**AUTOMATED TIMETABLE SCHEDULING SYSTEM USING BACKTRACKING ALGORITHM**

**Hernando Jr. J. Costelo**

**Michelle Emmanuel S. Gaetos**

**Johnuel M. Javier**

**Ren Russel E. Lavilla**

**Lester D. Quijano**

An undergraduate thesis presented to the faculty of the Department of Computer Studies Cavite State University-Cavite College of Arts and Trades Campus, Rosario, Cavite in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science with Contribution No. \_\_\_\_\_\_\_\_\_\_\_\_. Prepared under the supervision of Mr. Karlo Jose E. Nabablit.

**INTRODUCTION**

Despite ongoing advancements in technology that have led to the computerization and automation of various aspects of university organization work, the process of creating course timetables still remains a manual and challenging task. The preparation of course timetables had a significant time constraint for academic colleges. It involves addressing multiple constraints and issues, such as classroom availability, teachers' subject preferences, the number of available teachers and classroom capacities, potential conflicts between rooms and courses, and scheduling conflicts between courses and instructors. Generally, course timetabling in many universities is prepared manually and the scheduling committee should consider all available facilities and resources, such as courses, instructors, rooms and laboratories. Moreover, the instructors, course and sections time were important constraints to handle. Therefore, based on all the mentioned constraints, course timetabling is a very exhaustive and time-consuming task. (Al-Jarrah et al, 2017)

Cavite State University CCAT Campus had offered 9 courses, including BS Electrical Engineering, BS Computer Engineering, BS Computer Science, BS Information Technology, BS Hotel Management, BS Business Managements, BS Technical-Vocational Teacher Education, BS Education with 2 different Majors, and BS Industrial Technology with 9 different Majors. As the campus continued to grow, the number of students increased, making the manual creation of course timetables a complex task to accomplish. In CVSU CCAT, the creation of schedules was manually done by the faculty head or scheduling committee of each department. The previous scheduling process consumed more time as the plotting, revising, and arranging of schedules were done separately for each department. Each course had minor and major subjects, and the scheduling committee of each department had to wait for the major subjects to be plotted in the timetable before plotting the minor subjects.

Furthermore, the researchers came up with an idea and proposed the development of a web-based Automated Timetable Scheduling System to reduce the difficult work of the faculty head or scheduling committee in creating schedules and to save time in the scheduling process. The web application aimed to generate an optimal class schedule for Cavite State CCAT-Campus within a short period of time. Additionally, the web application assisted the university scheduling committee in managing schedules with ease through an intuitive user interface.

**Objective of the Study**

The main objective of the study was to create a dynamic web-based scheduling system capable of generating conflict-free schedules. The system specifically caters to the scheduling needs of Cavite State University-CCAT Campus especially in Department of Computer Studies, effectively eliminating the need for manual scheduling.

Specifically, the study aimed to:

1. Document and analyze the existing process of scheduling classes in Cavite State University – CCAT;
2. design and develop a web application that automate the generation of class schedules;
3. test the reliability of generated schedules based on testing conditions and test the functionality of system modules.
4. Evaluate the system using modified ISO-IEC 25101 software evaluation instrument; and
5. Create an implementation plan for the deployment of the software.

**Significance of the Study**

Through the implementation of a backtracking algorithm, the developed system made the process of generating a class schedule more efficient. This implementation resulted in a notable reduction in time requirements and a decrease in the likelihood of human errors during the scheduling process. As a result, Cavite State University CCAT-Campus may obtain an optimal class schedule. Moreover, the system may be beneficial for the scheduling committee as it simplified tasks related to adding, deleting, and updating vital information for the faculty loading process.

**Scope and Limitation of the Study**

The study was focused on the development of the Automated Scheduling System for Cavite State University - CCAT Campus. The system was designed with features that allowed users to automate the generation of class schedules using a backtracking algorithm. Furthermore, during the study, significant importance was given to the utilization of web-based system development to ensure the ease of accessing data, an intuitive user interface, and reliable storage.

ATS only had a separate view page for generated schedules for teachers, course year, section, and rooms. The output could be generated in PDF format for printing.

ATS was not designed for multiple users. The primary target user of the system. was the scheduling committee

ATS can have multiple accounts with the same access, such as adding, updating, and deleting data. The target user can also assign subjects to a teacher and then generate the automated schedule and print using PDF.

ATS was specifically designed to generate an automated schedule for the Department of Computer Science only. The generated schedules cannot be updated since conflicts may arise. However, if you want to add, update, or delete data in the faculty loading data, you need to regenerate the schedule to have an optimal schedule without conflicts.

ATS only generates schedules for full-time teachers and regular students. It does not consider irregular students and course schedules.

**Time and Place of the Study**

The study was conducted from May 2022 to May 2023 at the Cavite State University-CCAT Campus.

**Definition of terms**

***ATS*** stands for “Automated Timetable Scheduling System”.

***Backtracking Algorithm*** used to automate the schedules. The backtracking algorithm is a technique used to systematically search for solutions to a problem by incrementally building a solution and then undoing or "backtracking" when a dead-end was reached. This algorithm was used to recursively generate timeslots, rooms, days to a course.

***Backtrack*** when automating the schedules and conflicts occurs, the algorithm goes back to the previous state to explore other possibilities to find timeslots without conflicts and this is known as backtracking.

***Constraints*** are the conditions or rules that must be satisfied by a valid solution. they help prune the search space and guide the backtracking process.

***Course Timetabling*** is the assigning of class schedules to a specific course including teachers, rooms, timeslot, and subjects.

***Recursive Backtracking*** is the type of backtracking algorithm used by the developers to automated the assigning of timeslots to different courses.

***Solution space*** represents all the potential solution or potential valid schedules that are free of conflicts.

***TCPDF*** stands for "The PHP PDF Creation Library," which is a robust tool used for converting tables into PDF format. Its purpose in ATS SYSTEM was to enable the printing of the generated schedules in a visually pleasing and easily shareable format.

**REVIEW OF RELATED LITERATURE**

In this chapter the developers review studies, articles, software, case studies related to the development of the Automated Scheduling System. It also shows the relevant system, its features and method prior to the development of the automated scheduling system.

**Importance of gathering data**

According to Simple learn (2022), gathering and analyzing data before the software development help us to make better decisions and help us evaluate possible outcome.

According to CQL (2022), gathering data helps us identify existing problems and identify the appropriate approaches on how to solve those problems.

**Importance of using web application**

Web based application offers a wide variety of advantages compared to desktop application. Instead of installing the application or software to each of every computer your company or school has, it could be easily accessed through the use of internet. Maintaining a web application is a simple process since the host server could easily add or update data without the need to upgrade in every computer. (Khamooshi, 2019)

According to Souvik (2022), web application increase efficiency and reduce costs by automating the processes and task and reduce costs by allowing the user to add and update data through online, thus, it eliminates paper based manual data entry.

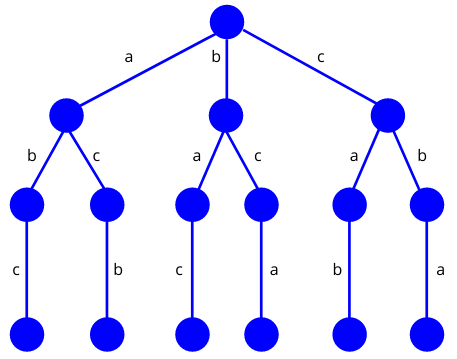
**Course Timetabling**

Course timetabling main objective is to assign a course to a specific time or set of time without conflicts to another course allotted time. It is also one of the university timetabling problems, it is resolved by meeting the specific constraints. This constraint is divided into two parts, the former is the hard constraints, and the latter is the soft constraints. (Burke and Petrovic, 2002). Hard constraints are the problems that must be satisfied/solved to generate a great timetable while soft constraints are the desirable problems but not essential to be solve. (Houhandi et. al, 2019). According to Ben Moreland (2015), these are the examples of hard and soft constraints. For hard constraints, Room capacity which means the number of students assigned to the room must best equal to the seating capacity of the room. For soft constraints, A preferred teaching week which means the teacher could choose when to teach the subject to specific time and day.

**Introduction to Backtracking Algorithm**

Backtracking is an algorithmic technique that solves problems through a recursive approach. It involves constructing a solution incrementally, discarding any choices that violate the problem's constraints at each step. It can be considered as an improvement over the brute force method. The fundamental concept behind backtracking is to systematically search for a solution among the available options. Initially, we begin with one possible option and check if the problem can be solved using that choice. If a solution is found, it is returned. Otherwise, we backtrack and explore another option from the remaining choices. In some cases, none of the options may lead to a solution, indicating that backtracking cannot provide a solution for that specific problem. Backtracking can be viewed as a form of recursion, as the process of searching for a solution from the available options is repeated recursively until a solution is found or the final state is reached. In essence, backtracking eliminates choices at each step that cannot lead to a solution, continuing towards choices that have the potential to lead to the desired solution. There are three types of problem in backtracking decision, optimization and enumeration ([Upadhyay](https://www.simplilearn.com/tutorials/data-structure-tutorial/backtracking-algorithm), 2023)

**"Backtracking: Non-Adjacent 'c' Placement in 'abc' Arrangement**

In backtracking, the initial step involves constructing a state-space tree. Subsequently, we explore all potential solutions and verify them against the provided constraint. Only the solutions that meet the given constraint are retained.

**(***Source https://www.baeldung.com/cs/backtracking-algorithmsGenerator)*

Figure 1. Non-adjacent Placement of ‘c’ in “abc” Arrangement.

