# Independent University, Bangladesh


## Project Draft Report

**Course Title:** Database Management

**Course Code:** CSE 303

**Section:** 04

**Prepared for:** Ms. Sadita Ahmed

**Prepared By:**

Mohammad Arshad Hossain Ratul (1930319)

Tunisha Yanoor Bristy (1930639)

Refat Ahmed Bhuiyan (1930053)

Nahin Islam (1930577)

Tasnia Tabassum Oishee (1910040)

Sadia Islam Semi (1930127)

Department of Computer Science & Engineering

School of Engineering, Technology & Science

Independent University, Bangladesh

# STUDENT ENROLLMENT AND REVENUE ANALYSIS

# Contents

# Table of Figures

# CHAPTER 1

# INTRODUCTION

## A: BACKGROUND OF THE ORGANIZATION:

Independent University, Bangladesh(IUB) is one of the first private universities of Bangladesh. It was established in 1993. Currently around 8,423 students are enrolled in IUB. Moreover, IUB currently has 13,745 alumni and 401 faculty members who are contributing to outstanding research in a variety of fields. IUB conducts its academics through various Schools. Currently, IUB has 5 Schools:

1. School of Entrepreneurship and Business.
2. School of Engineering, Technology and Sciences.
3. School of Environment and Life Sciences.
4. School of Liberal Arts and Social Sciences.
5. School of Pharmacy and Public Health.

Under each School, there are several departments. Currently, IUB has a total of 12 departments. As per the world university rankings of Times Higher Education (THE) IUB stands within 400th universities globally in terms of various aspects of impact analysis. Moreover, IUB provides various types of scholarships to encourage the students in academics. With the help of well-equipped laboratories as well as an enormous library, students and researchers get to invest their knowledge efficiently. [1]

## B: BACKGROUND OF THE PROJECT:

This project is used to analyze the enrollment process of a specific organization.This project focuses on decreasing the manual labor for tasks. Moreover, various kinds of users for an organization like IUB can use the software to view information. Both the enrollment data and revenue information are stored in the database. In addition, detailed data of all the courses for each department is added including credit hour, course ID, currently enrolled students, etc. The data stored can be used to generate various charts and tables in order to view the information required. In the enrollment process we can see the information of course's course name, course ID, course section, maximum capacity, enrolled capacity, class time and days. There is information available such as faculty name, faculty ID, classroom number, school name and others. The project works by taking an input from the user through which it can generate charts/tables that the user wants to view.

## C: OBJECTIVES OF THE PROJECT:

The objective of our project is to perform some of the significant tasks that need to be done repeatedly. With the help of this project, the time consuming tasks that are done manually are done within a short span of time. The basic functions of this project are:

- Store the enrollment information of students of each Semester for each School and Department.
- Analyze the revenue of each School for selected Semesters as well to view the percentage change in revenue for each School from a specific semester to another.
- Generate the classroom requirement for each day for a specific number of slots.
- Analyze the resources of IUB by observing the amount of unused resources.

With the help of this software, the human interference in these vital tasks are reduced and hence, the results are more precise as it has less chance of errors. Moreover, by automating the tasks, the time can be utilized elsewhere. In addition, data can be accessed easily.

Although building a software like this might seem expensive but on a larger scale considering doing the same tasks manually, it is more efficient and less costly to go for automated processes.

## D: SCOPE OF THE PROJECT:

This project is giving out solutions to enrollment issues and revenue changes. This project is helping students, faculty, department managers and others to use this web app for data:

- storing
- editing
- adding
- updating
- viewing

We designed possible users for the web-based system for the users and imagined how they would use it, as well as the information and data they would require.

We will create customized user interfaces and login capabilities for all participants who will be users of this system, because problems might develop at any stage in any business process.

We use this system for data storage, so obtaining relevant files, tabular data, formats, and reports is really simple, and we can interact with the data in real time.

We also build user interfaces that allow all users to quickly access data and generate and download reports. [2]

# CHAPTER 2

# REQUIREMENT ANALYSIS

## A: RICH PICTURE OF EXISTING BUSINESS SYSTEM

Rich picture AS-IS: A rich picture is a drawing of a situation that illustrates the main elements and relationships that need to be considered in trying to intervene in order to create some improvement. It consists of pictures, text, symbols and icons, which are all used to illustrate the situation graphically. It is called a rich picture because it illustrates the richness and complexity of a situation. [2] [3]

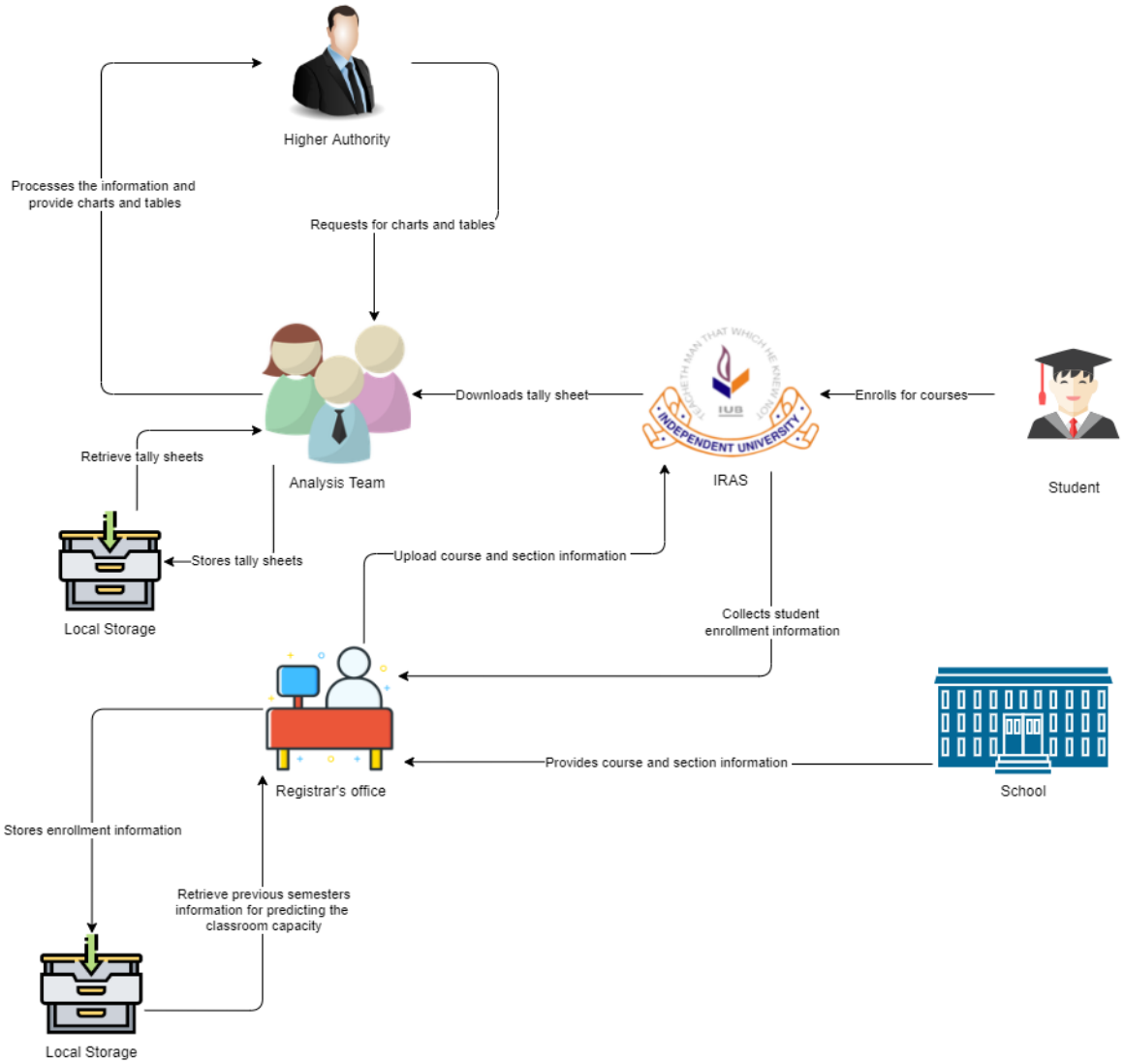



*Figure 1: Rich Picture AS-IS*

# B: EXISTING SIX SYSTEM ELEMENTS

SIX SEGMENT: The Six Elements Analysis provides a detailed description of the role of each element in each process. It is clear from the table below that Human entities dominate all key functions of this system, Non-Comp Hardware denotes the hardware needed for the system computing hardwares will not be included in this, Computing Hardware denotes the software related hardwares like computer or printer, Software contains which softwares will be used for this process, database dominate the server and storage. [4]

