**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:
   1. Student information.
   2. Reservation details.
2. Process:
   1. Checking for facility availability.
   2. Confirming reservation.
   3. Issuing slip.
3. Output:
   1. 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.

1. 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:

* 1. Student information.
  2. Event details.

1. Process:
2. Checking for facility availability.
3. Confirming event registration.
4. Writing student name in the attendance list.
5. Output:
   1. 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:
   1. Student information.
   2. Courses details.
2. Process:
   1. Enrolling the new student.
   2. Issuing student card.
   3. Registering courses.
   4. Storing student information and registered records.
   5. Providing student with the schedule
3. Output:
   1. Enrollment slip.
   2. Student card.
   3. 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.

1. 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:
   1. Data for summarization.
   2. Student information (contact information, results).
2. Process:
   1. Summarizing data.
   2. Sending result to students or other management departments.
3. Output:
   1. Summarized data.
   2. 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.

1. 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**

A diagram of a cell phone

Description automatically generated

**5.2 DFD ZERO DIAGRAM FOR THE SYSTEM**

A diagram of a computer

Description automatically generated

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

A diagram of a reservation system

Description automatically generated

**5.4 Communication and notification management subsystem Child diagram**

A diagram of a data flow

Description automatically generated

**6.0 System Analysis and Specification:**

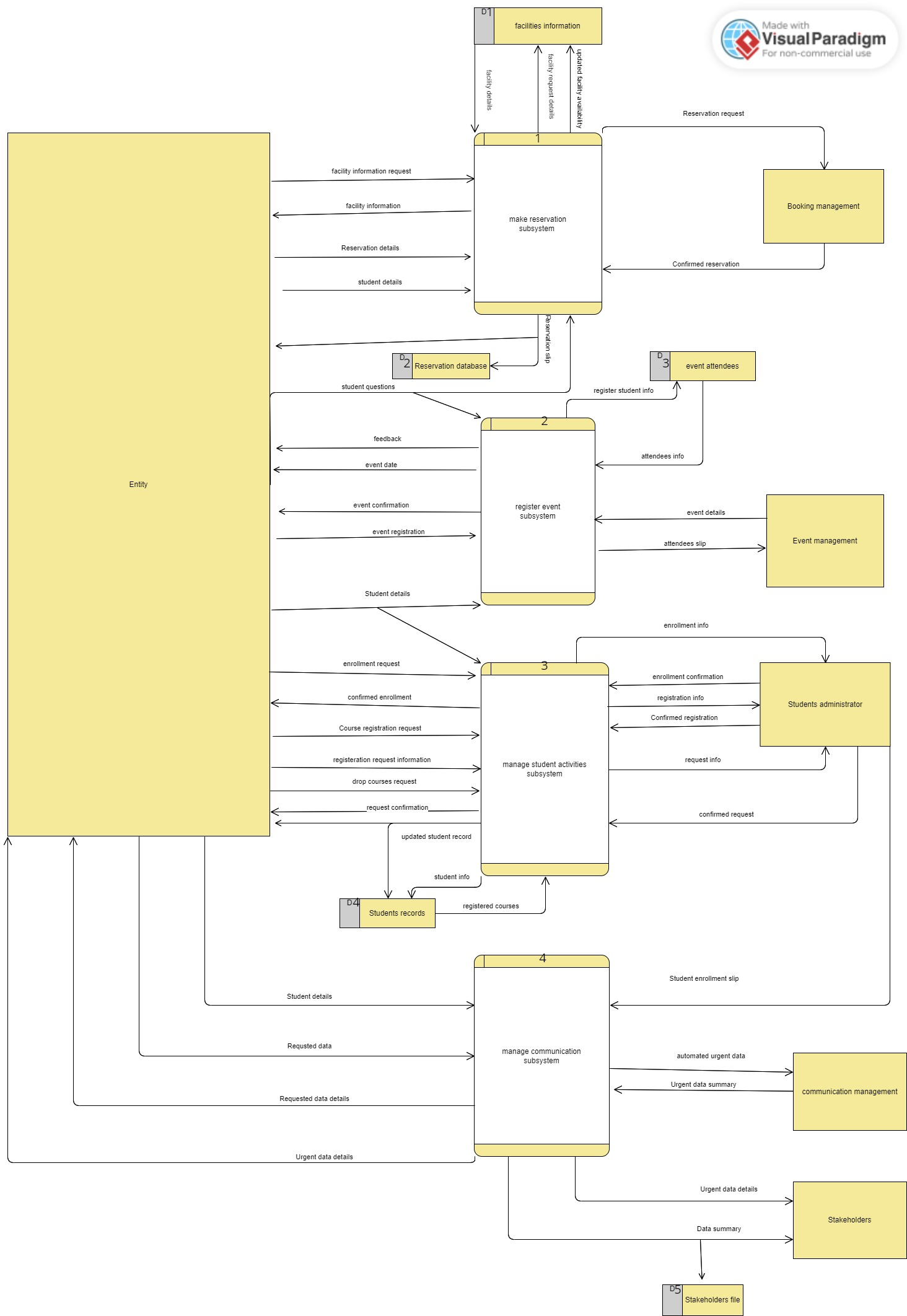
We could not find the same shapes for entities and data stores stated in the slides, so we used these ones.