In backtracking, the initial step involves constructing a state-space tree. Subsequently, we explore all potential solutions and verify them against the provided constraint. We retain only the solutions that meet the given constraint: For instance, considering the problem at hand, the possible solutions are: (a,b,c), (a,c,b), (b,a,c), (b,c,a), (c,a,b), (c,b,a). However, only the valid solutions that satisfy the constraint are included in the final solution set. In this case, the final solutions would be limited to (a,b,c) and (c,b,a) (Datta 2022)

**When to use backtracking algorithm?**

Backtracking finds applications in diverse problem-solving scenarios. For instance, it can be utilized to discover a viable solution for a decision problem. Additionally, backtracking algorithms have proven to be highly effective in solving optimization problems. In certain cases, it can be employed to identify all feasible solutions for an enumeration problem. However, it's important to note that backtracking is not considered an optimal problem-solving technique. Its utility lies in situations where time constraints are not imposed on finding a solution. ([Upadhyay](https://www.simplilearn.com/tutorials/data-structure-tutorial/backtracking-algorithm), 2023)

**Recursive Backtracking and Optimization**

Recursive backtracking is a technique used to solve challenging problems by exploring all possible solutions. According to Stanford University (2017) Backtracking algorithm can be applied to generate and count solutions, find specific solutions, or determine the best solution. Specific problems that can be tackled include generating permutations, subsets, combinations, and more. Backtracking can be effectively employed in three main scenarios:

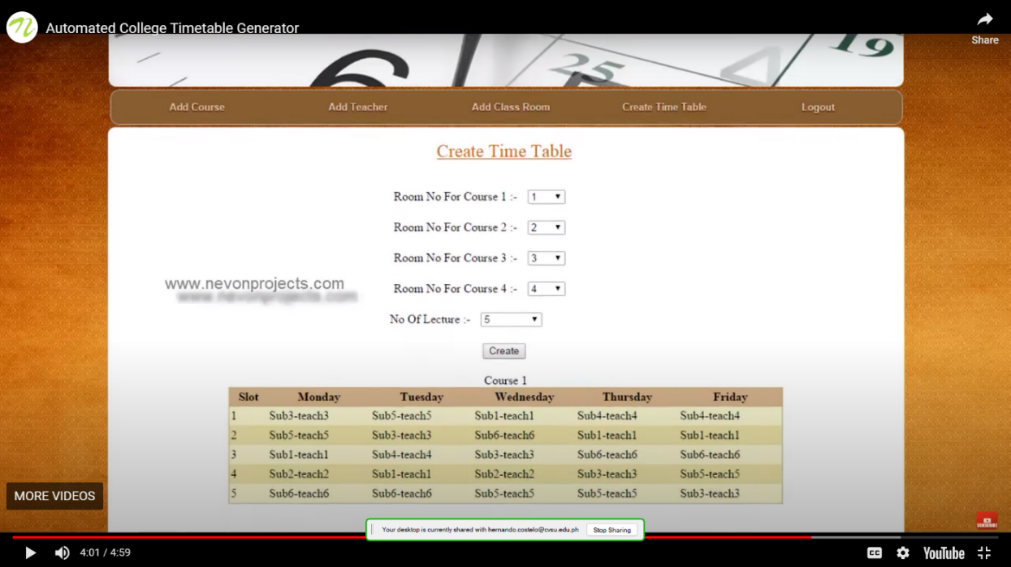
1. Enumeration: Backtracking can generate and count all possible solutions to a problem. This involves systematically exploring different paths and choices to exhaustively generate all valid solutions
2. Existence and search: Backtracking can be used to find a specific solution to a problem or prove its existence. By systematically exploring different paths and choices, it can determine if a valid solution exists and locate it.
3. Optimization: Backtracking can assist in finding the best possible solution to a given problem. It explores different paths and evaluates their quality, making informed decisions to optimize the solution based on defined criteria.

**Related System**

**Automated College Timetable Generator**

This web-based automated college timetable generator is used to automate the college timetable using genetic algorithm. The developed system is a dynamic system that allows user to update data such as add course, teachers, rooms and automate the generation of timetable

The related web application will be used by the developers as reference for the process flow of the system.

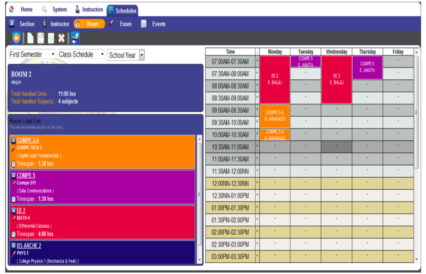


**(***Source: https: Automated College Timetable Generator)*

Figure 2. Create Timetable Tab

**Bohol State Island University Automated Scheduling System Capstone Project**

This study designed an automated class scheduling system for Bohol State Island University and test its functionality in terms of speed, accuracy, data handling, security, stability, and adaptability in making class schedules. This system is a dynamic system because the admin /user could update the data in the database like add teachers, add buildings, add course and adding new user.

 The developers will use this related system as a reference in creating a dynamic scheduling system.

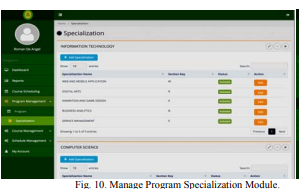
*(Source:* https: Automated Scheduling System Capstone Project)

Figure 3. Automate Schedule Tab

**Web based Course Scheduling System using Greedy Algorithm**

This study is a web-based system that focuses on course scheduling for College of Computer Studies under FEU institute of Technology, Manila Philippines. The developed system manages schedules on courses and assigning to faculty using greedy algorithm and the method of the research was research and development.

This related system will be used by the developers as a reference for creating admin dashboard.

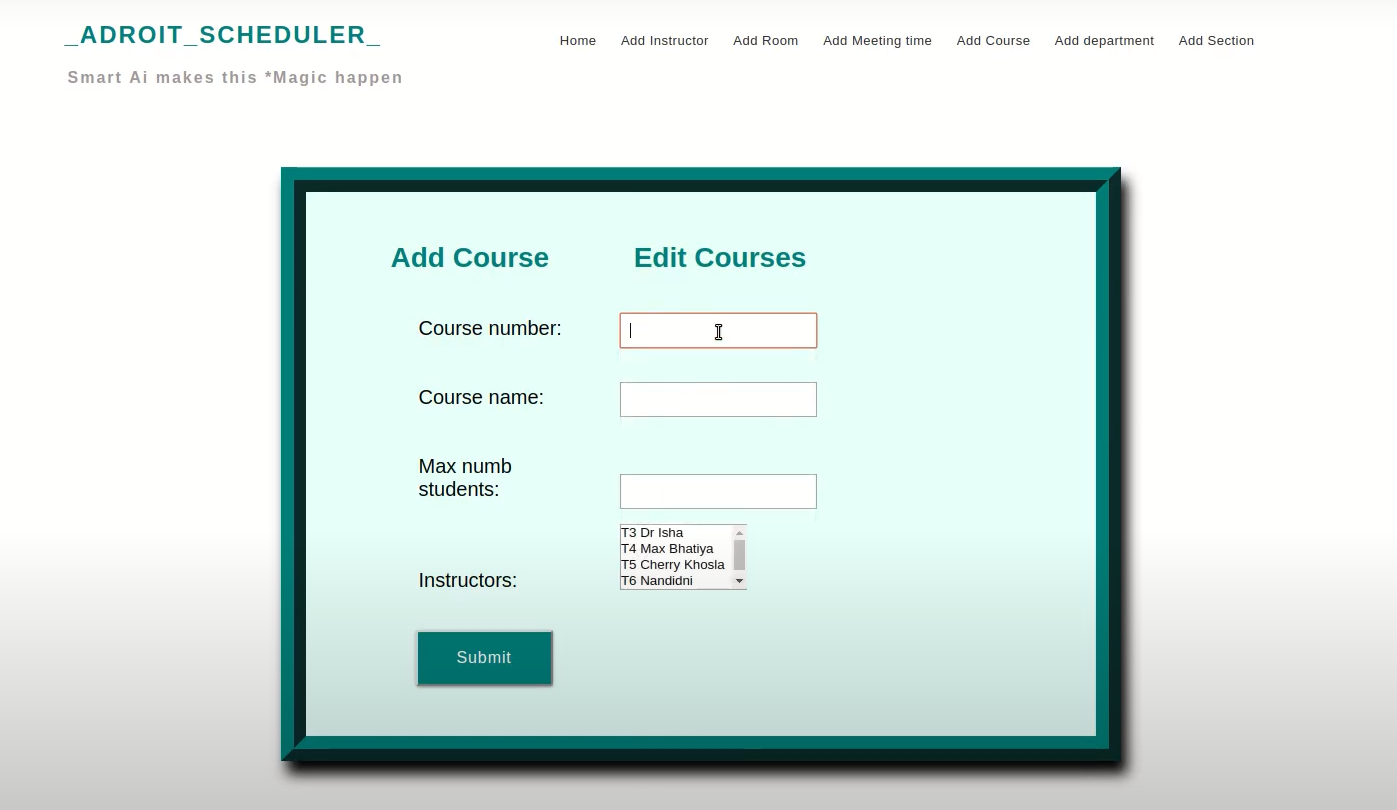


**(***Source*: https://ijssst.info/Vol-20/No-S2/paper14.pdf)

Figure 4. Manage Program Specialization Module

**Intelligent Timetable Generator using Genetic Algorithm**

This related web application uses genetic algorithm to generate university timetable. The development of this project is based on the hard and soft constraints. The researchers will use this related system as a reference for the implementation of genetic algorithm in python.

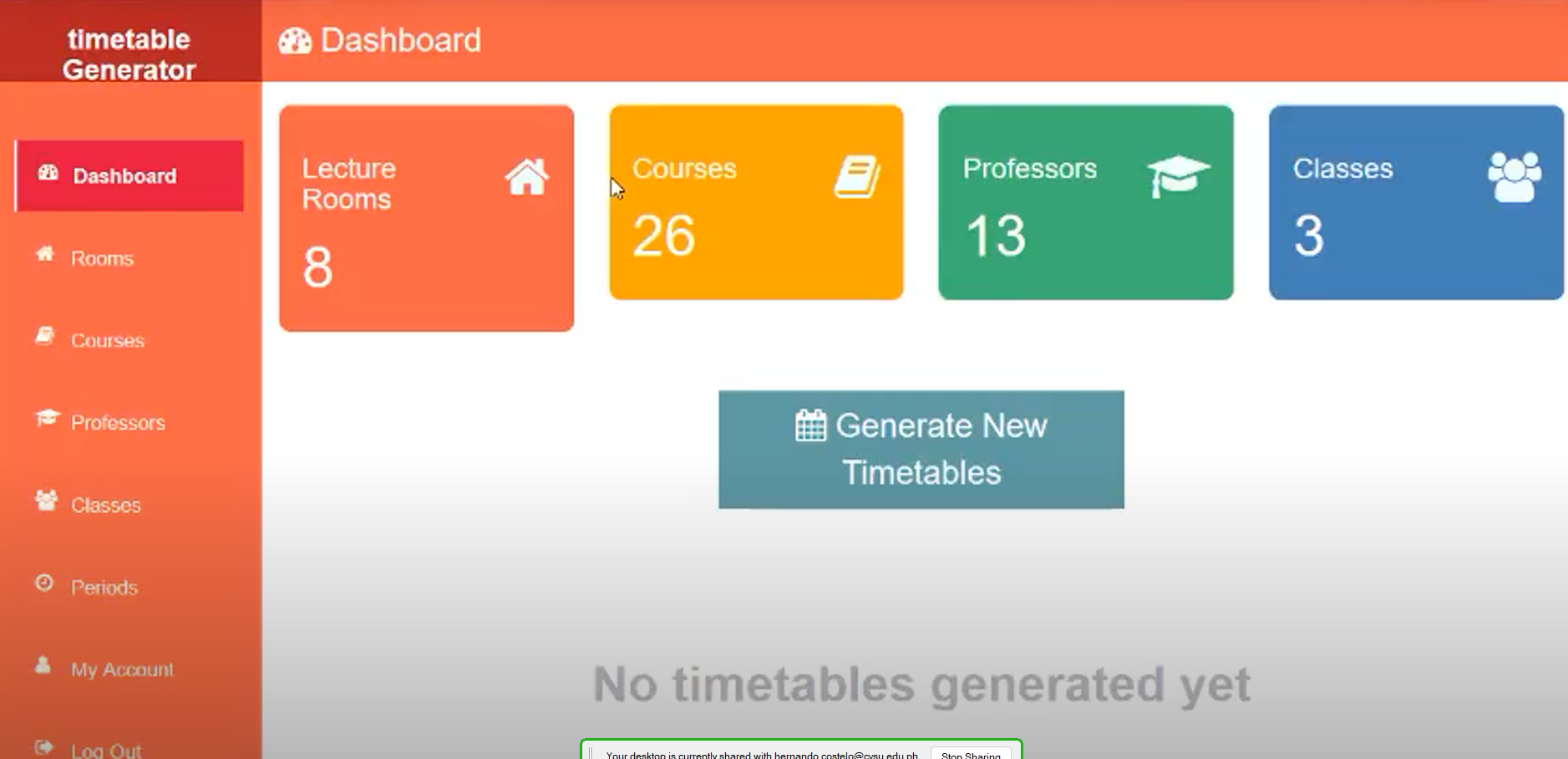
****

(*Source:* http://abhik.pythonanywhere.com)

Figure 5. Add Course Tab

**Timetable Generator using Genetic algorithm.**

Timetable Generator use genetic algorithm to automate the generation of university timetables, the developers of this web application used PHP as their programming language and used Laravel web framework.