| Process | System Roles | | | | | |
|---|---|---|---|---|---|---|
| | **Human** | **Non-Computing Hardware** | **Computing Hardware** | **Software** | **Database** | **Network & communication** |
| **Prepare revenue table and chart for each School** | **Higher authorities:**<br>1.Request the analysis team for the revenue table and chart for particular semesters via email.<br>2. Receive tables and charts from the analysis team via email.<br><br>**Analysis team:**<br>1.Receive email from the higher authority to send tables and charts.<br>2.Check the local storage to fetch data.<br>3.If data is not found, then use iRAS to download the data.<br>4.. Login to Iras.<br>5. Download the tally sheet from iras. | **Paper:**<br>1.Used to print the chart and table. | **Computer:**<br>1. Analysis team Used to login to iras.<br>2. Used to Download the tally sheet from iras.<br>**Printer :**<br>1. To print out the chart and data from the table | **Iras:**<br>1. Analysis team use iras to download the tables and charts in the excel file<br>**Microsoft Excel:**<br>1.Excel Sheets are used by Higher authorities to read the table and chart.<br><br>Gmail:<br>1. Used to send and receive emails. | **Local Storage:**<br>1. Analysis team use a local storage to store the tally sheets. | **Internet:**<br>1. The Internet is used to Communicate with IRAS to browse the software/ webpage where the analysis team can download required tally sheets.<br>2. To access email. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 6.Process table and chart from the tally sheets.<br>7. Sends the charts and tables to the higher authorities via email.<br>8.Store tally sheets to the local storage. | | | | | |
| **Comparative analysis of the number of sections in each school** | **Higher Authority:**<br>1. Send an email with a set range to the analysis team to prepare an analysis report for each school where the number of students enrolled is less than the given range.<br>2. Receives the report.<br>Analysis team:<br>1. Receives an email from the higher authorities.<br>2. Analyzes the given task.<br>3. Retrieves the needed tally sheet from its local storage.<br>4. If not stored previously then downloads the tally sheet from Iras.<br>3. Prepare the report in excel sheet.<br>4. Stores the<br>tally sheet in local storage. | **Paper:**<br>1. Used to print the hardcopy if needed. | **Computer:**<br>1. Used to login iRAS.<br>2. For preparing the comparative analysis sheet.<br>3. Download the analysis sheet from iRAS.<br>**Printer:**<br>1. To print out the analysis sheet. | **iRAS:**<br>1. The Analysis team uses iRAS to get the information.<br>2.Register office uses iRAS to download the analysis sheet.<br>**Gmail:**<br>1. Used to send and receive emails. | **Local Storage:**<br>1. Analysis team uses a local storage to store the tally sheets.<br>2. Analysis team uses a local storage to fetch the previously stored tally sheets. | **Internet:**<br>1. Used to send and receive emails.<br>2. Used to access Iras. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 5. Send the report to the higher authorities. | | | | | |
| **Provide course and section informations to iRAS** | **School:**<br>1. Emails course and section information to the registrar's office.<br><br>**Registrar's office:**<br>1. Receives chart information about course and section.<br>2.Retrieve previous semesters<br>information for predicting the<br>classroom capacity.<br>3. Uploads predicted course and section capacity,<br>to Iras. | **Paper:**<br>1. Used for keeping hardcopy of the information. | **Computer:**<br>1.For preparing the information sheet.<br>2. Sending emails. | **iRAS:**<br>1.Used by the registrar's office to upload data..<br><br>**Microsoft Excel:**<br>1. Used to make the required information list by the school.<br>2. Used to make the predicted list by the register's office.<br><br>Gmail:<br>1. To email the course and section information to the registrar's office. | **Local Database:**<br>1. Used by registrar's office to store<br>the hardcopy of data. | **Internet:**<br>1. To access iRAS. |
| **Information of student enrollment** | **Student:**<br>1.Enrolls for courses in iras.<br>iRAS: | **Paper:**<br>1.Print the information if needed. | **Printer:**<br>1.Printer is used to print the information | **iRAS:**<br>1.iRAS is used to access enrollment. | **Local Storage:**<br>1.Information has been sent to the local storage | **Internet:**<br>1.Internet is used to access iRAS for enrollment by students. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 1.Collects information regarding enrollment.<br>2.Sends information to the Registrar's Office.<br>**Registrar's Office:**<br>1.Receiving the enrollment information.<br>2. Sending to local storage for storing the information.<br>Local Storage:<br>1.Receiving the information and storing the enrollment information. | | into hardcopy.<br>**Computer:**<br>1.Computer is used to enroll for the courses through iRAS.<br>2.Sending information from iRAS to the registrar's office. | **Gmail:**<br>1.It is used to send enrollment information to the registrar's office | from the registrar's office to store the information. | 2.Internet is used to send enrollment information to the registrar's office by iRAS. |
| **Prepare classroom requirement summary** | **Higher Authority:**<br>1.Emails the analysis team to prepare a classroom requirement summary as per the class size desired.<br>2. Receives the information.<br>**Analysis Team:**<br>1.Receives email to prepare classroom requirement summary via email.<br>2. Analyzes the number of slots per day. | **Paper:**<br>1.Used to store the hardcopy of classroom requirement summary. | **Computer:**<br>1.To receive and send emails<br>2.To make the charts required. | **Gmail:**<br>1. To receive and send email<br>**Microsoft Excel:**<br>1.To prepare the classroom requirement summary on a chart. | **Local Database:**<br>1.Used to store the summary for future references. | **Internet:**<br>1.Used to send email and receive email. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 3. Prepares classroom requirement summary as per the class size provided as a Chart.<br>4.Email the chart to the higher authority. | | | | | |
| **Analysis of percentage of unused resources** | **Analysis Team:**<br>1. Receives email from the higher authorities.<br>2. Look for the needed tally sheets.<br>3. If the tally sheet is stored in the local storage then fetch the data. 4. If the tally sheet is not present then login to iras and download the required tally sheet.<br>5. Calculate the percentage of unused resources.<br>6. Produce a report.<br>7. Send the report to the higher authorities.<br><br>**Higher authorities:**<br>1. Asks the analysis team to make a report for unused of each school via email.<br>2. Asks the analysis team to make a report for the total | **Paper:**<br>1. To store a hard copy of the report. | **Computer:**<br>1. To make the report. | **Microsoft Excel:**<br>1. To generate the report.<br><br>**Gmail:**<br>1. To email the Analysis team to ask for the report.<br>2. To email the report to the higher authorities. | **Local Database:**<br>1. Used by the analysis team to store the report of unused resources. | **Internet:**<br>1. To access iras.<br>2. To mail. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | unused percentage for the university overall based on average enrollment per section versus average capacity per room. | | | | | |

# C: BUSINESS PROCESS DIAGRAM (AS-IS)

BPMN:Business Process Model and Notation (BPMN) is a graphical representation for specifying a business process in a business process model.BPMN has been designed to provide a standard notation readily understandable by all business stakeholders, typically including business analysts, technical developers and business managers.

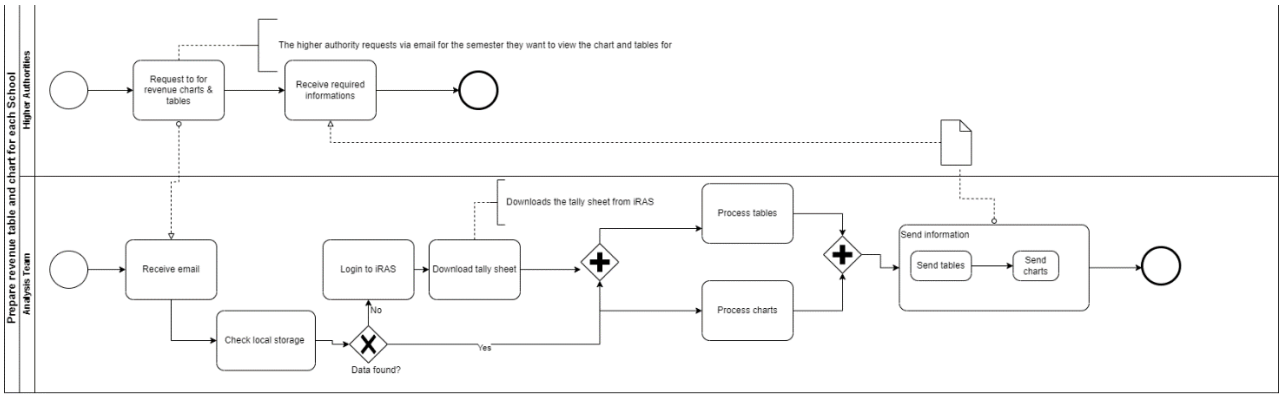



*Figure 2:Prepare revenue table and chart for each School (AS-IS)*

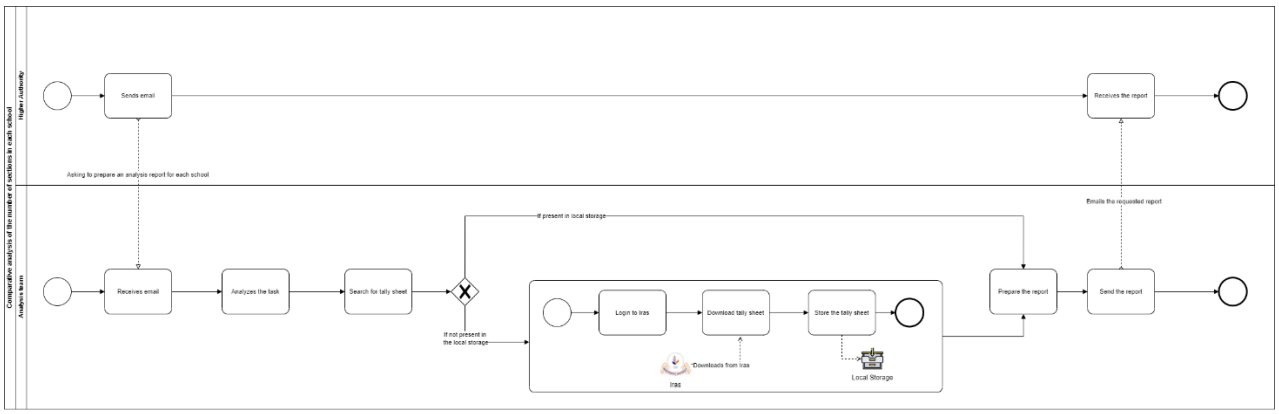



*Figure 3: Comparative analysis of the number of sections in each school (AS-IS)*