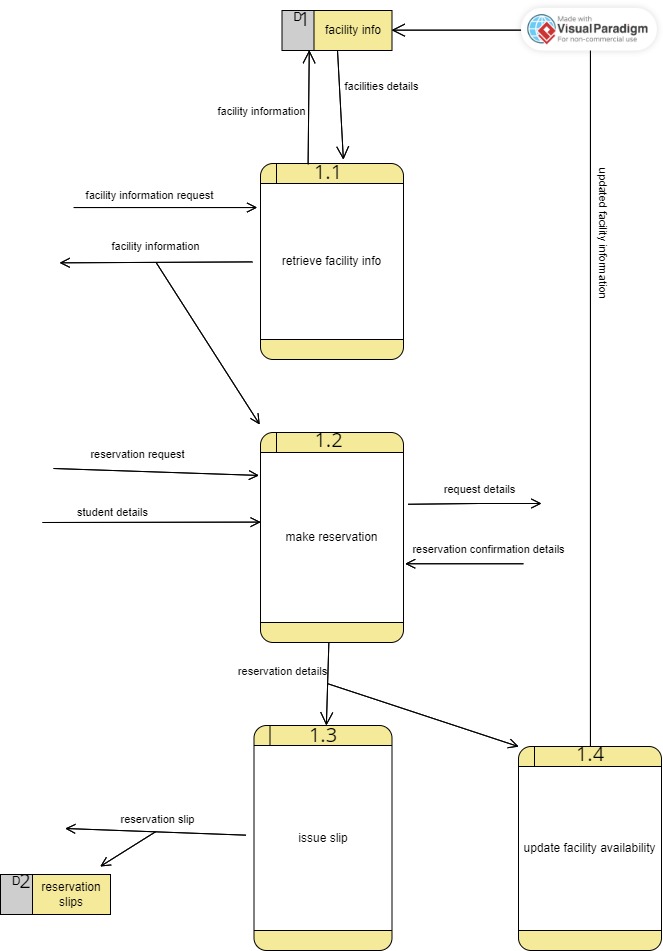
**6.1 Logical DFD (TO-BE):**

**6.1.1 DFD Context diagram**

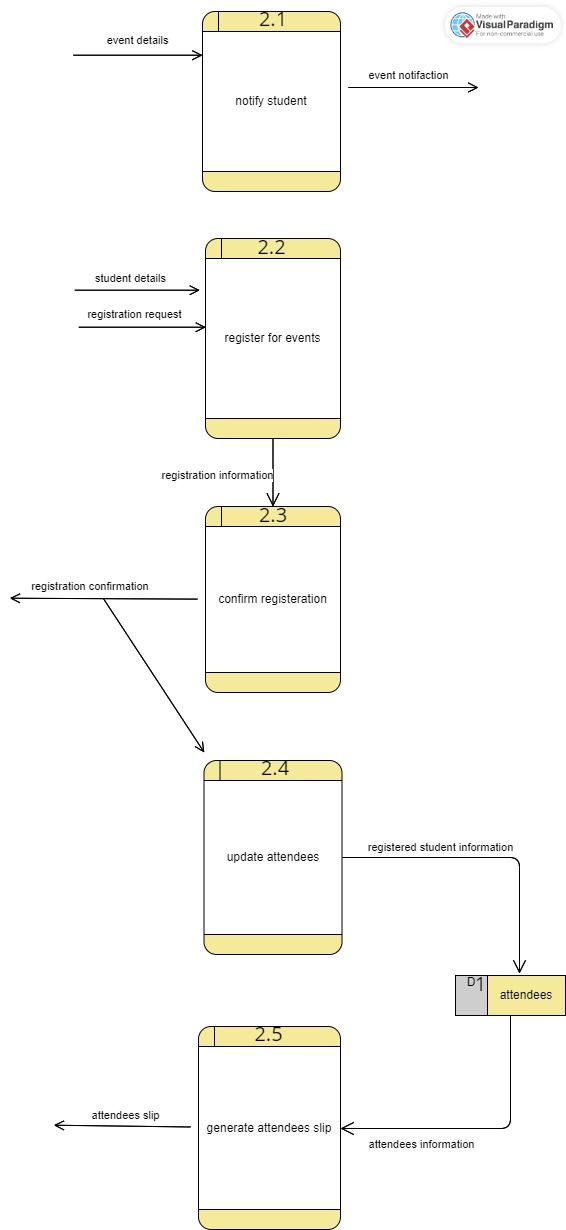
**(context diagram image)**

**6.1.2 DFD Diagram 0**

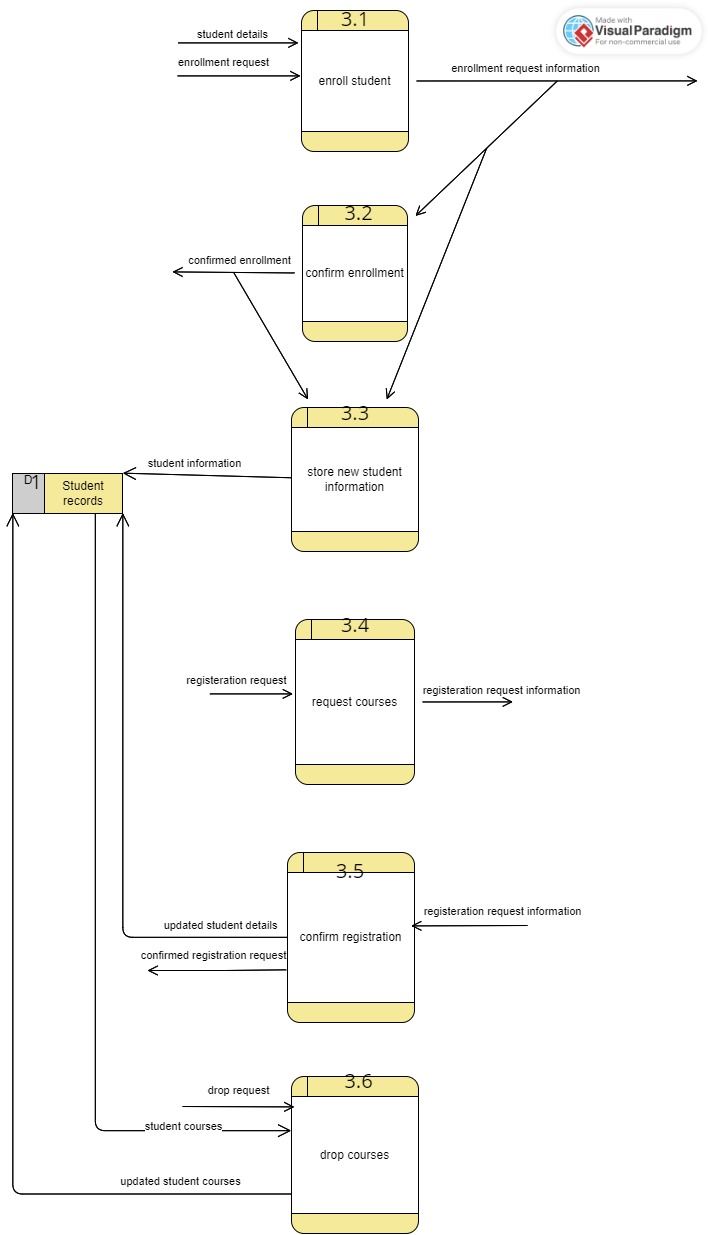


**6.1.3 Make reservation subsystem child diagram** 

**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: 1.1

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:

Online  Manual  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:

Structured English  Decision Table  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:

unavailable

2

Send error message

Receive confirmation details info

invalid

1

5

available

3

4

valid

Valid student details?

6

Create Reservation and request details

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: 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:

Online  Manual  Batch

Process Logic:

BEGIN

ACCEPT Event details.

Send Event notification to student.

END

Refer to Name:

Structured English  Decision Table  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:

invalid

Send error message

2

Receive student details

1

6

valid

Send registration information

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: Subprogram/Function Name:

Online  Manual  Batch

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: 3.1

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:

Online  Manual  Batch

Process Logic:

BEGIN

READ student details

READ enrollment request

SEND enrollment information to student administrator.

END

Refer to Name:

Structured English  Decision Table  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: Subprogram/Function Name:

Online  Manual  Batch

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 store 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:

Online  Manual  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)**

**A screenshot of a computer screen

Description automatically generated**

**7.1.2 Make reservation subsystem Physical child diagram**

**A diagram of a diagram

Description automatically generated**

**7.1.3 register events subsystem physical child diagram**

**A diagram of a diagram

Description automatically generated**

**7.1.4 Manage student activities subsystem physical child diagram**

**A diagram of a computer

Description automatically generated**

**7.1.5 Partitioning: (waiting to put the pictures)**

**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:**

|  |  |
| --- | --- |
| **Activity** | **Attendees list** |
| 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

|  |  |
| --- | --- |
| **Activity** | **Student record database** |
| 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 reserved 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 reserved 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 record in student record database | Student ID, offer letter, passport | 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:**

**7.1.9: System architecture:**