****

(*Source:* https://Timetable Generator)

Figure 6. Admin Dashboard

**Synthesis**

In summary, the information obtained from the review of related literature provided valuable insights to the researchers for their study. It enabled them to understand the various stages and processes of backtracking algorithm and genetic algorithm, aiding in the selection of the most suitable method for system development. Additionally, prior to system development, it was crucial for the developers to identify constraints associated with the existing scheduling process at Cavite State University CCAT Campus. This knowledge allowed them to effectively design the process flow of the system.

**Table of Comparison**

The researcher compared the features of proposed system to the other related system as shown below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Features** | **Automated College Timetable Generator** | **Bohol State Island University Automated Scheduling System** | **Web based Course Scheduling System using Greedy Algorithm** | **Intelligent Timetable Generator using Genetic Algorithm** | **Timetable Generator using Genetic algorithm.** | **Automated Timetable Scheduling System using Backtracking Algorithm** |
| **1. Generation of Reports** | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| **2. Faculty loading** | Close with solid fill | Close with solid fill | Close with solid fill | Close with solid fill | Close with solid fill | Checkmark with solid fill |
| **3. Admin Dashboard** | Close with solid fill | Close with solid fill | Checkmark with solid fill | Close with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| **4. View Page for generated schedule** | Close with solid fill | Close with solid fill | Close with solid fill | Close with solid fill | Close with solid fill | Checkmark with solid fill |
| **5. Manual Schedule Integration** | Close with solid fill | Close with solid fill | Close with solid fill | Close with solid fill | Close with solid fill | Checkmark with solid fill |
| **5. Convert Output to a PDF file** | Close with solid fill | Close with solid fill | Close with solid fill | Close with solid fill | Close with solid fill | Checkmark with solid fill |

Figure 7. Table of Comparison

**METHODOLOGY**

In this section, the research method used to conduct the study was defined. The developers explained how they addressed the research objective by utilizing the gathered data and information. The Rapid Application Development Model was used by the developers to achieve faster output. This approach focused on prototyping to quickly deliver or introduce the product, rather than spending excessive time on planning.

**Conceptual Diagram**

**Current Scheduling Process Information**

* Process flow of the current scheduling
* Checklist of Programs offered at CVSU CCAT
* Current Faculty loading data

**Knowledge**

* Course Timetabling
* Backtracking Algorithm
* Web programming
* Database Management

**Software and Hardware**

* MySQL
* PHP
* HTML
* CSS
* JavaScript
* Boostrap5
* JQuery
* Figma
* TCPDF

**INPUT**

**Requirements Analysis Phase**

* The gathered information about the current scheduling process was analyzed.
* The related articles, books, studies, and systems were studied.
* The criteria for assigning timeslots and rooms were identified.

**Prototype Cycle Phase**

* The user interface design was created using Figma.
* The UI Design was demonstrated and refined

**Developing and Testing Phase**

* The UI was converted into a real web application.
* The User interface of the web application was coded.
* The Backtracking algorithm was implemented into the web application.
* The reliability and functionality of the generated output were tested.

**Deployment Phase**

* An implementation plan was created for the deployment of the system.

.

**PROCESS**

**Automated Timetable Scheduling System using Backtracking Algorithm**

**OUTPUT**

Figure 8. Input Process-Output model of Automated Timetable Scheduling System

**System Analysis and Design**

****

Figure 9. Research Framework

* **Requirements Analysis phase**

During the Requirements Analysis phase, the developers identified the different requirements needed and analyzed the related systems, studies, articles, and books for the development of the system.

* **Prototype Cycle Phase**

In the Prototype Cycle Phase, the researchers designed and created a prototype, demonstrated it for refining, and repeated the process until it was ready for the development phase.

* **Developing and Testing Phase**

In the Developing and Testing Phase, the developers converted the prototype into a real web application and tested its reliability and functionality. If the web application met all the criteria needed, it proceeded to the next phase. If not, it went back to the first phase until it met the criteria required for the deployment phase.

* **Deployment Phase**

The developers created an implementation plan specifically for Cavite State University CCAT-Campus to deploy the ATS system. The plan outlined the necessary steps for system installation, configuration, data migration, user training, and post-implementation support.

**Software/System/ Product Description or Process Design**

**Hardware Requirements**

In system development, the researchers used an HP LAPTOP with the following specifications: 8 GB DDR4-3200 MHz RAM, AMD Ryzen 5 5500U with Radeon Graphics operating at a speed of 2100 megahertz, and 512 GB PCIe storage, running on a 64-bit Windows 11 operating system.

**Software Requirements**

* **Front-end:**
  + **HTML**–stands for Hypertext Markup Language and it was used to structure ATS web page and it also used to organize the contents of the webpage.
  + **CSS** – stands for Cascading Style Sheet, it is one of the core technologies of web development. It is used for the design of web pages such as fonts, layouts, colors, shapes and it also used for responsive web pages to different device.
  + **Bootstrap 5** – it is a free and open-source front-end development CSS framework. It was used to provide responsive web design.
  + **JavaScript** - it is a scripting language that are used to create an interactive or dynamic website.
  + **JQuery –** It was used in system development to enhance the functionality and interactivity of the web application. It provided features and utilities for tasks such as DOM manipulation, event handling, AJAX requests, and animations, making it easier to implement dynamic and responsive elements in the system.
* **Back-end:**
  + **PHP -** is a widely-used server-side scripting language specifically designed for web development. It is embedded within HTML code and executed on the server to generate dynamic web pages. PHP offers a wide range of functionalities and is popular for building dynamic websites, web applications, and handling server-side tasks. It can interact with databases, handle form data, and generate dynamic content. PHP is known for its simplicity, flexibility, and strong community support, making it a popular choice for web development.
  + **MySQL** – is a relational database management system that used for storing or handling data using Structure Query Language.
  + **XAMPP** – It is a software package that provides a local development environment for building and testing web applications. It stands for "Cross-Platform, Apache, MySQL. XAMPP is a bundled software solution that includes several components necessary for web development, such as the Apache web server, MySQL database server, PHP programming language
  + **TCPDF** - TCPDF is a PHP library used for generating PDF documents. It stands for "TCPDF - PHP class for PDF". TCPDF provides various functions and methods to create PDF files dynamically, allowing developers to generate PDFs from scratch or based on templates.

***Developmental Tools***

* + **Google Drive -** Google Drive is a file storage, it is used to store research paper, images, diagrams, UI Design and Wireframes
  + **One Drive** – It is a cloud storage that allow you to edit and synchronize file.
  + **Microsoft Teams –** it is used to communicate with one another and schedule a meeting twice a week to report the progress of their task.
  + **Figma –** is an online collaboration tool used for creating digital products. It is used to design user interface of the system.
  + **Lucid Chart** – is a diagramming tool web application, it is used for creating diagrams (e.g., flowcharts, Site map).
  + **Visual Studio Code –** is a code editor and its supports different programming language like Python.
  + **Adobe Photoshop CS6** – is photo editing software, it is used for the design and image of the website.
  + **GitHub -** GitHub is a web-based platform that serves as a version control repository for developers. It provides a centralized location where developers can store, manage, and collaborate on their code projects. The platform uses Git, a distributed version control system, to track changes to files and facilitate collaboration among team members.