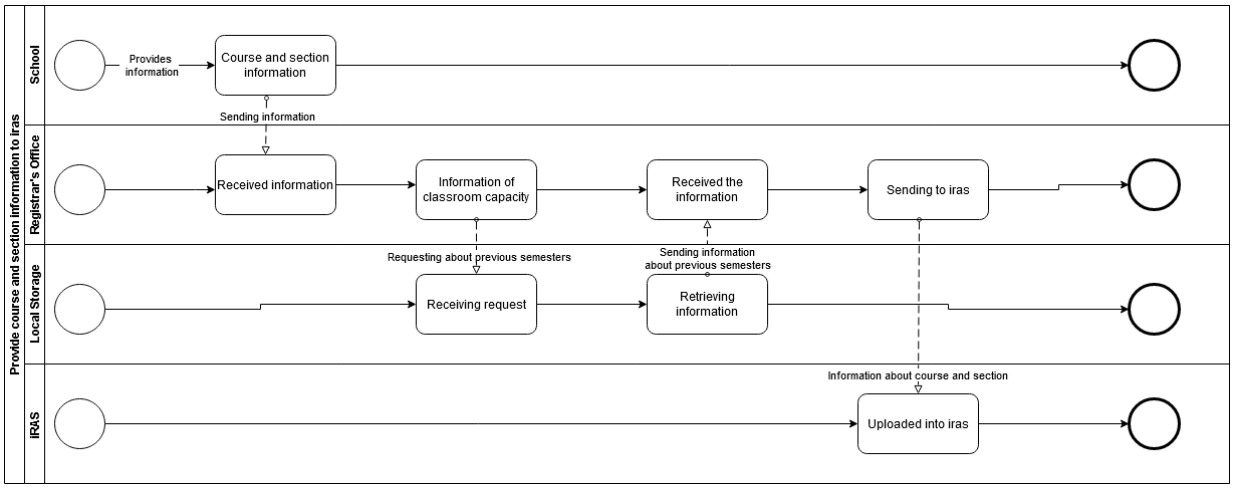



*Figure 4: Provide course and section informations to iRAS (AS-IS)*

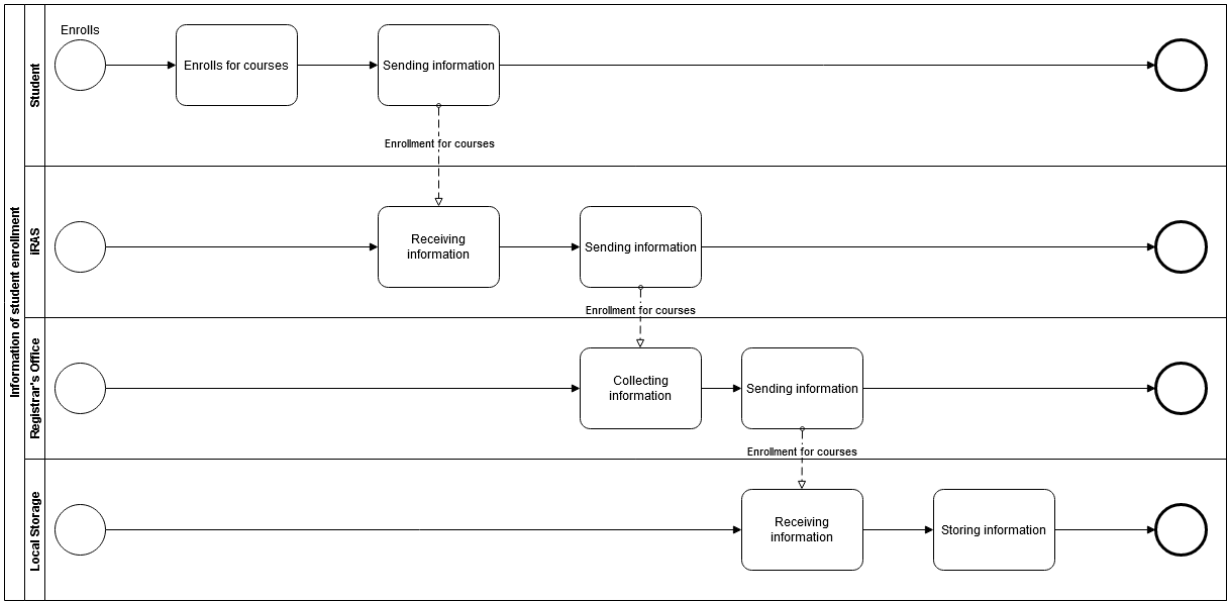



*Figure 5: Information of student enrollment (AS-IS)*

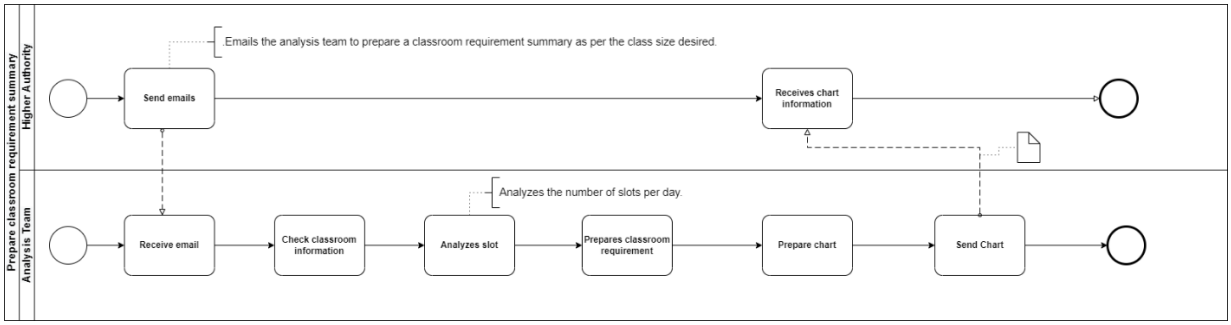



*Figure 6: Prepare classroom requirement summary (AS-IS)*

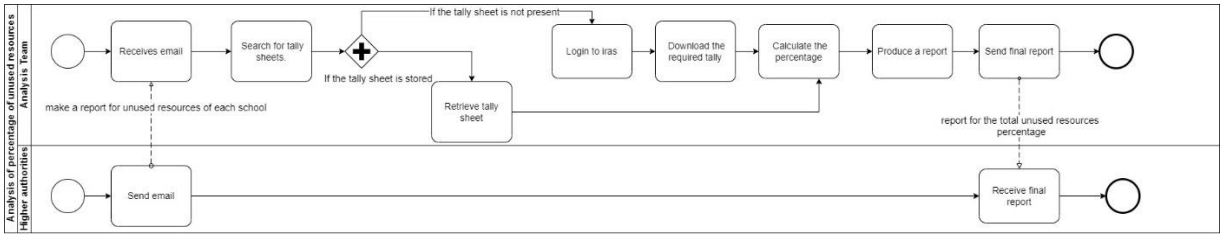



*Figure 7: Analysis of percentage of unused resources (AS-IS)*

# D: PROBLEM ANALYSIS OF THE EXISTING BUSINESS SYSTEM

**Problem analysis:** Based on the existing systems' Six Elements Analysis, the shortcomings in each process were identified. There is a repeating pattern in the far-right column of this table. It appears that the facilitation of a private online platform will improve the system in many ways.

| Process Name | Stakeholders | Concerns (Problems) | Analysis (Reason of the Problems) | Proposed Solution |
|---|---|---|---|---|
| Prepare revenue table and chart for each School | 1.Higher Authority<br>2.Analysis Team | The process is time consuming | The communication is done via email as the higher authorities have to send an email first if they want to view the information. After they have sent the email, then the analysis team prepares the revenue table which takes a lot of time. | The system will automatically generate charts and tables so that the delay in between the tasks can be avoided. This way, the higher authority can view the information within a few seconds. |
| Comparative analysis of the number of sections in each school | 1.Higher Authority<br>2.Analysis Team | Delay in receiving the analysis | As the communication is done via email, the process is slower. In addition, the analysis team has to go through the local storage first to retrieve the required tally sheets and if they are unable to find it then they have to login to iRAS to download the tally sheet.All these tasks are extremely time consuming and there can be errors in between as well. | The tally sheets can be stored in the system and then whenever an user wants to view the analysis, the system would automatically calculate it.Hence, delays are avoided and it is more precise since there are no manual searches to be done |
| Prepare classroom requirement summary | 1.Higher Authority<br>2.Analysis Team | The process is time consuming | The communication is done via email as the higher authorities have to send an email first if they want to view classroom requirement summary. After they have sent the email, then the analysis team analyzes the | The system will automatically generate the required charts from user requests so that the delay in between the tasks can be avoided. This way, the higher authority can view |

| | | | | |
|---|---|---|---|---|
| | | | number of slots per day and make a chart about classroom requirement | the information within a few seconds from the website. |
| Analysis of percentage of unused resources | 1.Higher Authority<br>2.Analysis Team | Delay in the analysis | The communication is done via email, hence, the process becomes slower. In addition to this, the analysis team has to go through the local storage first to find the tally sheets and if they are unable to find, then they have to login to iRAS in order to download the tally sheet and then they have to make the analysis of the unused percentage of resources. After all these, they have to send the analysis to the higher authorities. This makes room for a lot of errors and also delays the tasks in between. | The system can be used to store the tally sheets and then whenever an user requests to view the analysis, the system can automatically generate charts from the tally sheets that are stored. This eliminates errors as well as delay. |

# E: RICH PICTURE OF PROPOSED BUSINESS SYSTEM

Here archive, database, department and an online platform called SEARI will be visible SEARI will contain its own database that host the data of all the courses, faculties, as well as updated tables every semester to keep track of which courses have been assigned to which faculties in a given semester. We are making the new system to track updates of the enrollment system. Briefly, we can see that the SEARI relational database (a non-human) plays a significant role in the student enrollment system. Also, this entity holds the greatest number of interconnections between all other processes. We will use different user interfaces designed for specific user needs based on the concerns and problems we found in the problem analysis. The Higher authority, School, Department, Register office , Student, all these stakeholders mentioned will have access to view the system.

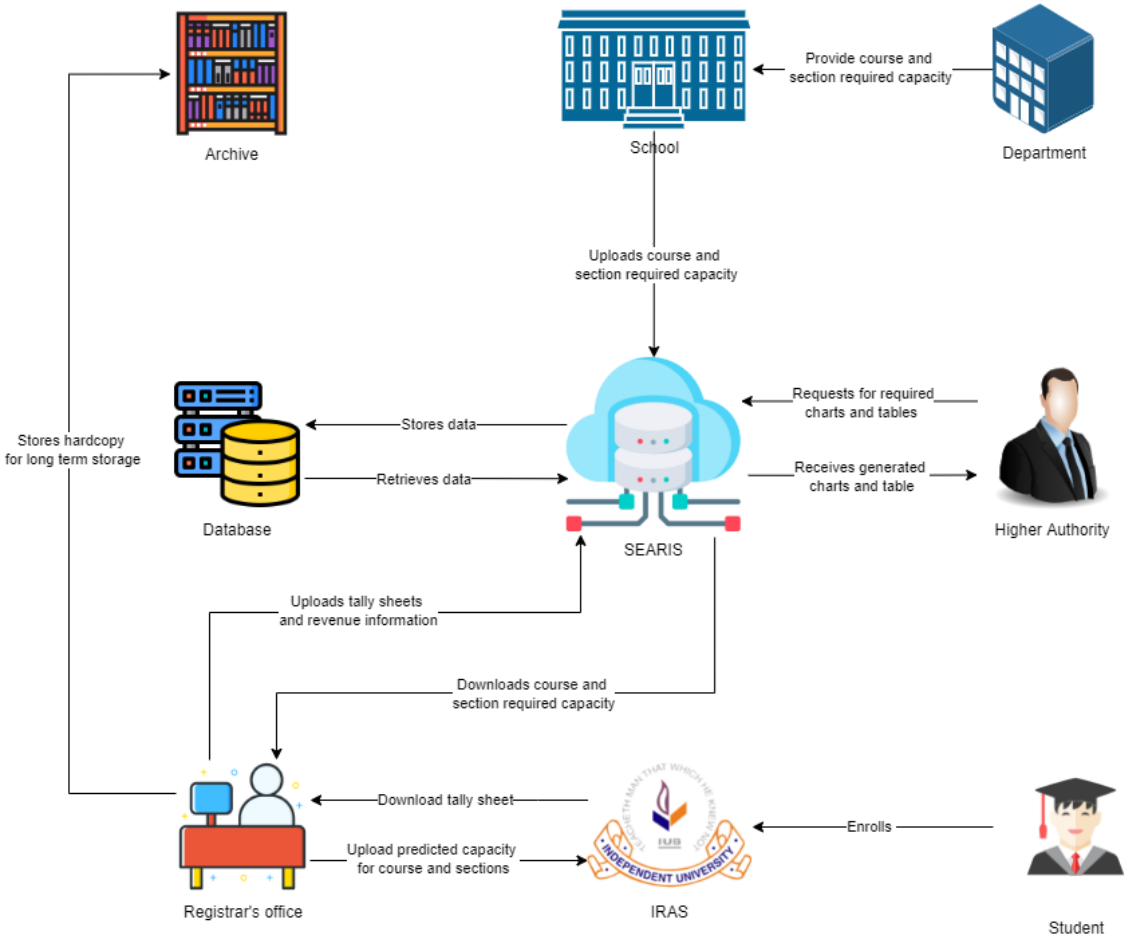



*Figure 8: Rich picture of proposed system*

# F: PROPOSED SIX SYSTEM ELEMENTS

BPMN: After understanding the role of each element in each process, the Business process model and notation provides dictation of the exact sequence of steps so that we can fulfill each process. Every module of this diagram will serve an upgraded system.

