTM logo on the left, the text "COURSES REGISTRATION REQUESTS" in the center, and a "BACK" button on the right. A large white pop-up window is centered in the middle. Inside the pop-up, there is a message: "Course registration is confirmed and the course will be added to the student courses with its schedule." At the bottom of the pop-up is a "BACK" button.

Name	Course name
Eyad Aimen	C++
Othman Hassan	Math
Abdalla Ali	System Analysis and Design

If the user clicks "REJECT" this pop up will be displayed.



Returning to the administrator home page if the user clicks "Communication" button this page will be displayed. It has 2 buttons, one for communicating with students and the other for the other stakeholders (Dean, teachers, other staff member, etc..)



If the user clicks “Notify students” this page will be displayed. It has 2 textboxes one to enter the student ID and the other is to enter the message for the user.



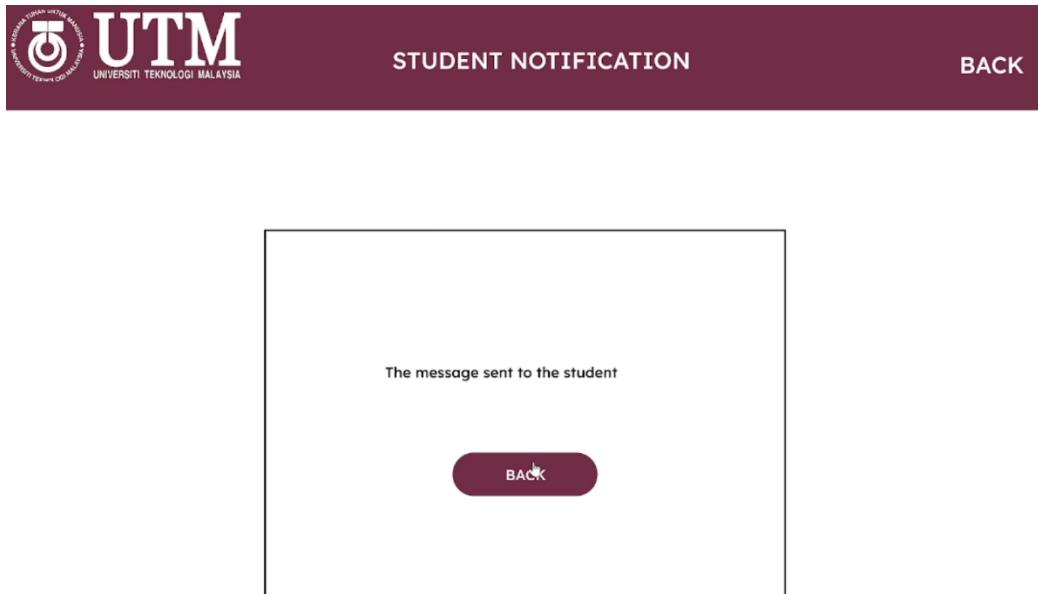
Student ID

Message

SEND

A form consisting of two input fields and a central "SEND" button. The first field is labeled "Student ID" and the second is labeled "Message". Below these is a rounded rectangular button labeled "SEND".

If the user clicks “SEND” button this pop up will be displayed.



If the user clicks "Communicate with other stakeholder" button this page will be displayed. It has one drop down menu to select which stakeholder the user wishes to send a message to, and a text box for the message to be sent



If the user clicks “SEND” this pop up will be displayed.



Returning to the administrator home page if the user clicks "Received messages" button this page will be displayed. It contains all the received messages with the names of the senders and their type (students, teachers and etc...). if the user clicks "VIEW" a pop up will be displayed with the received message.

Name	Matric	Type	
Eyad Aimen	2024333212	Student	<button>VIEW</button>
Othman Hassan	-	Administrator	<button>VIEW</button>
Abdalla Ali	-	CS faculty dean	<button>VIEW</button>

For example, if the user clicks the first "VIEW" button this pop up will be displayed.

|Eyad Aimen:  
Dear Administrator,  
I want to see all my registered  
courses.

[BACK](#)

## **9.0 summary of the proposed system:**

The new system introduces the new communication feature that allows the staff members and students to communicate with each other and share important messages. The new system also provides a common place to do all related functions, reduces the input error and simplifies the processes. It also improves the user experience and the productivity of the users (because they were open to change, and they liked the new system).