**Implementation of the Backtracking Algorithm in the System**

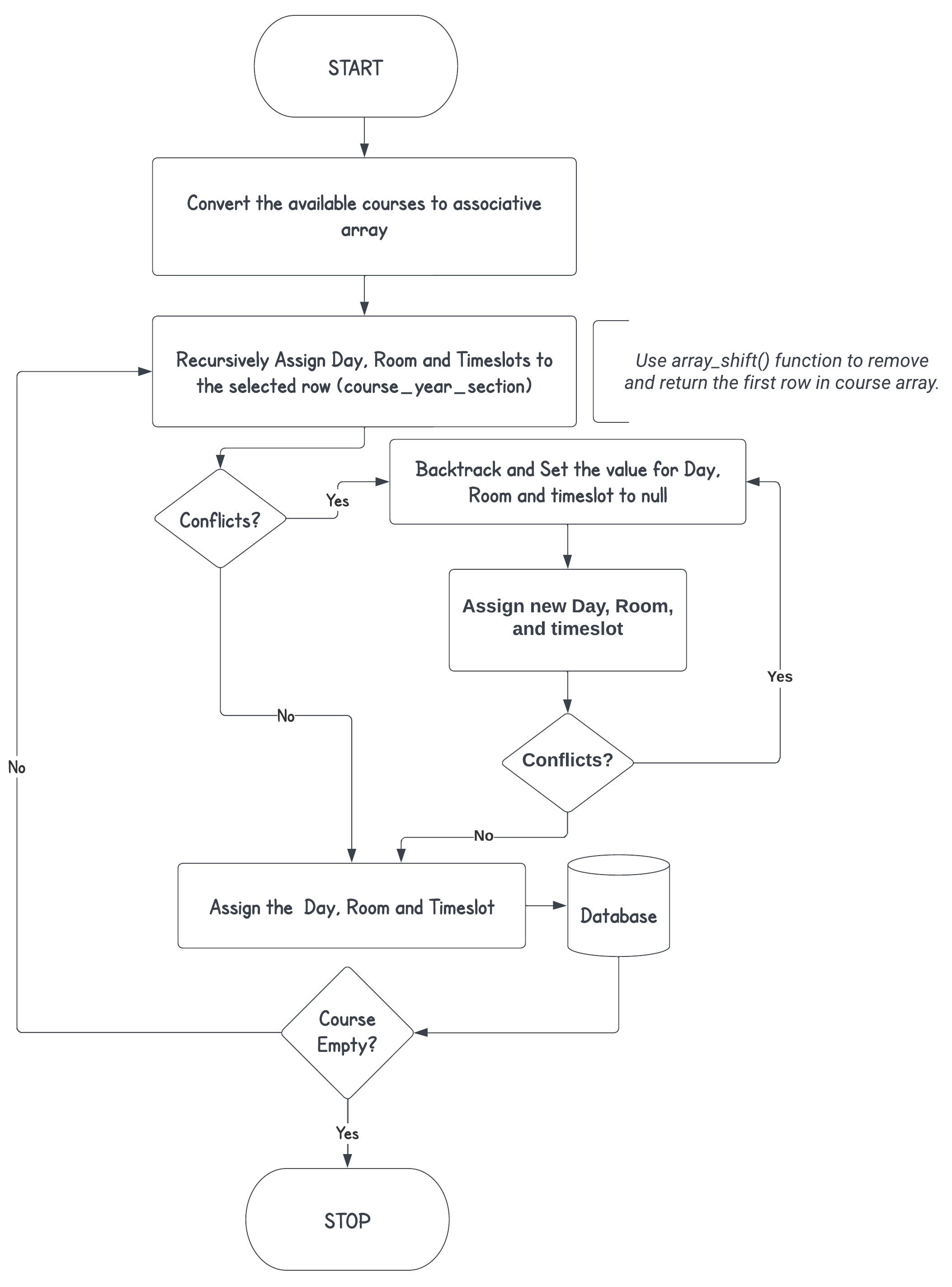
****

Figure 10: Backtracking Algorithm Implementation Flowchart

In Figure 10, the implementation of the backtracking algorithm in the ATS System is demonstrated. The developers followed a series of steps to implement the algorithm. They began by converting all available courses from the database into an associative array. Subsequently, the backtracking algorithm recursively assigned a unique timeslot to each selected course. Before inserting the assigned timeslots, a helper function was utilized to check for conflicts. In cases where overlaps occurred with other schedules, the algorithm backtracked, set the value to null, and attempted a different timeslot until a non-overlapping one was found. The algorithm continued this process recursively until all courses in the array were assigned a timeslot.

**Principles of Operation**

The researchers developed an Automated Timetable Scheduling System for Cavite State University CCAT Campus, especially designed for the Department of Computer Studies. The scheduling committees or teachers had to register an account first to log in to ATS. After successfully logging in, the user needed to add data for faculty loading to assign a subject to a specific teacher. After all the data was added to the faculty loading, the user needed to choose between Automated or Manual creation of the schedule. Then, the user had to wait approximately 1 to 3 minutes for automated generation. On the other hand, in the manual creation of the schedule, the user could assign specific days, timeslots, and rooms to the teachers listed in the faculty loading table. However, as of now, the manual generated schedule was not able to check if it conflicted with the automated schedules. Lastly, the ATS had a specific view page for room, teacher, and course schedules. Each teacher, room, and course had separate tables displaying their respective schedules. The generated schedule can be converted in pdf format for printing purposes.

**Testing and Evaluation**

The developers tested the following:

1. Reliability

The researchers tested the generated output(timetables) based on the following conditions:

* + - * Teachers, Students and Rooms must have a conflict-free generated schedules.
      * The timeslot duration must align with the subject hours.
      * The generated output should be free of conflicts.
      * The manual added schedule should not overlap with the existing schedule,
      * The room type should be corresponded to the subject type.

1. Functionality – in this testing the researchers performed alpha testing to test the following modules:

* Automated Generation Module

To test each module or tabs the researchers used the test case shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **OBJECTIVE** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **NUMBER OF DEBUGGED** |
|  |  |  |  |  |

* **Test Case ID –** this is the unique identification number and to identify what is being tested.
* **Objective –** this is the primary goal of the testing.
* **Expected Results –** this is the expected outcome of the test.
* **Actual Results –** it refers to the real outcome of the testing.
* **Number of Debugged -**it refers to the number of debug the tester made to achieve the objective.

**Data Analysis**

The researcher evaluated 40 respondents as shown below:

Table 1:  Evaluators description

|  |  |
| --- | --- |
| **EVALUATOR** | **NUMBER OF EVALUATOR** |
| Web Developers | 10 |
| Students | 30 |

The researchers computed the evaluation results by getting the total of each criterion then divided it by the total evaluators and interpret the total score using the table below:

Table 2. Descriptive Interpretation of total score

|  |  |
| --- | --- |
| **SCORE** | **INTERPRETATION** |
| 4.51 -5.00  3.51 - 4.50  2.51 - 2.50  1.51 - 2.50  1.00 - 1.50 | Excellent  Very Satisfactory  Satisfactory  Unsatisfactory  Needs Improvement |

**Implementation Plan or Deployment**

The implementation of the system follows the succeeding schedules:

1. Request for the campus approval of the system to be implemented in CVSU CCAT Campus.
2. Checking for the campus equipment capabilities to use the system.
3. Installation of the system prototype.
4. Testing of the system with large data for 2 weeks.
5. The results of the testing will be analyzed.
6. If the system passes the certain criteria, the system will be used, and the output of the system will be linked to the CVSU CCAT Campus Portal.

**RESULTS AND DISCUSSION**

In this chapter, the findings and discussions of the study are presented. The chapter encompasses the outcomes of the testing and evaluation and software design, offering a comprehensive analysis and interpretation of the data collected

**System Design**

Figure 11 shows the register page, where user can register account to login in ATS. The user needs to add unique email, name and password.

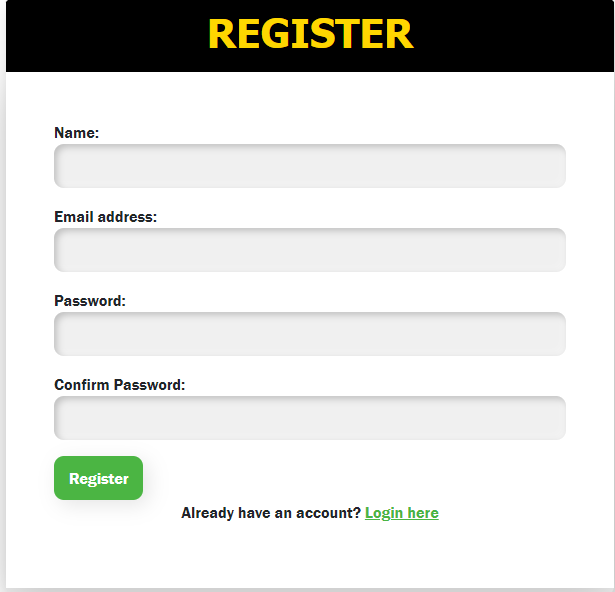
****

Figure 11: Register Page

Figure 12 shows the login page where the user needs to input the data, they used in registration page.

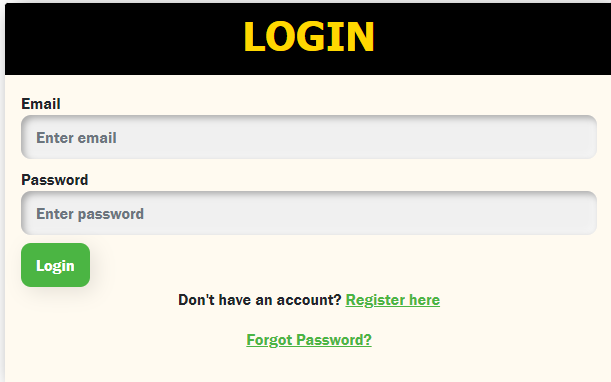


Figure 12 Log in Page

Figure 13 shows that the user already logged in and redirected to the dashboard that display the frequency of all data in ATS.

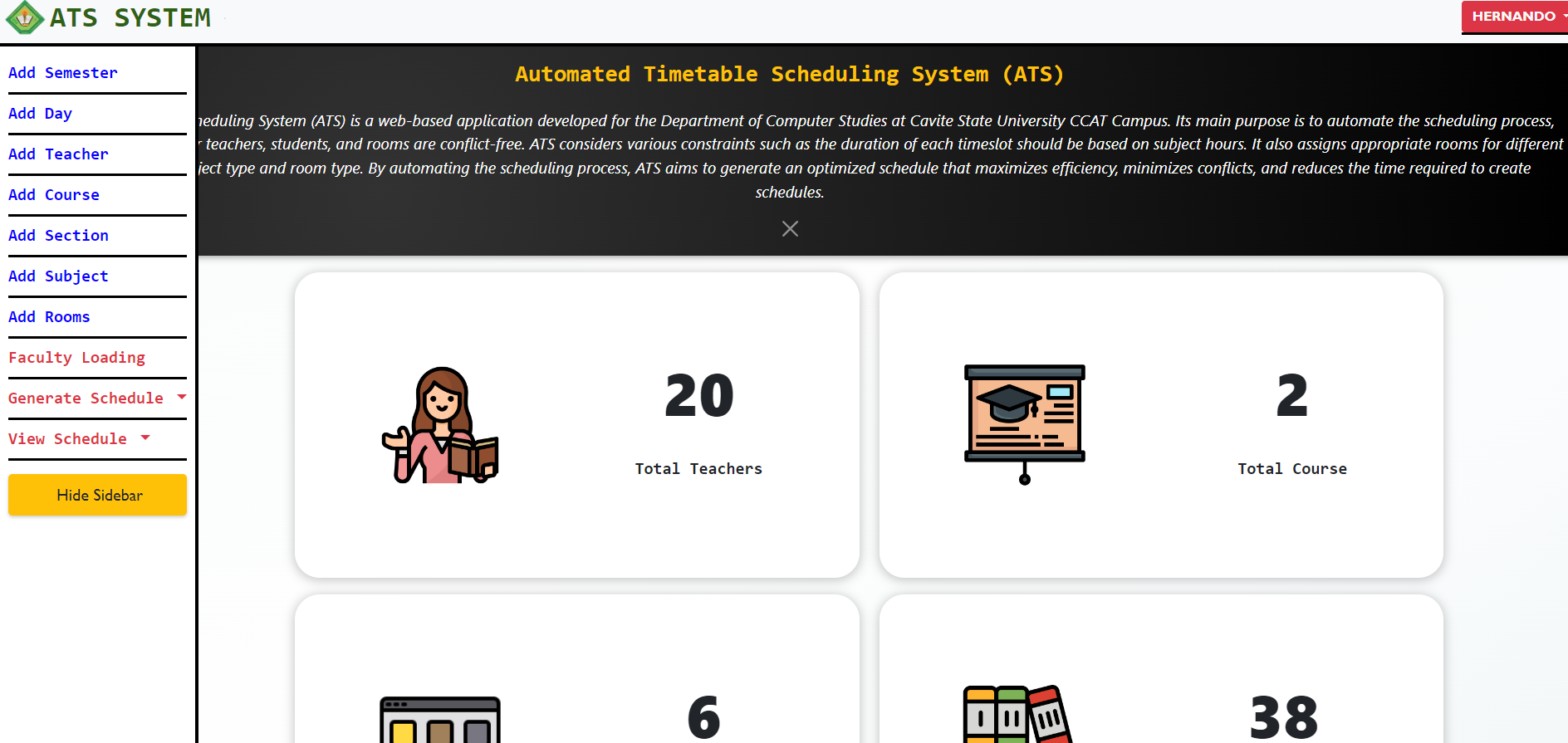


Figure 13: Admin Dashboard

Figure 14 shows the list of semesters, the user can add another semester by clicking the add semester button and the current semester can be updated and deleted.