| Process | System Roles | | | | | |
|---|---|---|---|---|---|---|
| | **Human** | **Non-Computing Hardware** | **Computing Hardware** | **Software** | **Database** | **Network & communication** |
| **Prepare revenue table and chart for each school** | **Higher Authorities:**<br>1.Login to SEARIS<br>2.Click on revenue of semester<br>3.Enter semester by selecting a session and year from the drop-down list<br>4.Click on submit<br>5.Click on print if needed<br>6.Logout of SEARIS | **Paper:** Used to print the chart and table. | **Computer:**<br>1.Used to login to SEARIS.<br>2.Used to generate charts and Table.<br><br>Printer: Used to print the chart and table. | **SEARIS:**<br>1.Higher authorities use "SEARIS'' to view the revenue for a particular semester in a chart or table. | **SEARIS database:** Used to fetch the required data to create a table and chart | **Internet:**<br>Used to access SEARIS. |
| **Analysis of the number of sections** | **Higher Authorities:**<br>1. Login to SEARIS<br>2. Click on number of sections<br>3. Select range<br>4. Enter semester by selecting a session and year from the drop-down list<br>5. Click on submit | **Paper:**<br>1. Used to print hardcopy. | **Computer:**<br>1. Used to access "SEARIS"<br>2. Used to see the analysis. | **SEARIS:**<br>1. Used to generate the analysis charts.<br>2. See the breakdown tables. | **SEARIS database:**<br>1. Used to fetch the required data to create the analysis chart and detailed breakdown table. | **Internet:**<br>1. Used to access "SEARIS". |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 6. Click on view total number of sections<br>7. There will be an option ”Show detailed view”<br>which will show<br>detailed breakdown of the analysis.<br>8. If the user wants to change the range or the chosen semesters then the user needs to click the “back” button which will take the user to the interface where he needs to enter the range and semesters.<br>9. There will be an option for printing the analysis and also the breakdown, if the user wants to print.<br>10. Logout of SEARIS | | | | | |
| **Provide course and section information to SEARIS** | **School:**<br>1. Emails course and section information to the registrar’s office.<br>**Registrar's office:**<br>1. Receives chart information about course and section.<br>2.Retrieve previous semesters | **Paper:**<br>1. Used for keeping hardcopy of the information<br>. | **Computer:**<br>1.For preparing the information sheet.<br>2. Sending emails. | **iRAS:**<br>1.Used by the registrar’s office to download enrollment information.<br>**Microsoft Excel:** | | **Internet:**<br>1. To access iRAS. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | information for predicting the<br>classroom capacity.<br>3. Uploads predicted course and section capacity,<br>to Iras. | | | 1. Used to make the required information list by the school.<br>2. Used to make the predicted list by the register's office.<br><br>**Gmail:**<br>1. To email the course and section information to the registrar's office. | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Information of student enrollment** | **Student:**<br>1.Enrolls for courses in iras.<br>iRAS:<br>1.Collects information regarding enrollment.<br>2.Sends information to<br>the Registrar's Office.<br>**Registrar's Office:**<br>1.Receiving the enrollment information.<br>2. Sending to local storage for storing the information.<br>Local Storage:<br>1.Receiving the information and the store the enrollment information. | **Paper:**<br>1.Used to print the information . | **Printer:**<br>1.Printer is used to print the information into hardcopy.<br>**Computer:**<br>1.Computer is used to enroll for the courses through iRAS.<br>2.Sending information from iRAS to the registrar's office. | **iRAS:**<br>1.iRAS is used to access enrollment.<br>**Gmail:**<br>1.It is used to send enrollment information to the registrar's office. | **Local Storage:**<br>1.Information has been sent to the local storage from the registrar's office to store the information. | **Internet:**<br>1.Internet is used to access iRAS for enrollment by students.<br>2.Internet is used to send enrollment information to the registrar's office by iRAS. |
| **Prepare classroom requirement summary** | **Higher Authorities:**<br>1. Login to SEARIS<br>2.Click on prepare classroom requirement summary<br>3.Enter semester by selecting a session and year from the drop-down list<br>4.Enter the class size, total slots per | **Paper:** Used to print the summary if needed. | **Computer:**<br>1.Used to login to SEARIS.<br>2.Used to generate charts.<br>**Printer:**<br>1.Used to print the summary if required. | **SEARIS:**<br>1. Used to generate classroom requirement summary. | **SEARIS database:**<br>1.Contains the past data and with the help of that, a chart for classroom requirement summary is generated. | **Internet:**<br>Used to access SEARIS. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | day and sections offered<br>5.Click on submit<br>6.Click on view classroom requirement summary<br>7.Click on print if needed<br>8.Logout of SEARIS | | | | **Archive:** Contains data from the past. | |
| **Analysis of percentage of unused resources for each school** | **Higher Authority:**<br>1. Login to SEARIS<br>2.Click on view unused resources button<br>3.Enter semester by selecting a session and year from the drop-down list<br>4.Click on submit<br>5.A pie chart appears<br>6.Click on print if needed.<br>7.Logout of SEARIS. | **Paper:**<br>1.Used to print the analysis if required. | **Computer:**<br>1.Used to login to SEARIS and view the analysis.<br>**Printer:**<br>1.Used to print the analysis if needed. | **SEARIS:**<br>1.Used to usegenerate tables by using the data stored in the database. | **SEARIS database:**<br>1.Contains all the past data which is used to generate tables. | **Internet:**<br>1.Used to access SEARIS. |
| **Comparative analysis of number sections offered in SETS** | **Higher Authorities:**<br>1.Login to SEARIS<br>2.Click on view number of sections offered<br>3.Enter semester by selecting a session and year from the drop-down list | **Paper:**<br>1.The paper is used to print the charts. | **Computer:**<br>1.Computer is used to access the SEARIS and see the needed information.<br>**Printer:** | **SEARIS:**<br>1.SEARIS is used to access to login to the software to see the needed results. | **SEARIS database:**<br>1. Used to fetch the required data to create the analysis chart. | **Internet:**<br>1.Internet is used to access SEARIS. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 4.Click on submit<br>5.A chart appears<br>6.Click on print if needed<br>7.Logout of SEARIS | | 1.Printer is used to print the needed results. | | | |
| **Upload tally sheet and revenue information** | **Registrar's Office:**<br>1.Login to SEARIS<br>2.Click on upload tally sheet and revenue information<br>3.Enter all the information of tally sheet and revenue<br>4.Click on submit<br>5.Click on view tally sheet and revenue information<br>6.Click on print if needed<br>7.Logout of SEARIS | **N/A** | **Computer:**<br>1.Used to login to SEARIS and iRAS to download the tally sheet and to upload the tally sheet and revenue information files. | **iRAS:**<br>1.Used to download the tally sheet.<br>**SEARIS:**<br>1.Used to upload the tally sheet and revenue information files. | **SEARIS database:**<br>1.Used to store the tally sheet and revenue information. | **Internet:**<br>1. Used to access SEARIS and iRAS. |

# G: BUSINESS PROCESS DIAGRAM (TO-BE)

SIX ELEMENTS ANALYSIS: The six elements analysis of the proposed system is a continuation of an analysis process where each analysis is based on the one that comes before it. Based on the rich picture, the role of each element in the new system is further understood in the table below.

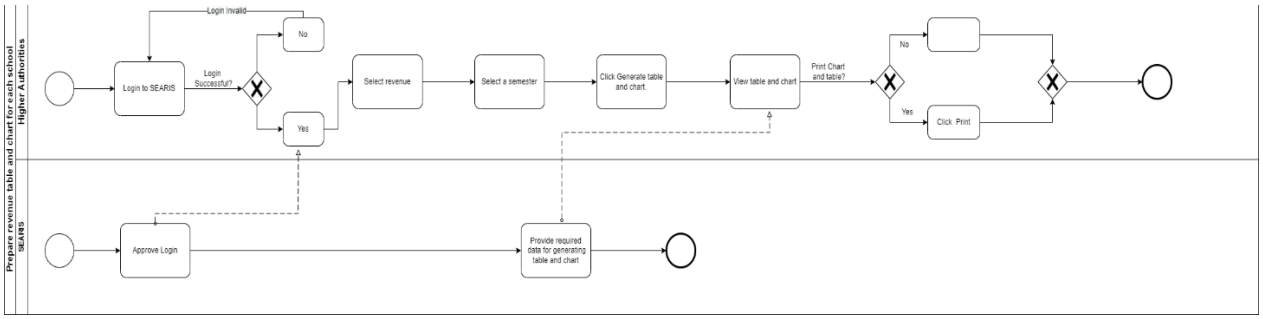



*Figure 9: Prepare revenue table and chart for each school (TO-BE)*

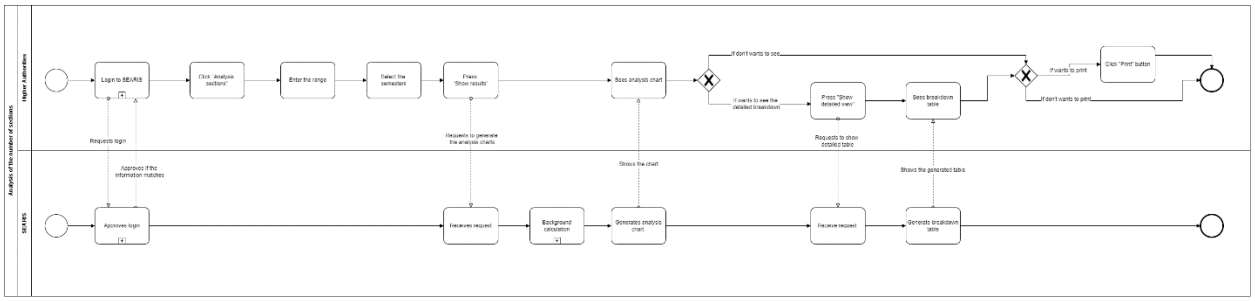


*Figure 10: Analysis of number of section (TO-BE)*

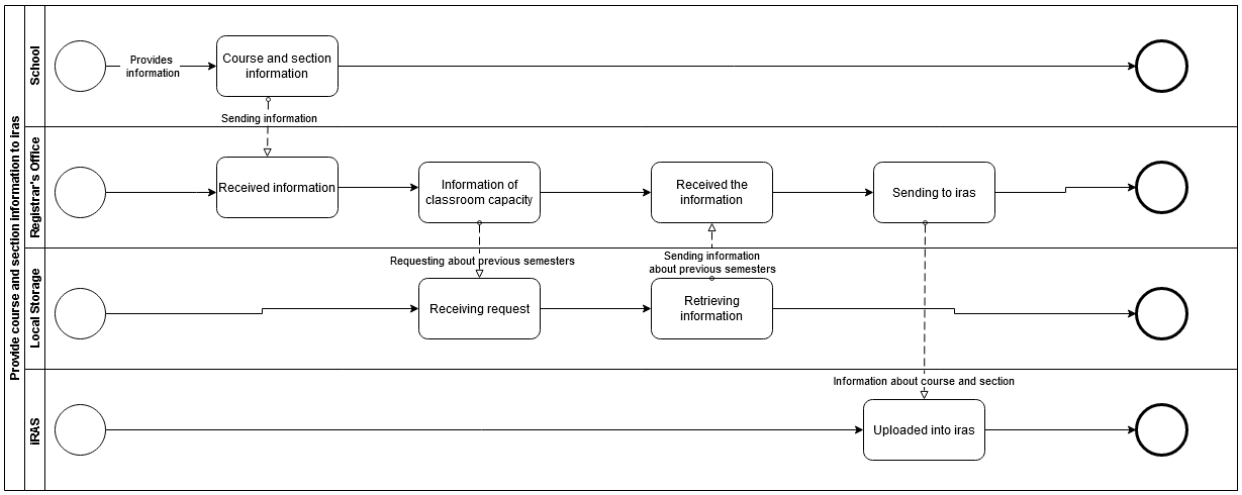



*Figure 11: Provide course and section information to SEARIS (TO-BE)*

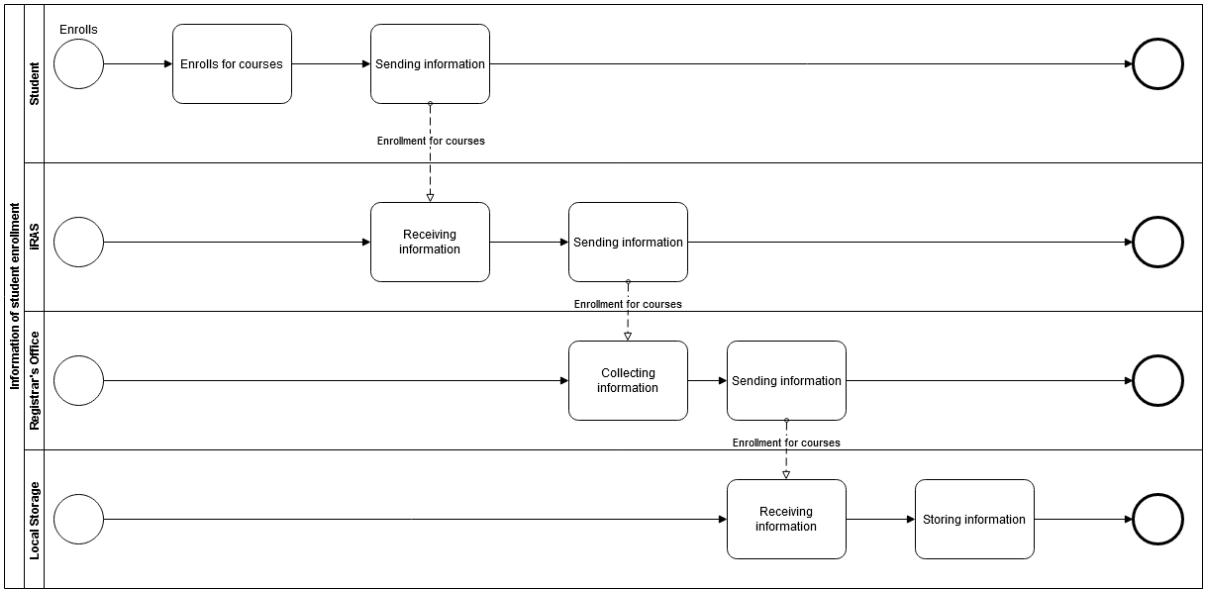



*Figure 12: Information of student enrollment (TO-BE)*

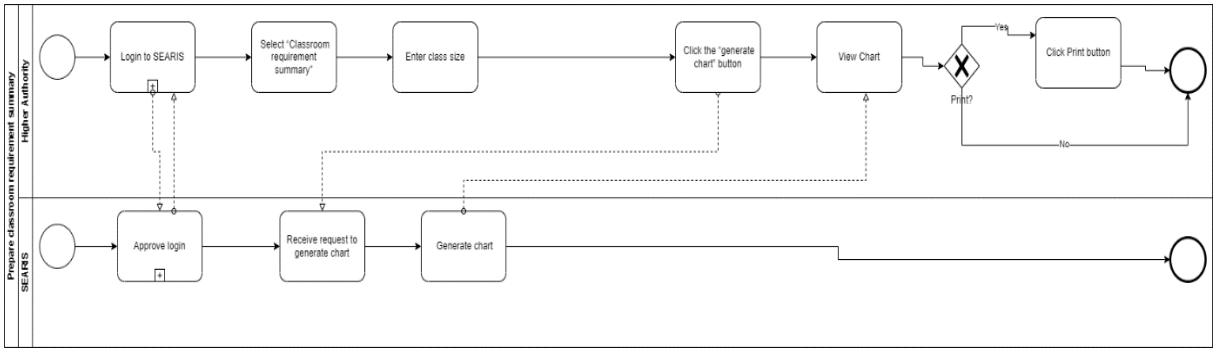



*Figure 13: Prepare classroom requirement summary (TO-BE)*

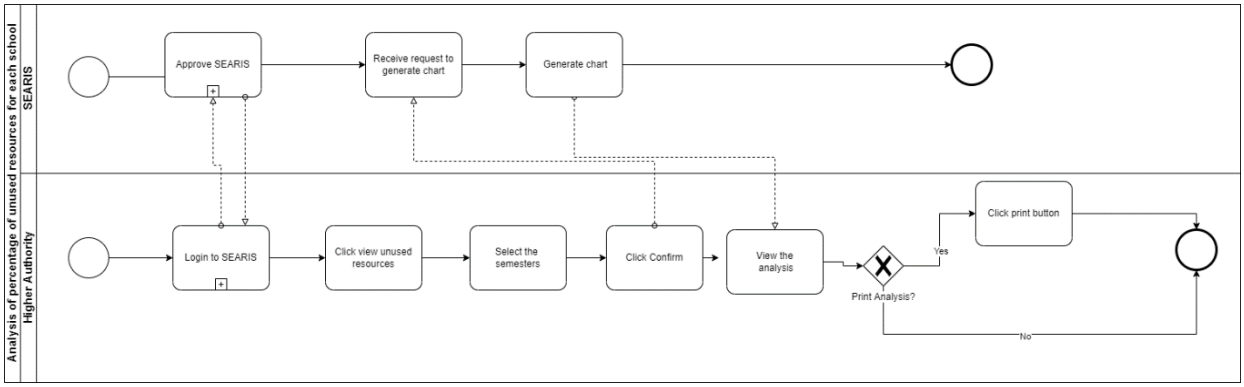



*Figure 14: Analysis of percentage of unused resources for each school (TO-BE)*

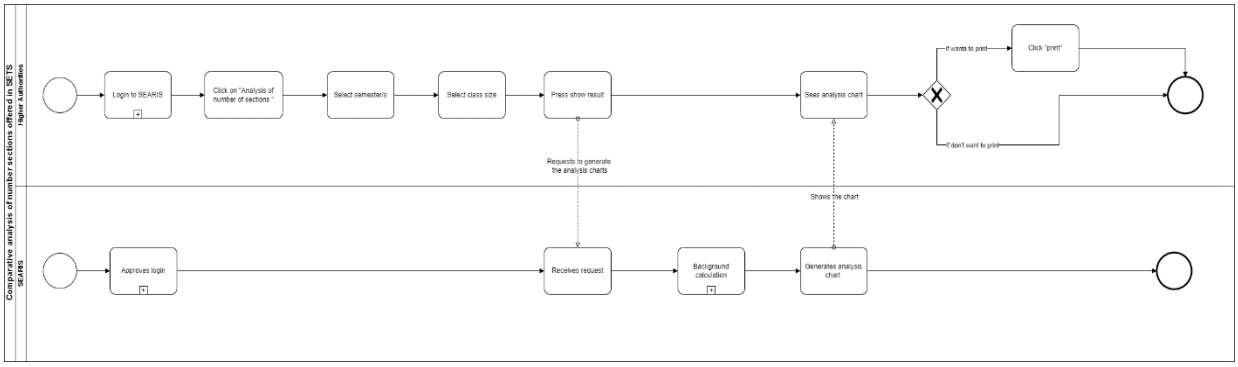



*Figure 15: Comparative analysis of number sections offered in SETS (TO-BE)*

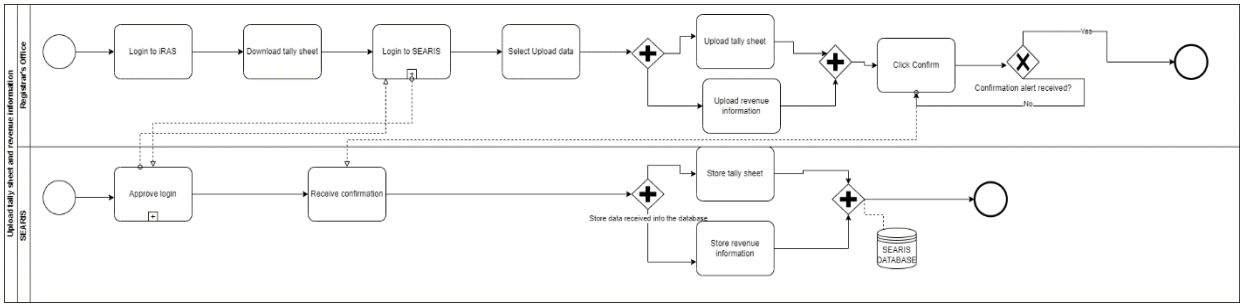



*Figure 16: Upload tally sheet and revenue information (TO-BE)*

# CHAPTER 3

# LOGICAL SYSTEM DESIGN

## A: BUSINESS RULES

1.The academics of IUB consists of School and Department.

2.Each School has a unique identifier which is school_title.

3.Each School consists of several Departments. A department must belong to a School and a School must have at least one or more departments.

4.Departments have dept_id to uniquely identify each department. Departments also have dept_name.

5.Departments offer courses. A department must offer one or many courses. A course must belong to a department.

6.Courses have course_id, course_name, dept_id, cooffered_c, credit_hour, room_id.

7.One course must have at least one or more sections.

8.A section must be taught by one Faculty. A faculty may teach one or more sections.

9.A course can be co-offered with one or more courses.

10.Faculty has faculty_id, faculty_name, course_id.

11.Courses are offered every semester. Semesters consist of a session and year.

12.Sections have section_no, course_id, semester that consist of session and year, room_id,faculty_id, enrolled_capacity, capacity, class_time which has the starting and ending time of a class and days.

13.A faculty can hand over a course of a specific section to another faculty for a particular semester.

14.Classrooms are assigned to a specific section based on the enrollment capacity.

15.Classrooms have room_id and room_capacity.

16.Classrooms are used in slots. For IUB, it is 6 slots or 7 slots for each day.

17.A section is assigned to only one slot at a time.

18.Each course must have at least one slot per week. For some courses, it can be two slots per week.