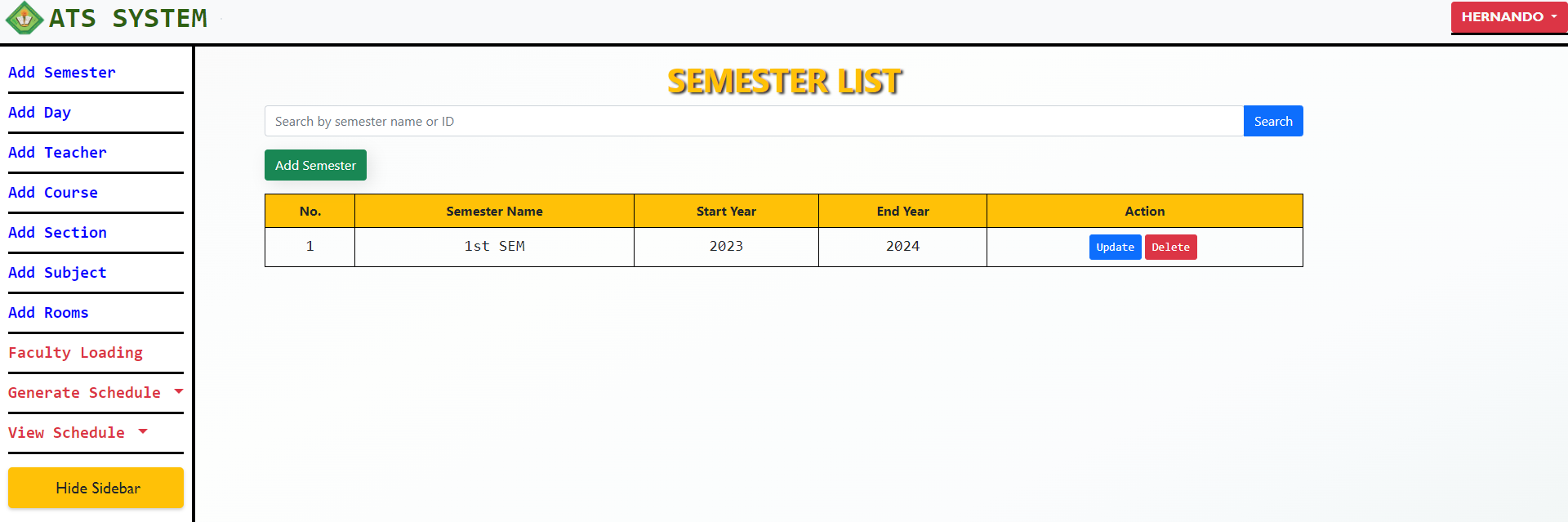


Figure 14: Semester List

Figure 15 shows the add semester page where user need to input semester id, semester name, start year and end year and then click the create button to save it to the database.

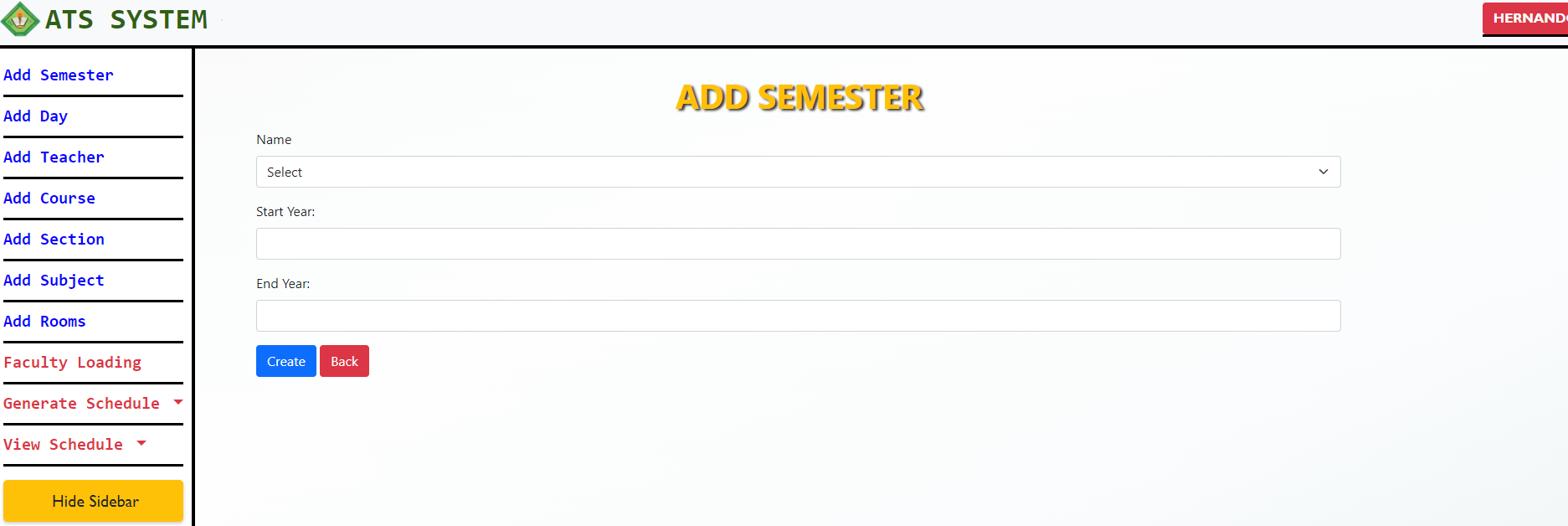


Figure 15: Adding Semester Page

Figure 16 shows the day list page where the user can see the available days for the current’s semester, the user can add, update and delete data.

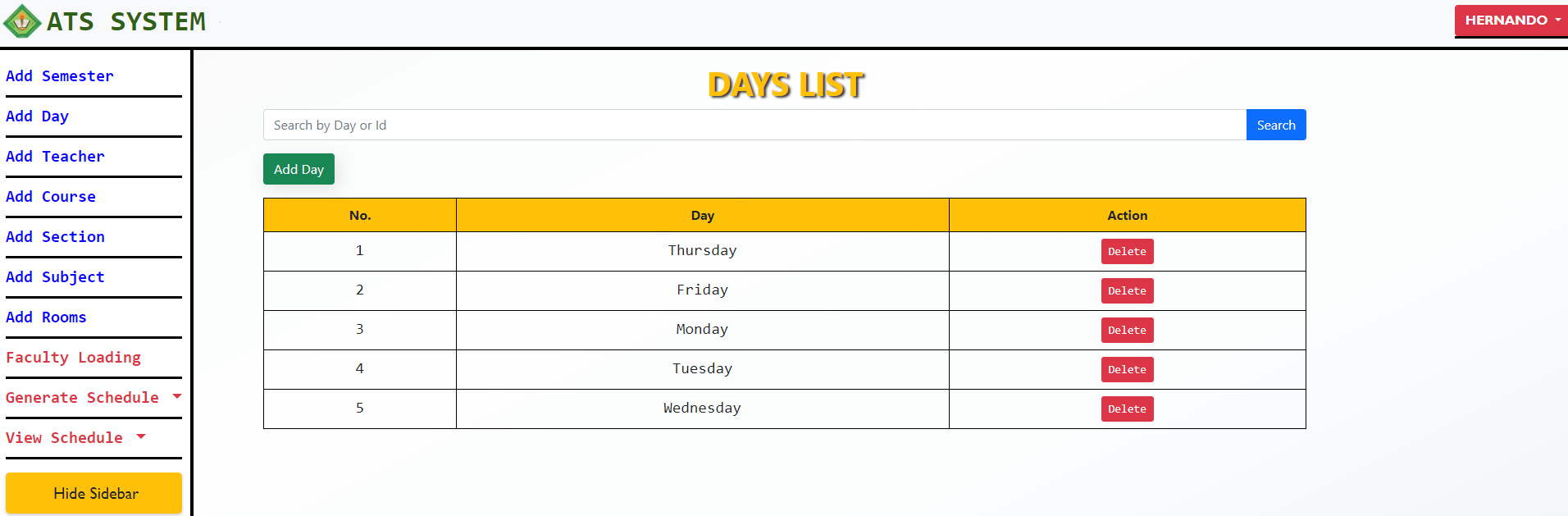


Figure 16: Day List

Figure 17 shows the page for adding new day, the user needs to select a day between Monday to Friday but if the day is already existing the system will not allow to add the day.

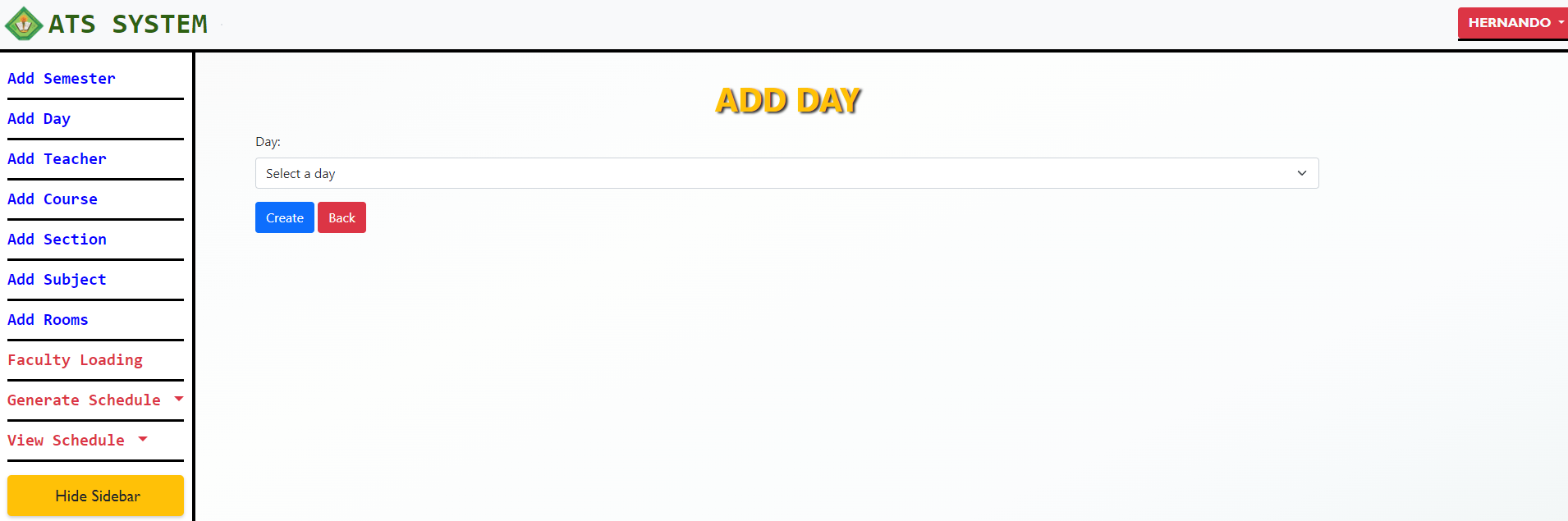


Figure 17: Adding Day Page

Figure 18 shows the list of available teachers for faculty loading. The user can add, update and delete teachers’ data but if the teacher is already existing in the list the system will not allow to add new teacher.

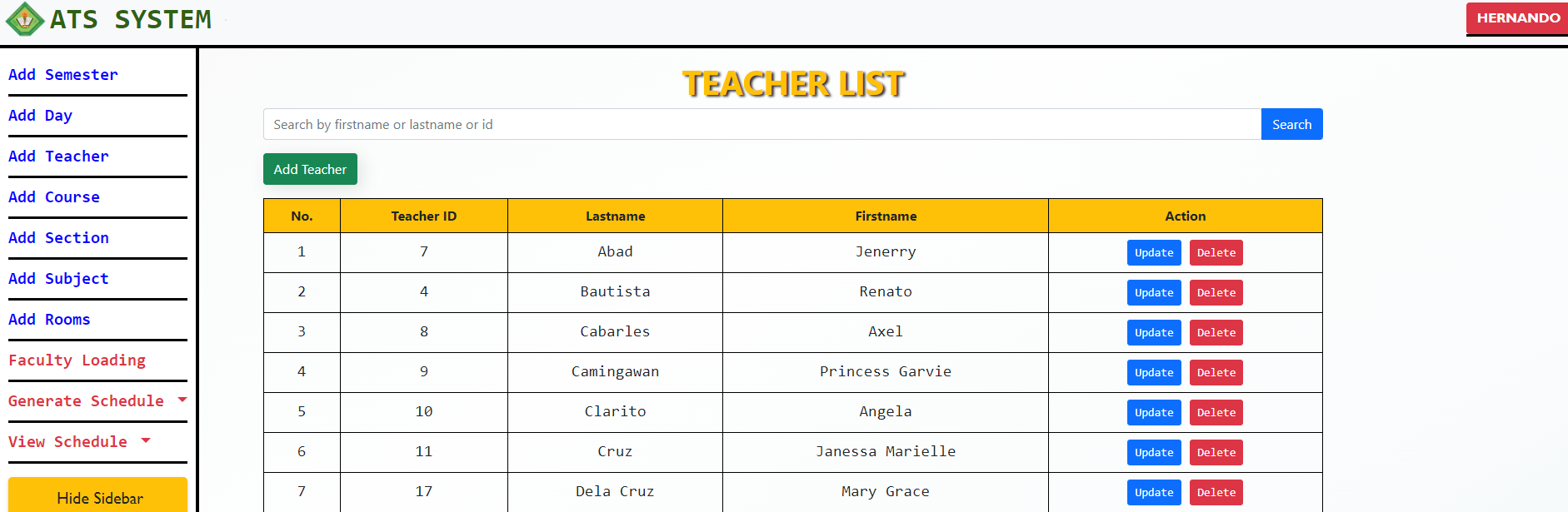


Figure 18: Teacher List

Figure 19 shows the page for adding new teacher, the user needs to input a unique teacher id and the teacher’s name but if the teacher id is already existing the system will not allow the insertion of new teacher.

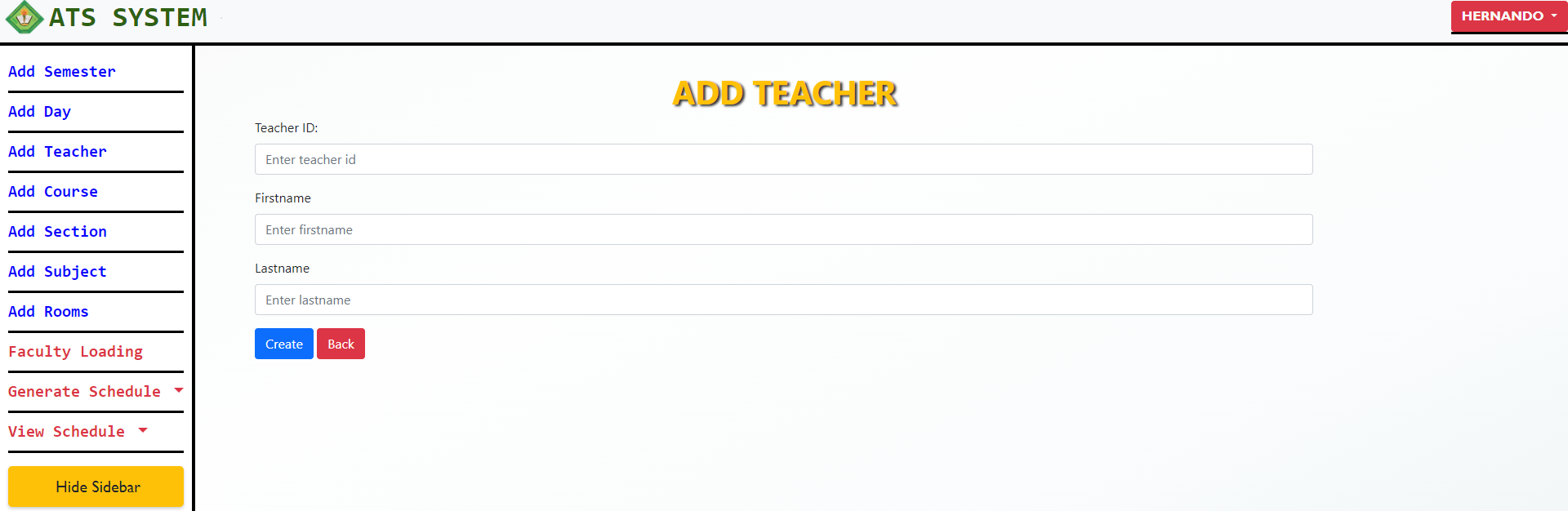


Figure 19: Adding Teacher Page

Figure 20 shows the list of available courses for the semester. The user can add new course to the list or update and delete the existing courses.

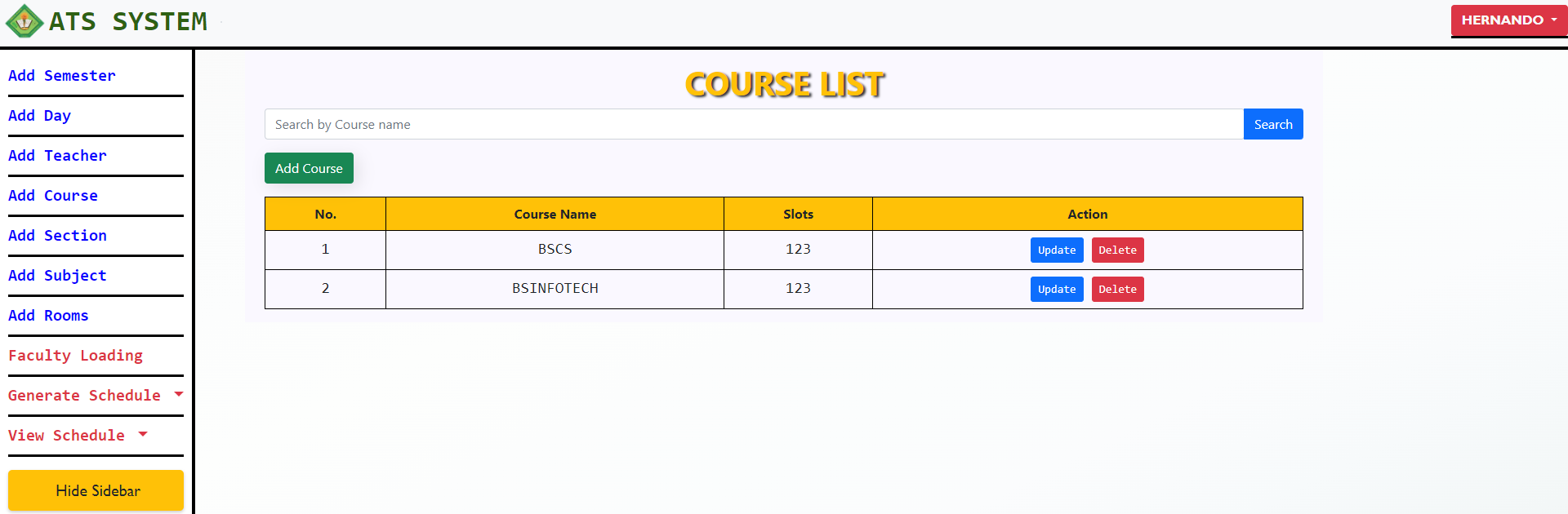


Figure 20: Course List

Figure 21 shows the page for adding courses, the user needs to add course name, slots and a unique id but if the course id is existing it will block by the system.