# B: ENTITY RELATION DIAGRAM (ERD)

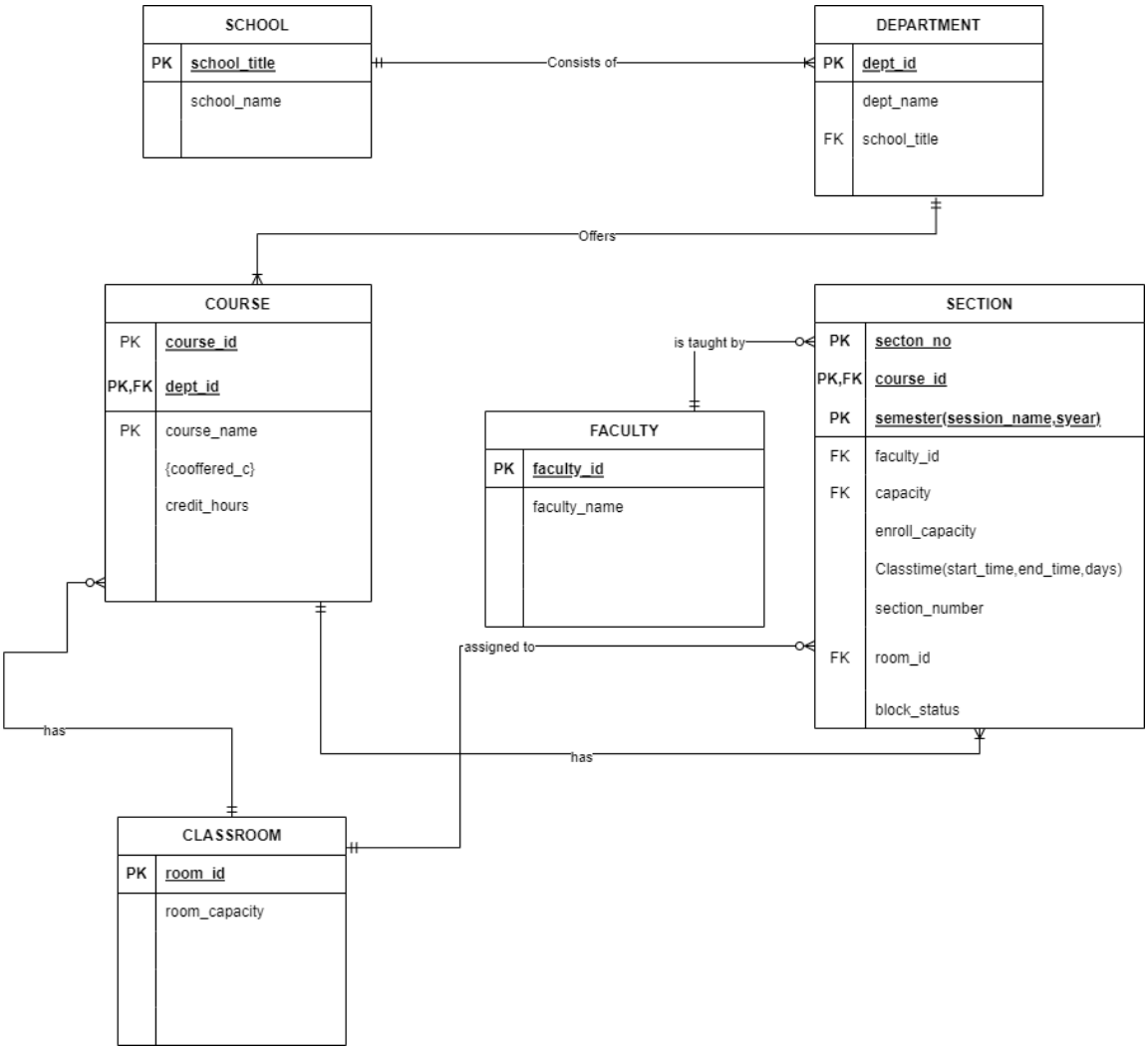



*Figure 17: Entity Relation Diagram*

# C: RELATIONAL SCHEMA

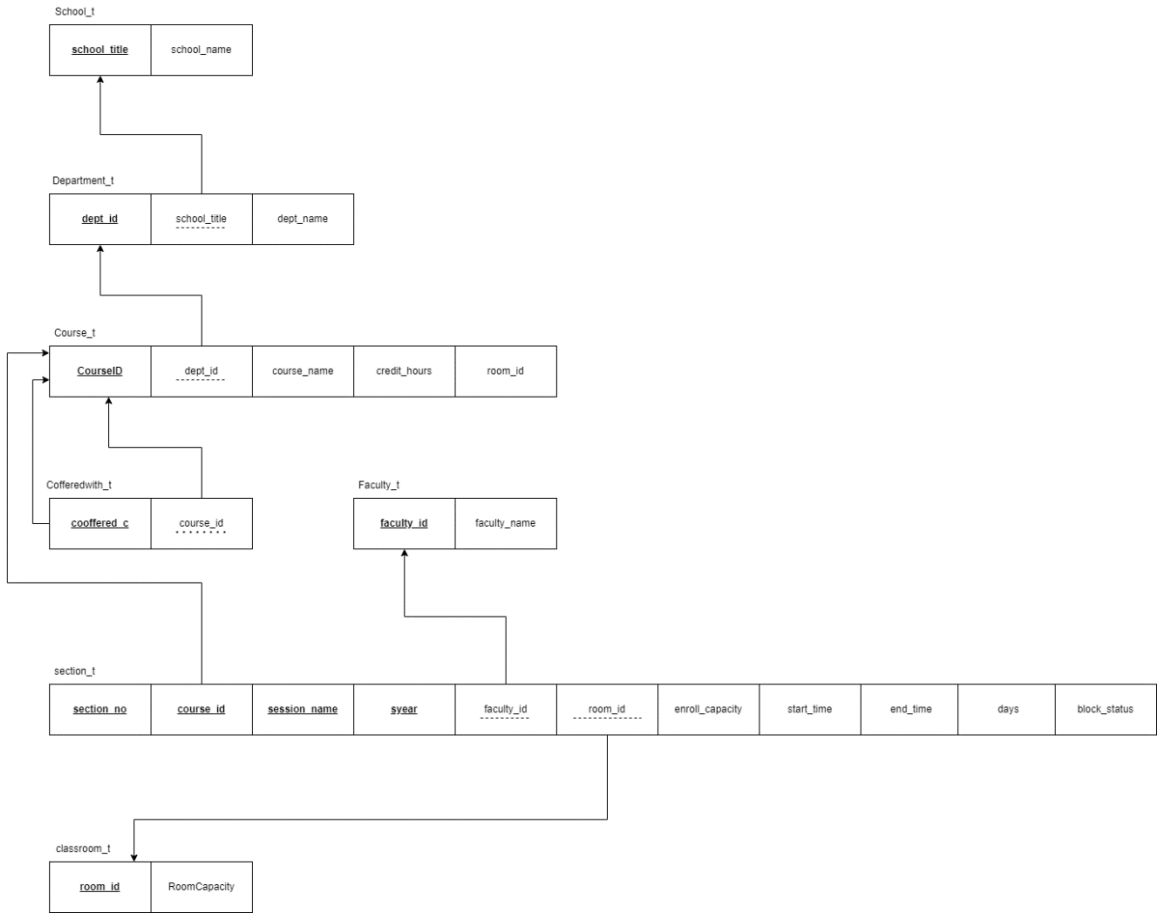



*Figure 18: Relational Schema*

# D: NORMALIZATION

faculty_id -> faculty_name

school_title -> school_name

dept_id -> dept_name, school_title

course_id -> credit_hour, course_name

room_id -> room_capacity

dept_id, course_id, session_name, syear, section_no -> room_id, faculty_id, capacity, enroll_capacity, block_status, start_time, end_time

coffered_c -> course_id


*Figure 19: Normalization*

# E: DATA DICTIONARY

SCHOOL:

| Name | type | Size | Remark |
|---|---|---|---|
| school_title | VARCHAR | 6 | This is the primary<br>Key of this relation. It contains the ID of the school.<br>Ex: SETS |
| school_name | VARCHAR | 30 | This is the name of the school<br>Ex: School of engineering and technology. |

DEPARTMENT:

| NAME | TYPE | SIZE | REMARK |
|---|---|---|---|
| dept_id | VARCHAR | 5 | This is the primary key<br>Of this relation which contains the ID of dept<br>Ex: CSE, EEE |
| dept_name | VARCHAR | 30 | This is the name of dept.<br>Ex: Computer Science and Engineering. |
| school_id | VARCHAR | 6 | This is the foreign key from SCHOOL table. |

Course:

| NAME | TYPE | SIZE | REMARK |
|---|---|---|---|
| course_id | VARCHAR | 10 | This is the primary key<br>Of this relation which contains the ID of course.<br>Ex: CSE303 |
| course_name | VARCHAR | 30 | This is the name of course.<br>Ex: Database management system |
| credit_hour | NUMBER | 1 | It contains the number of credits for a course.<br>Ex: 3 credits for DBMS |
| dept_id | VARCHAR | 10 | This is the foreign key from Department table. |
| room_id | NUMBER | 6 | It contains the number of classrooms<br>Ex: 5 |
| coffered_c | VARCHAR | 6 | It contains number of prerequisite courses.<br>Ex: 203 prerequisite 201 |

FACULTY:

| NAME | TYPE | SIZE | REMARK |
|---|---|---|---|
| Faculty_id | VARCHAR | 4 | This is the primary key of the relation. It contains the Id of the faculty.<br>Ex:4242 |
| faculty_name | VARCHAR | 30 | It contains the name of the faculty.<br>EX: Ms. Sadita Ahmed |
| Course_id | VARCHAR | 7 | Course Id is the foreign |

| | | | |
|---|---|---|---|
| | | | Key from COURSE table. |

SECTION:

| NAME | Type | Size | Remark |
|---|---|---|---|
| Section_no | NUMBER | 10 | This is the primary key of this relation.<br>It contains the total number of sections.<br>Ex: CSE 303 HAS 4 SECTIONS |
| course_id | NUMBER | 10 | This is the primary key as well as a foreign key from COURSE table. |
| session | VARCHAR | 15 | This is a primary key of this relation. It contains the session of the semester.<br>Ex: Summer |
| year | NUMBER | 4 | Contains the year |
| faculty_id | NUMBER | 10 | Faculty ID is the foreign key from FACULTY table. |
| room_id | NUMBER | 10 | Classroom number is a foreign key from CLASSROOM table. |
| Enrolled_capacity | NUMBER | 2 | It contains the number of enrolled capacity.<br>Ex: CSE303 has 30 seats |
| capacity | INT | 2 | It contains the capacity of the room |
| start_time | VARCHAR | 6 | It contains the time and date of a course.<br>Ex:8AM |

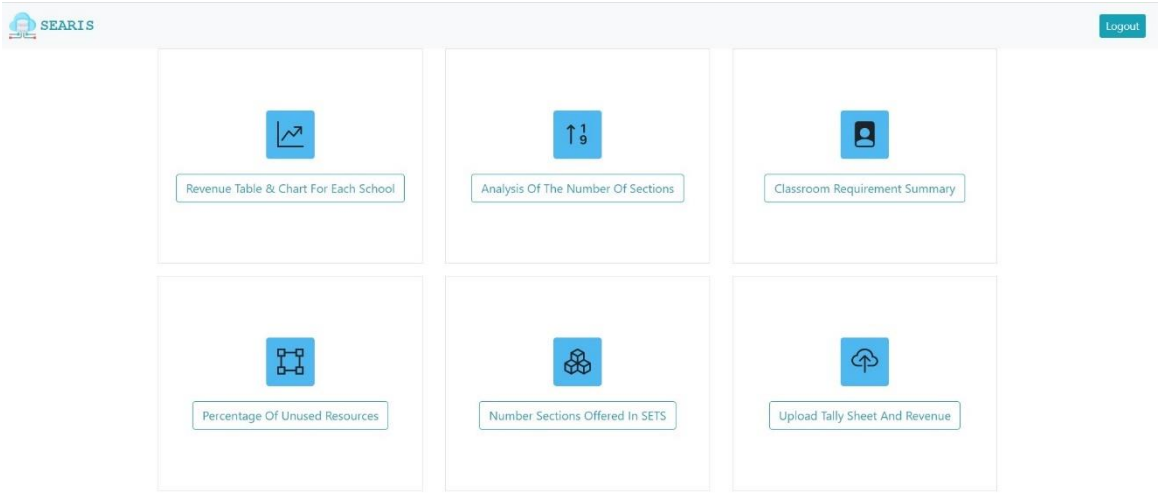
| | | | |
|---|---|---|---|
| end_time | VARCHAR | 6 | It contains the time and date of a course.<br>Ex: 11am |
| days | VARCHAR | 2 | It contains the time and date of a course.<br>Ex: MW |

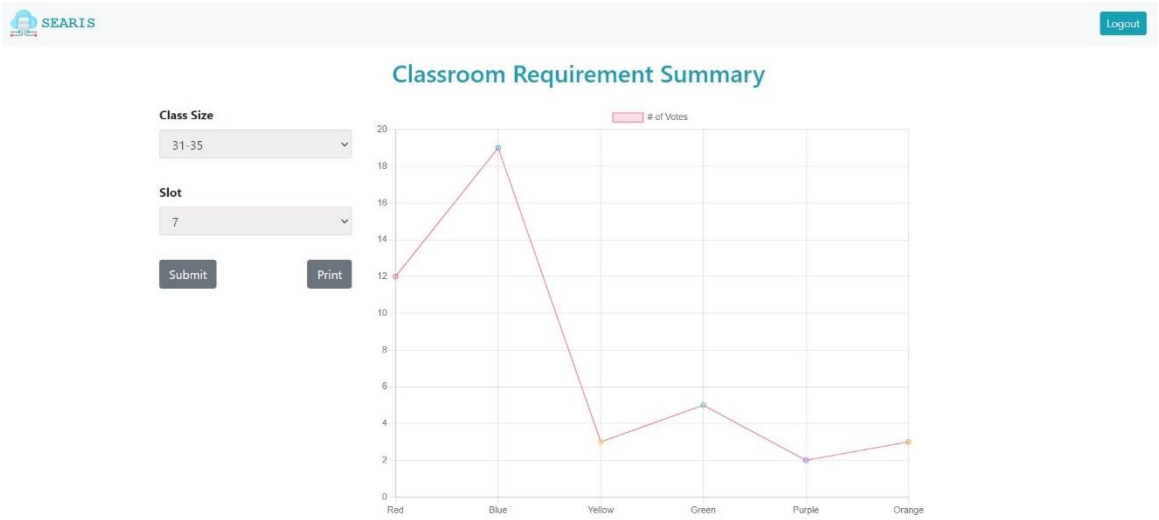
CLASSROOM:

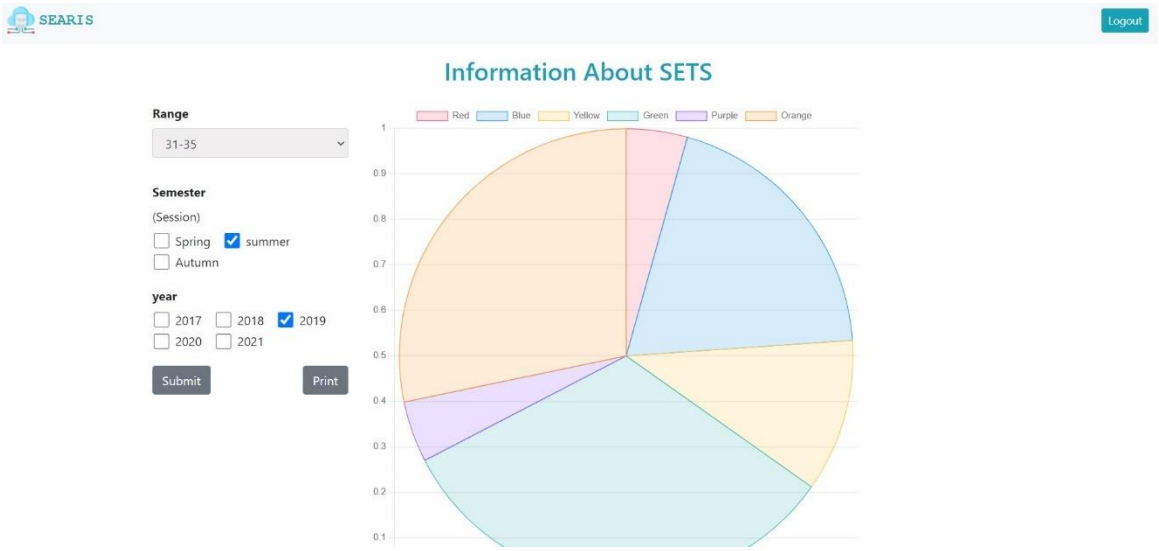
| Name | Type | Size | Remark |
|---|---|---|---|
| Room_id | VARCHAR | 6 | This is the primary key of this relation it contains the classroomNumber.<br>Ex:BC5004 |
| capacity | NUMBER | 6 | It contains the maximm capacity of the class.<br>Ex: CSE303(30) |

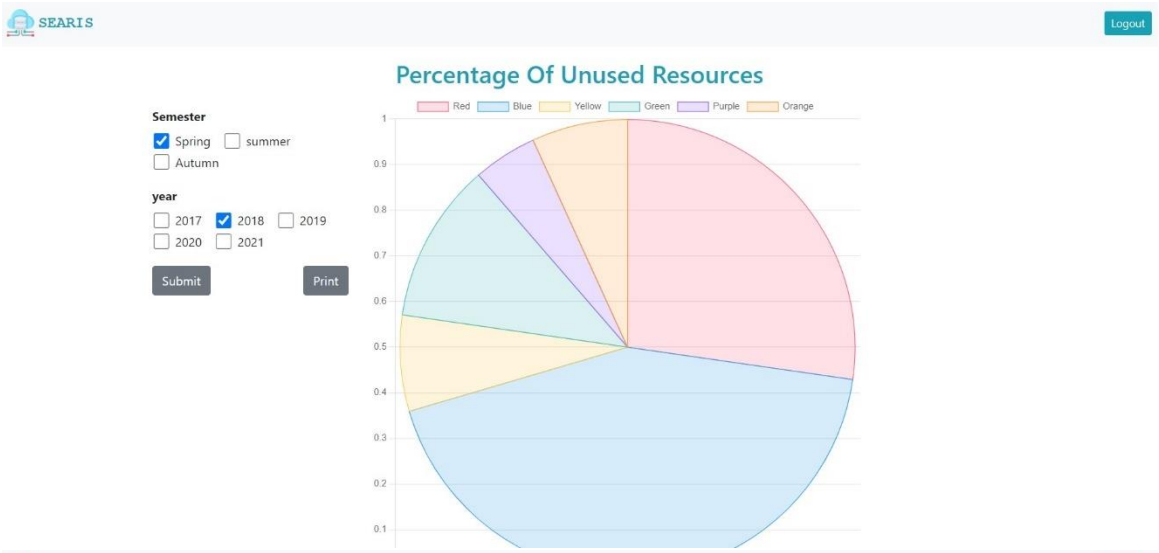
# CHAPTER 4

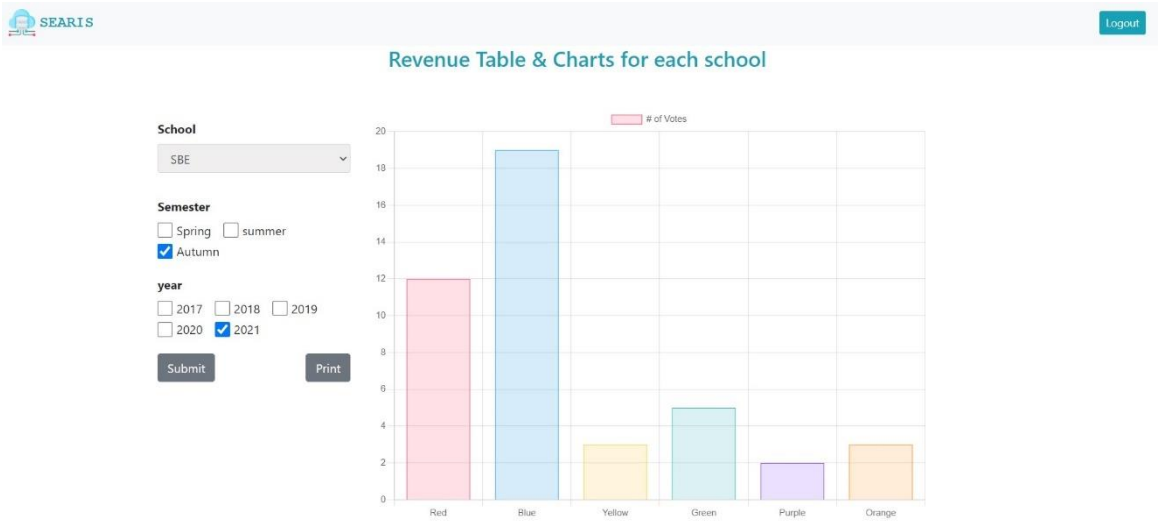
## A. INPUT FORMS

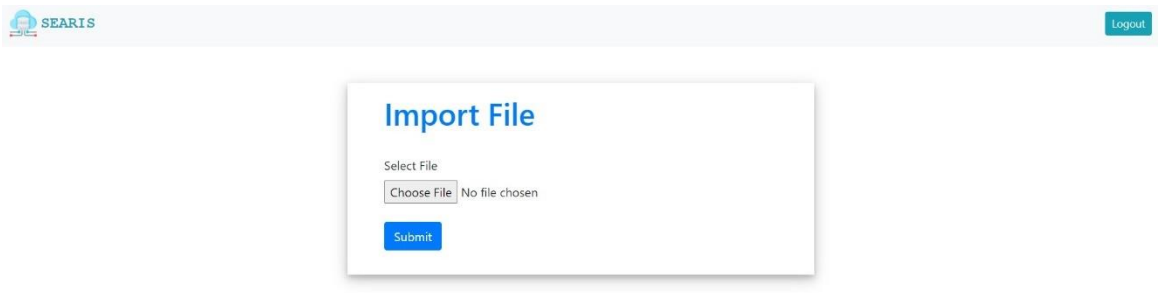




SEARIS
Logout
Classroom Requirement Summary
Class Size
31-35
Slot
7
Submit
Print
# of Votes
20
18
16
14
12
10
8
6
4
2
0
Red
Blue
Yellow
Green
Purple
Orange


SEARIS
Logout
Information About SETS
Range
31-35
Semester
(Session)
Spring
summer
Autumn
year
2017
2018
2019
2020
2021
Submit
Print
Red
Blue
Yellow
Green
Purple
Orange
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


SEARIS
Logout
Percentage Of Unused Resources
Semester
Spring
summer
Autumn
year
2017
2018
2019
2020
2021
Submit
Print
Red
Blue
Yellow
Green
Purple
Orange


SEARIS
Logout
Revenue Table & Charts for each school
School
SBE
Semester
Spring
summer
Autumn
year
2017
2018
2019
2020
2021
Submit
Print
# of Votes
Red
Blue
Yellow
Green
Purple
Orange