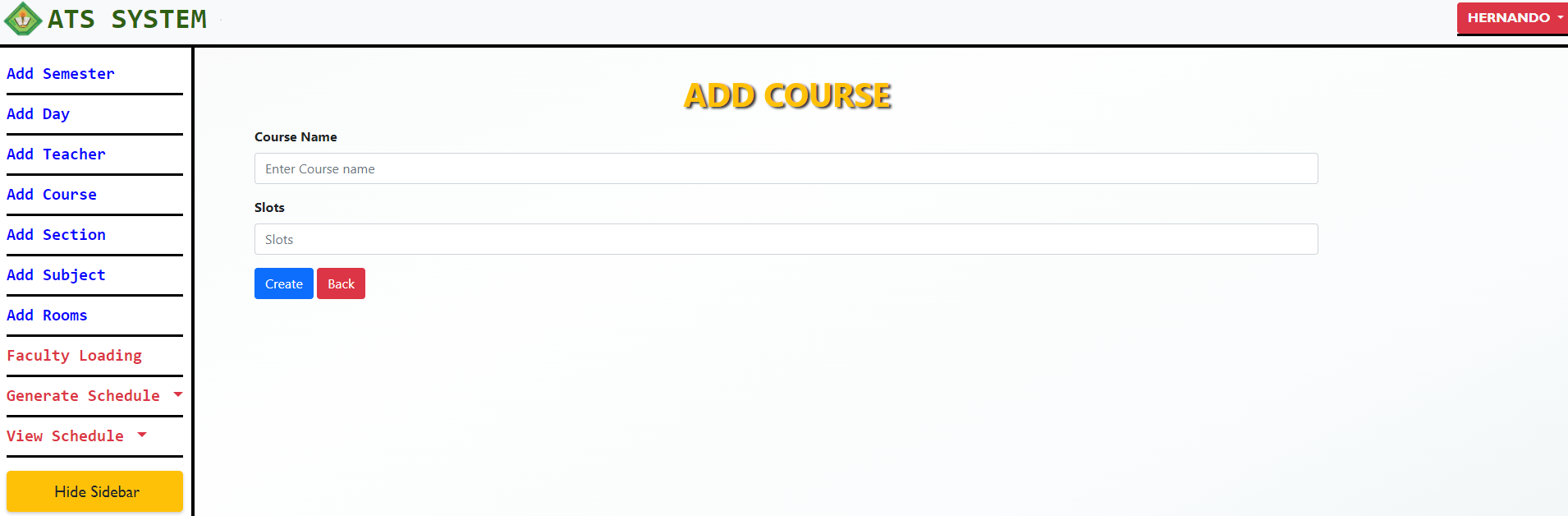


Figure 21: Adding Course Page

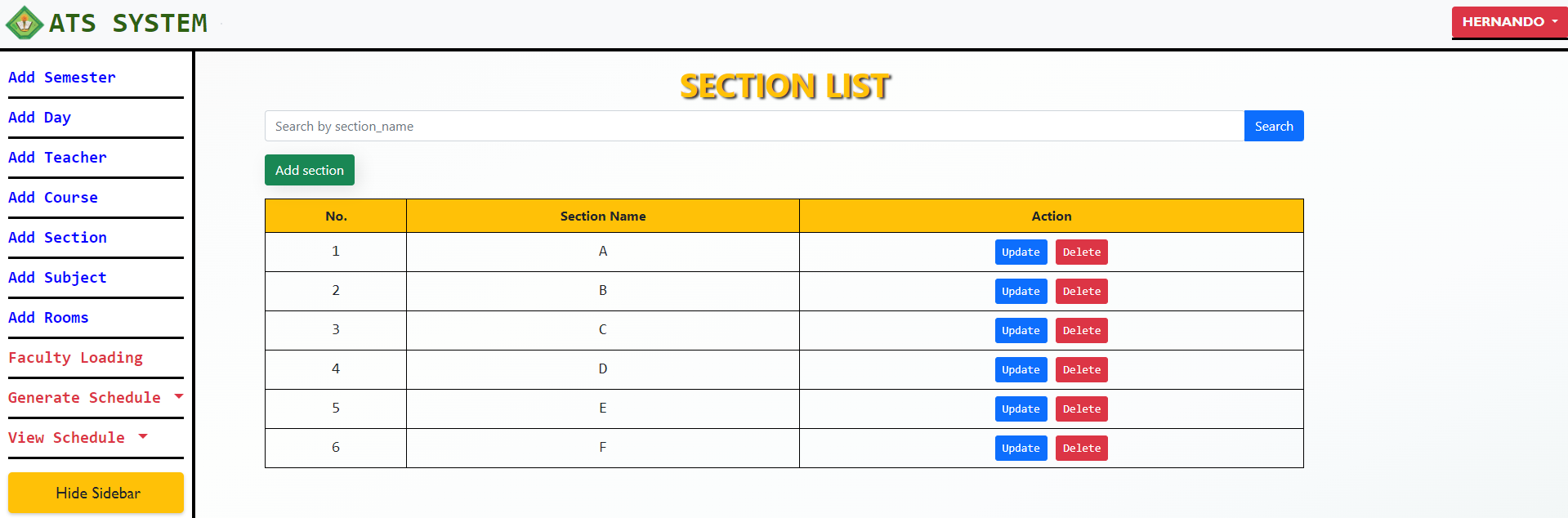
Figure 22 shows the list of available sections for the semester. The user can add new section to the list or update and delete the existing sections. 

Figure 22: Section List

Figure 23 shows the page for adding sections, the user needs to add section name and a unique id but if the section id is existing the system will prompt section already exist.

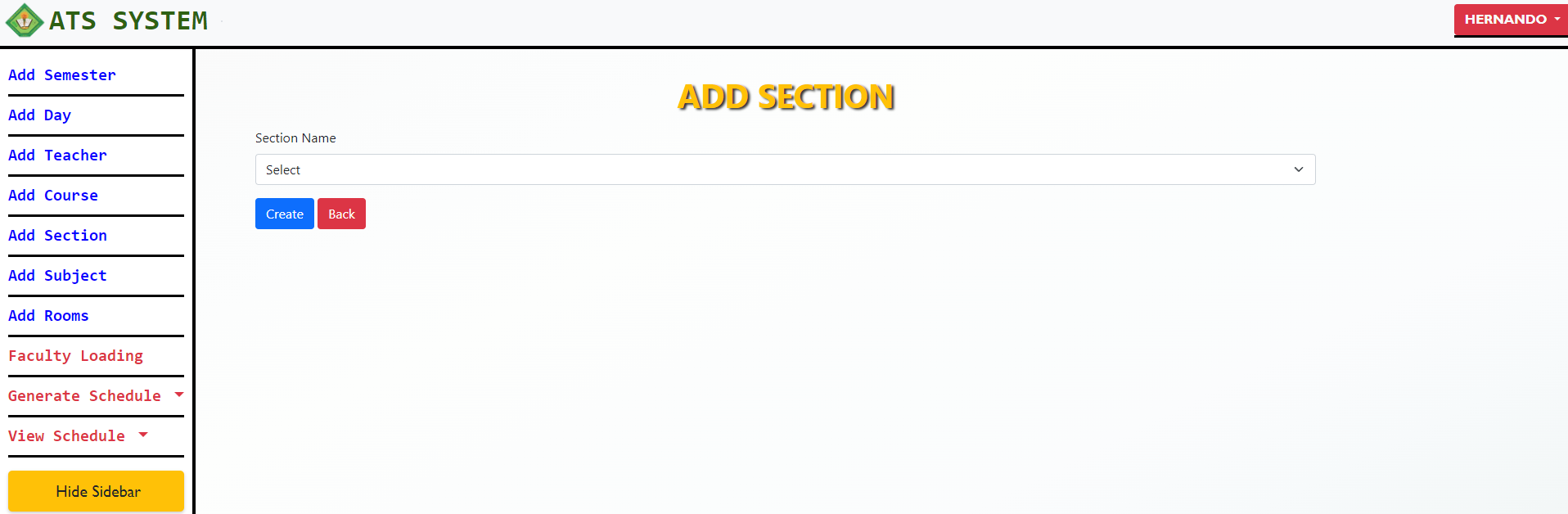


Figure 23: Adding Section Page

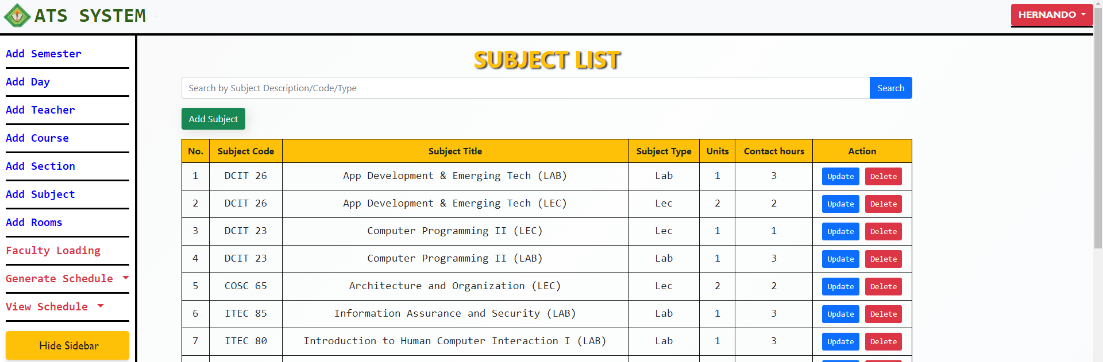
 Figure 24 shows the list of available subjects for the semester. The user can add new subjects to the list or update and delete the existing subjects.

Figure 24: Subject List

Figure 25 shows the page for adding subject, the user needs to add subject name, subject title, subject type, units and subject hours but if the subject is existing the system will prompt subject already exist.

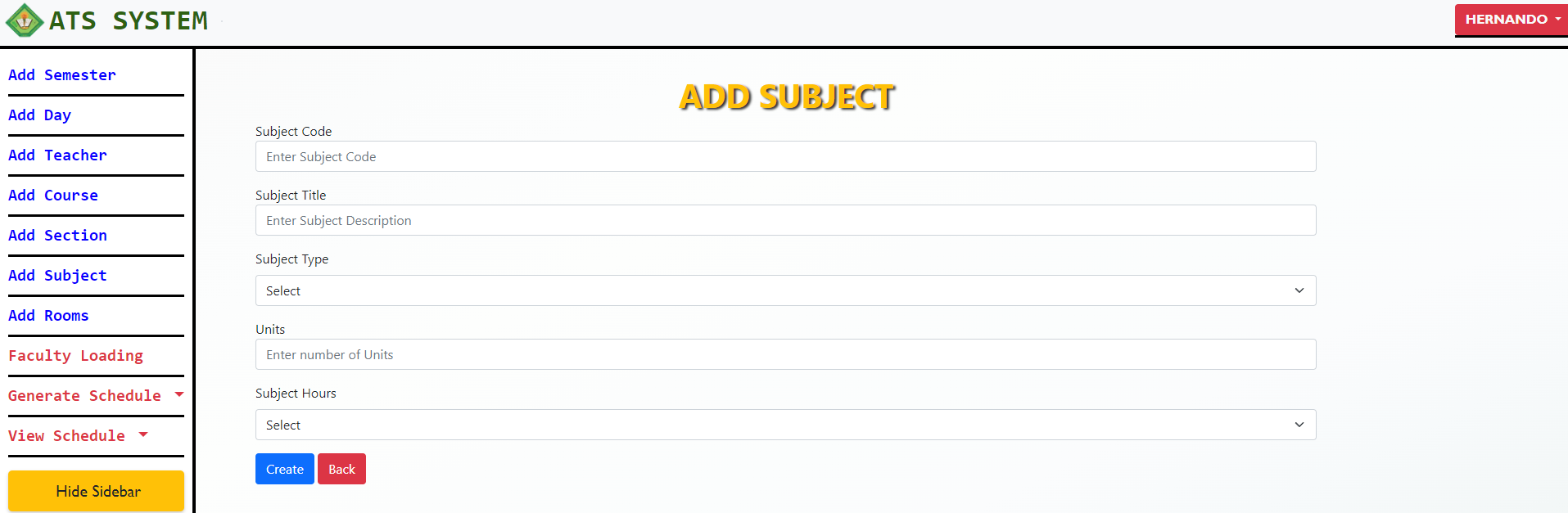


Figure 25: Adding Subject Page

Figure 26 shows the list of available room for the semester. The user can add new rooms to the list or update and delete the existing rooms.



Figure 26: Room List

Figure 27 shows the page for adding rooms, the user needs to add room name, room type and capacity but if the room is existing the system will prompt room already exist.