SEARIS
Logout
Import File
Select File
Choose File
No file chosen
Submit

# B. OUTPUT QUERY & REPORTS

```
from django.db import connection
from django.shortcuts import render

# Create your views here.


def Revenue( request ):

    query = """
        SELECT school_title AS sch
        FROM school_t;
    """
    with connection.cursor() as cursor:
        cursor.execute( query )
        schools = [ row[0] for row in cursor.fetchall() ]

    sqlClause = ""
    for school in schools:
        sqlClause += f"SUM( CASE WHEN E.School = '{school}' THEN Revenue ELSE 0 END ) AS {school},\n"



    query = f"""
SELECT SUM( Revenue ) AS Total,
            ROUND(
                100 * (
                     SUM( Revenue ) - LAG( SUM( Revenue ), 3,  SUM( Revenue ) ) OVER ()
                ) / SUM( Revenue )
            ) AS '% Change'
        FROM (
            SELECT
                syear AS Years,
                session_name AS Sessions,
                SUM( S.enroll_capacity * C.credit_hours ) AS Revenue,
                D.school_title AS School
            FROM
                section_t S,
                coofferedwith_t O,
                course_t C,
                department_t D,
                (
                    SELECT syear AS Years, session_name AS Sessions
                    FROM Section_T
                    GROUP BY syear, session_name
                ) M
            WHERE
                    S.course_id = O.course_id
                AND O.course_id = C.course_id
                AND S.dept_id = D.dept_id
                AND S.syear = M.Years
                AND S.session_name = M.Sessions
            GROUP BY Years, Sessions, School
            ORDER BY Years, Sessions
        ) E
        GROUP BY Years, Sessions
```

```
    ORDER BY Years, Sessions DESC
"""
with connection.cursor() as cursor:
    cursor.execute( query )
    labels = [ col[0] for col in cursor.description ]
    data = cursor.fetchall()

# xAxis, yAxis, totals, changes =RevenueChartDataPacker( data, labels )

return render( request, "revenue/revenue_iub.html", {
        'colNames': labels,
        'revenues': data,
        # 'xAxis': xAxis,
        # 'yAxis': yAxis ,
        # 'totals': totals,
        # 'changes': changes
    }
)
```

```
from urllib import request
from django.shortcuts import render

# Create your views here.
def chart(request):
    query="""
            SELECT (
                CASE

                    WHEN S.enroll_capacity BETWEEN  1 AND 10 THEN '  1-10'
                    WHEN S.enroll_capacity BETWEEN 11 AND 20 THEN ' 11-20'
                    WHEN S.enroll_capacity BETWEEN 21 AND 30 THEN ' 21-30'
                    WHEN S.enroll_capacity BETWEEN 31 AND 35 THEN ' 31-35'
                    WHEN S.enroll_capacity BETWEEN 36 AND 40 THEN ' 36-40'
                    WHEN S.enroll_capacity BETWEEN 41 AND 50 THEN ' 41-50'
                    WHEN S.enroll_capacity BETWEEN 51 AND 55 THEN ' 51-55'
                    WHEN S.enroll_capacity BETWEEN 56 AND 60 THEN ' 56-60'
                    WHEN S.enroll_capacity > 60 THEN ' 60+'
                END
            ) AS Enrollment, S.dept_id AS department, COUNT(*) AS Total_Sections
            FROM section_t S, department_t D
            WHERE
                S.dept_id = D.dept_id
                AND D.school_title="SETS" AND
                S.session= {n} AND S.year={m} AND Enrollment= 1-10
            GROUP BY Enrollment, department
            ORDER BY  Enrollment,department ASC
            """
        with connection.cursor() as cursor:
            cursor.execute(query)
            result = cursor.fetchall()

    return render(request,'Number_of_section_offered_SETS.html')
```

```
from django.db import connection
from django.shortcuts import render

# Create your views here.
def generate(request):


    query1= """
            select d.school_title, count() AS Count
            from section_t AS S, department_t AS D
            where S.dept_id=D.dept_id AND
            S.session= {n} AND S.year={m} AND
            enroll_capacity<50
            group BY S.dept_id
            """
    with connection.cursor() as cursor:
        cursor.execute(query1)
        result = cursor.fetchall()
    query2= """
                select sum(Count) AS total
                from(
                    select d.school_title, count() AS Count
                    from section_t AS S, department_t AS D
                    where S.dept_id=D.dept_id AND
                    S.session= {n} AND S.year={m} AND
                    enroll_capacity<50
                    group BY S.dept_id) AS total
            """
    result=[]
    with connection.cursor() as cursor:
        cursor.execute(query2)
        result = cursor.fetchall()
        context={
            'school_title':school_title
            'school_total':count
            'total':total
        }
        return render(request,'Analysis_the_number_of_sections.html')
```

```
from django.shortcuts import render
from django.db import connection

'''
Classroom requirement summary based on class size, total slots per day, and sections
offered:
Based on a given range of class size (any of the following ranges selected by user: 1-20,
21-30, 31-35, 36-40, 41-50, 51-54, 55-64, 65-124, 125-168; class size = number of
students enrolled in a section), and accordingly the number of sections offered in a
semester, how many classrooms would be needed if each day's schedule was divided
into seven slots versus the number of such such classroom that are actually available?
What about the same for eight slots?
'''

def classroom_requirement(request):

    query = """
            SELECT session_name      #to get session names
            FROM section_t
            GROUP BY session_name
            """
    sessions = [] #creating a dictionary containing all session names
    with connection.cursor() as cursor:
        cursor.execute(query)
        sessions = cursor.fetchall() #fetches all the rows of the query above and stores them as tuples


    query = """
            SELECT syear
            FROM section_t
            GROUP BY syear
            ORDER BY syear
            """
    years = []
    with connection.cursor() as cursor:
        cursor.execute(query)
        years = cursor.fetchall()


# need to add custom code as per table/chart
    if request.method == 'POST':
        year = request.POST.get('selected_year', "2020")
        session = request.POST.get('selected_session', "Autumn")

    else:
        year = years[-1][0]
        session = sessions[-1][0]

    #Used round function here to round the value in 1 decimal place
```

```
query = """
        SELECT DATA.size AS "Class Size", DATA.sections, DATA.class7 AS "Slots-7", DATA.class8 AS "Slots-8"
        FROM(
            SELECT (

                CASE
                    WHEN s.enroll_capacity  BETWEEN  1 AND  10 THEN '1-10'
                    WHEN s.enroll_capacity  BETWEEN 21 AND 30 THEN '21-30'
                    WHEN s.enroll_capacity  BETWEEN 30 AND 35 THEN '30-35'
                    WHEN s.enroll_capacity  BETWEEN 36 AND 40 THEN '36-40'
                    WHEN s.enroll_capacity  BETWEEN 41 AND 50 THEN '41-50'
                    WHEN s.enroll_capacity  BETWEEN 51 AND 54 THEN '51-54'
                    WHEN s.enroll_capacity  BETWEEN 55 AND 64 THEN '55-64'
                    WHEN s.enroll_capacity  BETWEEN 65 AND 124 THEN '65-124'
                    WHEN s.enroll_capacity  BETWEEN 124 AND 168 THEN '124-168'

                    ELSE '0'

                END
            ) AS size, ROUND(COUNT(*)/14, 1) AS class7, ROUND(COUNT(*)/16, 1) AS class8,COUNT(*) AS sections

            FROM section_t AS s
            WHERE
            syear = %(year)s
            AND session_name = %(session)s
            GROUP BY size
            HAVING size NOT IN('0')
            ORDER BY size
        ) DATA
    UNION
     SELECT COUNT(*)
     FROM (
      SELECT (
                CASE
                    WHEN s.capacity>=1 AND s.capacity >=10 AND s.enroll_capacity=0 THEN '1'
                    WHEN s.capacity>=1 AND s.capacity >=30 AND s.enroll_capacity=0 THEN '21-30'
                    WHEN s.capacity>=1 AND s.capacity >=35 AND s.enroll_capacity=0 THEN '30-35'
                    WHEN s.capacity>=1 AND s.capacity >=40 AND s.enroll_capacity=0 THEN '36-40'
                    WHEN s.capacity>=1 AND s.capacity >=50 AND s.enroll_capacity=0THEN '41-50'
                    WHEN s.capacity>=1 AND s.capacity >=54 AND s.enroll_capacity=0THEN '51-54'
                    WHEN s.capacity>=1 AND s.capacity >=64 AND s.enroll_capacity=0 THEN '55-64'
                    WHEN s.capacity>=1 AND s.capacity >=124 AND s.enroll_capacity=0 THEN '65-124'
                    WHEN s.capacity>=1 AND s.capacity >=168 AND s.enroll_capacity=0 THEN'124-168'

                    ELSE '0'

                END

     )AS Available_Class

            FROM section_t AS s
            WHERE
                syear = %(year)s
```

```
                AND session_name = %(session)s
                GROUP BY size
                HAVING size NOT IN('0')
                ORDER BY size
            ) DATA
        UNION
         SELECT COUNT(*)
         FROM (
          SELECT (
                    CASE
                        WHEN s.capacity>=1 AND s.capacity >=10 AND s.enroll_capacity=0 THEN '1'
                        WHEN s.capacity>=1 AND s.capacity >=30 AND s.enroll_capacity=0 THEN '21-30'
                        WHEN s.capacity>=1 AND s.capacity >=35 AND s.enroll_capacity=0 THEN '30-35'
                        WHEN s.capacity>=1 AND s.capacity >=40 AND s.enroll_capacity=0 THEN '36-40'
                        WHEN s.capacity>=1 AND s.capacity >=50 AND s.enroll_capacity=0THEN '41-50'
                        WHEN s.capacity>=1 AND s.capacity >=54 AND s.enroll_capacity=0THEN '51-54'
                        WHEN s.capacity>=1 AND s.capacity >=64 AND s.enroll_capacity=0 THEN '55-64'
                        WHEN s.capacity>=1 AND s.capacity >=124 AND s.enroll_capacity=0 THEN '65-124'
                        WHEN s.capacity>=1 AND s.capacity >=168 AND s.enroll_capacity=0 THEN'124-168'

                        ELSE '0'

                    END

         )AS Available_Class

                FROM section_t AS s
                WHERE
                    syear = %(year)s
                    AND session_name = %(session)s

        ) M
        """

    val={
        "year" : str(year),
        "session" : session,
    }

    with connection.cursor() as cursor:
        cursor.execute( query , val)

    header = [ col[0] for col in cursor.description ]
    data = cursor.fetchall()
    context = {
        'tableHeaders': header,
        'tableData'   : data,
        'years'       : years,
        'sessions'    : sessions,
        'selectedSession' : session,
        'selectedYear'    : year,
    }
    return render(request, 'Classroom_Requirement_Summary.html', context)
```

```
from django.shortcuts import render
from django.db import connection


# Create your views here.
'''
Comparative analysis of percentage of unused resources for each school in selected
semester/s, as well as w.r.t to total unused percentage for the university overall based on
average enrollment per section versus average capacity per room.
'''
def unusedresources(request):

#to get session names
    query =   """
            SELECT session_name
            FROM section_t
            GROUP BY session_name
            """
    sessions = [] #creating a dictionary containing all session names
    with connection.cursor() as cursor:
        cursor.execute(query)
        sessions = cursor.fetchall() #fetches all the rows of the query above and stores them as tuples
   # return render(request, 'Percentage_of_unused_resources.html')


    yquery = """
            SELECT syear
            FROM section_t
            GROUP BY syear
            ORDER BY syear
            """
    years = []
    with connection.cursor() as cursor:
        cursor.execute(yquery)
        years = cursor.fetchall()

    year=''
    session= ''
#use custom code to get selected year and session which counts for specific semester
    if request.method == 'POST':
        year = request.POST.get('selected_year', "2020")
        session = request.POST.get('selected_session', "Autumn")

    else:
        year = years[-1][0]
        session = sessions[-1][0]

#rounded the values, grouped by school title as the query asks for each school's  info

    rquery =  """
            SELECT School_Title AS School_Title,
             Enroll_Avg AS Enroll_Avg, Avg_Capacity_Room AS Avg_Capacity_Room, unused AS Unused_Resources_Percentage

            FROM (
                SELECT
                    d.school_title AS School_Title,
                    ROUND(AVG( s.enroll_capacity),3) AS Enroll_Avg,
                    ROUND(AVG( c.room_capacity ),3) AS Avg_Capacity_Room,
                    ( AVG(c.room_capacity) - AVG( s.enroll_capacity )) AS value,
                    ROUND((value/Avg_Capacity_Room)*100, 3) AS unused

                FROM department_t AS d, section_t as s, classroom_t c
                WHERE
                    d.dept_id = s.dept_id
                    AND s.room_id= c.room_id
                    AND s.session = %(session)s
                    AND s.syear = %(year)s
                    GROUP BY d.school_title
                ) AS DATA
            """

    val={
        "year" : str(year),
        "session" : session,
    }
    with connection.cursor() as cursor:
        cursor.execute( rquery , val)
        # print(connection.queries)
        tableHeaders = [ col[0] for col in cursor.description ]
        tableData = cursor.fetchall()
```

# CHAPTER 5

## PROBLEM & SOLUTION

| Problems | Solutions |
|---|---|
| The tally sheets did not contain department ID's for courses. | We manually inserted the department ID for the values. |
| Blocked columns had different values like B-0 B-1 B- which was confusing. | We only kept two set of values- 0 and -1. |

| Problems | Unsolved |
|---|---|
| Importing Tally Sheets | We tried to use pandas to import the excel file. However, it did not work. |

## ADDITIONAL FEATURE & FUTURE DEVELOPMENT

1. Every time a user refreshes the website the same queries run on the database. If the cache feature could be implemented, it could be more efficient.
2. IRAS database can be connected with our website's database which would enhance the upload tally sheet process.
3. We can use Machine Learning to predict the bare minimum number of seats required for an upcoming semester.

# REFERENCES

[1] IUB website.

[2] Wikipedia, "Business_Process_Model_and_Notation".

[3] A. M. A. S. HOWARD, methods & tools.

[4] ,. T. K. C. b. Di-An Hong a, "A three-dimensional, six-segment chain analysis of forceful".