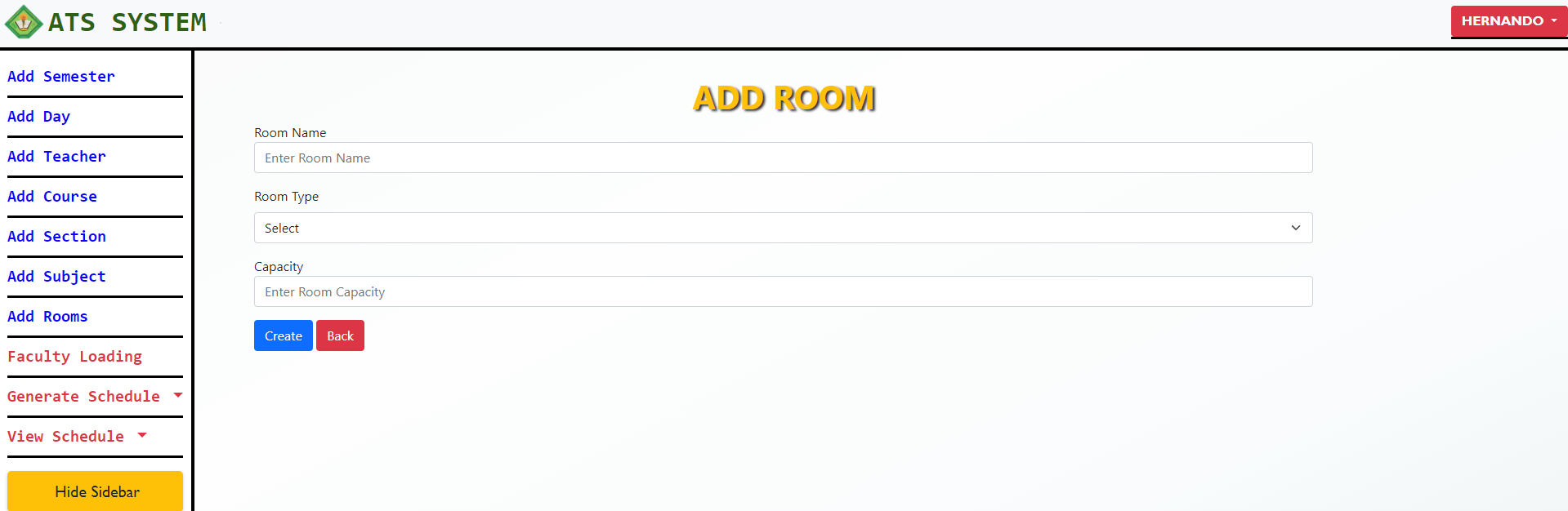


Figure 27: Adding Room Page

Figure 28 shows the list of assigned subject to the teacher in the faculty loading page. The user can add assign new subject or update and delete the existing data.

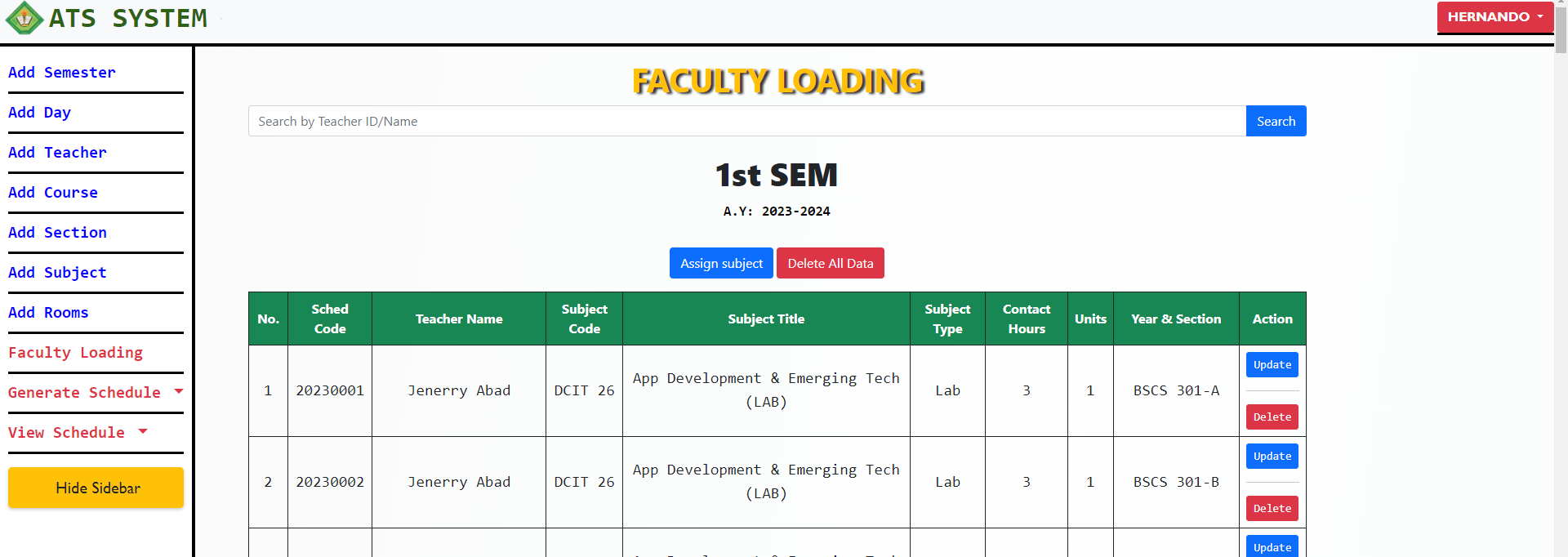


Figure 28: Faculty Loading List

Figure 29 shows the page for assigning a subject to a teacher. If the selected subject was already assigned the system will not allow the user to assign that subject.

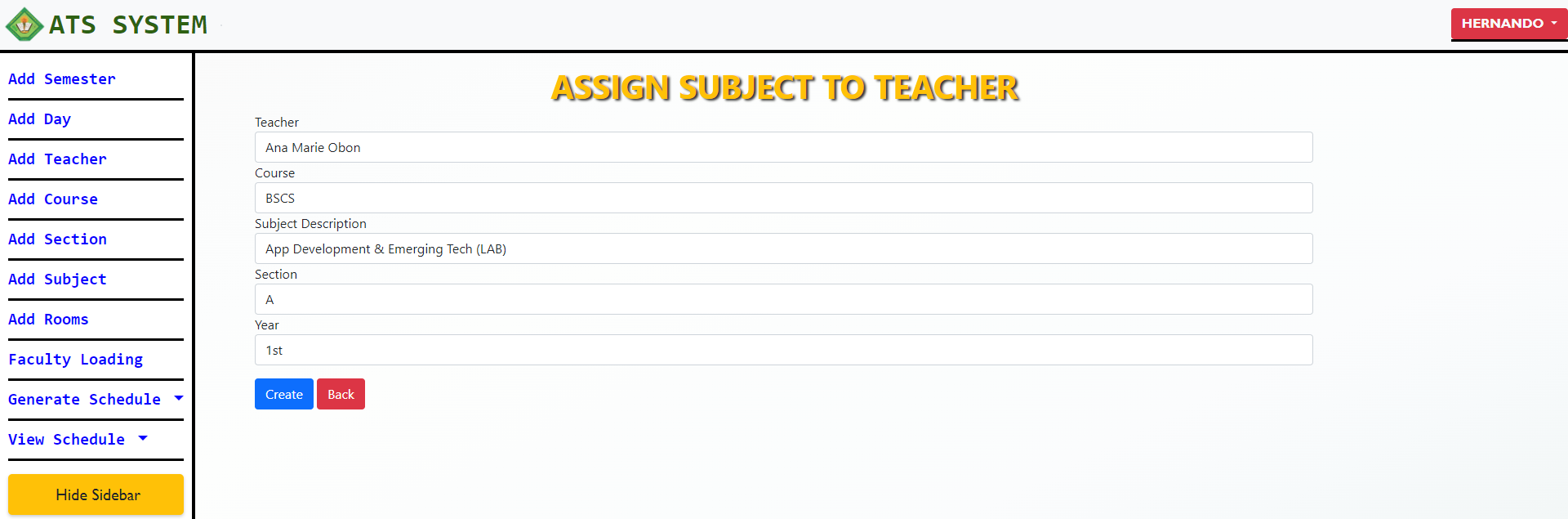


Figure 29: Adding Faculty Loading Page

Figure 30 shows the page for the automated generation page, where the user will click the generate button and wait approximate 1 to 5 minutes to generate the schedules.

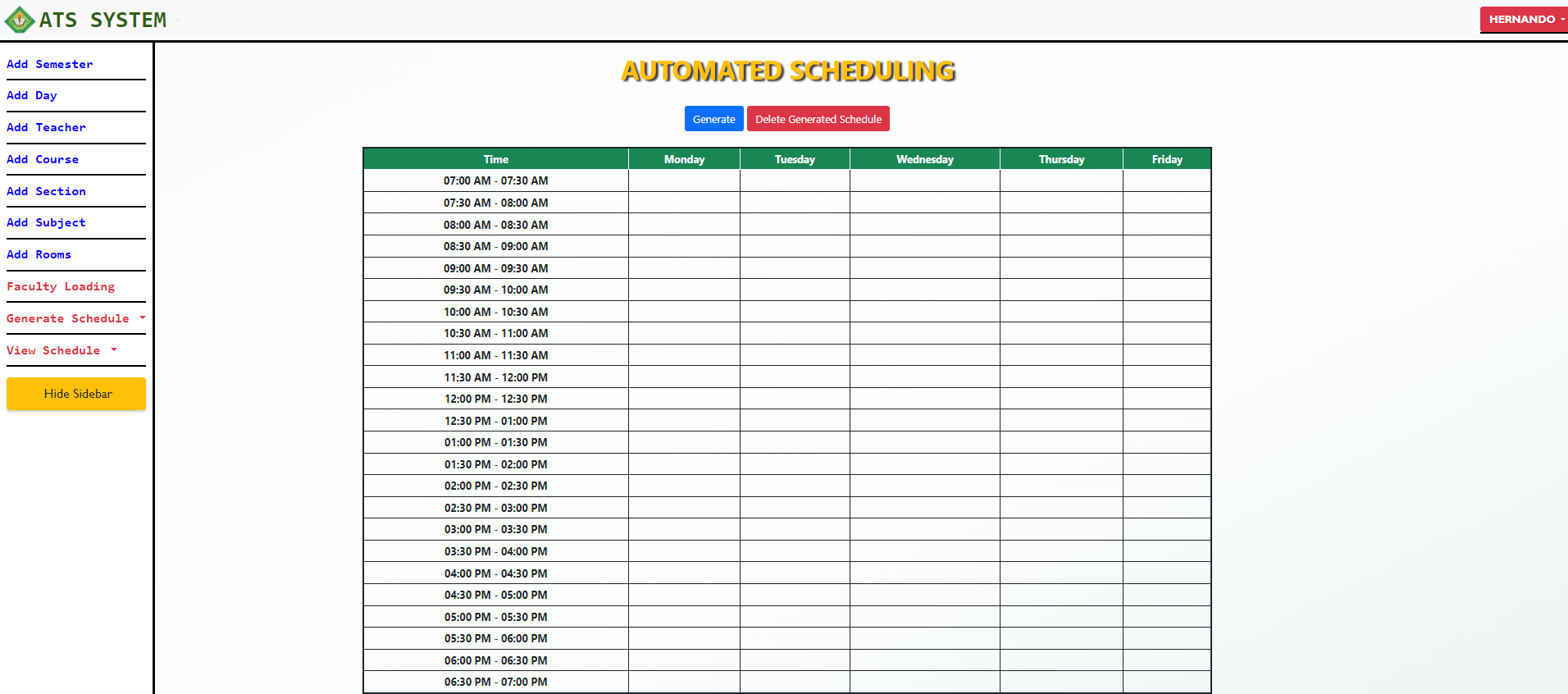


Figure 30: Automated Timetable Page

Figure 31 shows the generated schedules, the user can regenerate the schedules by clicking the regenerate button and user can also delete the generated schedules using delete button.

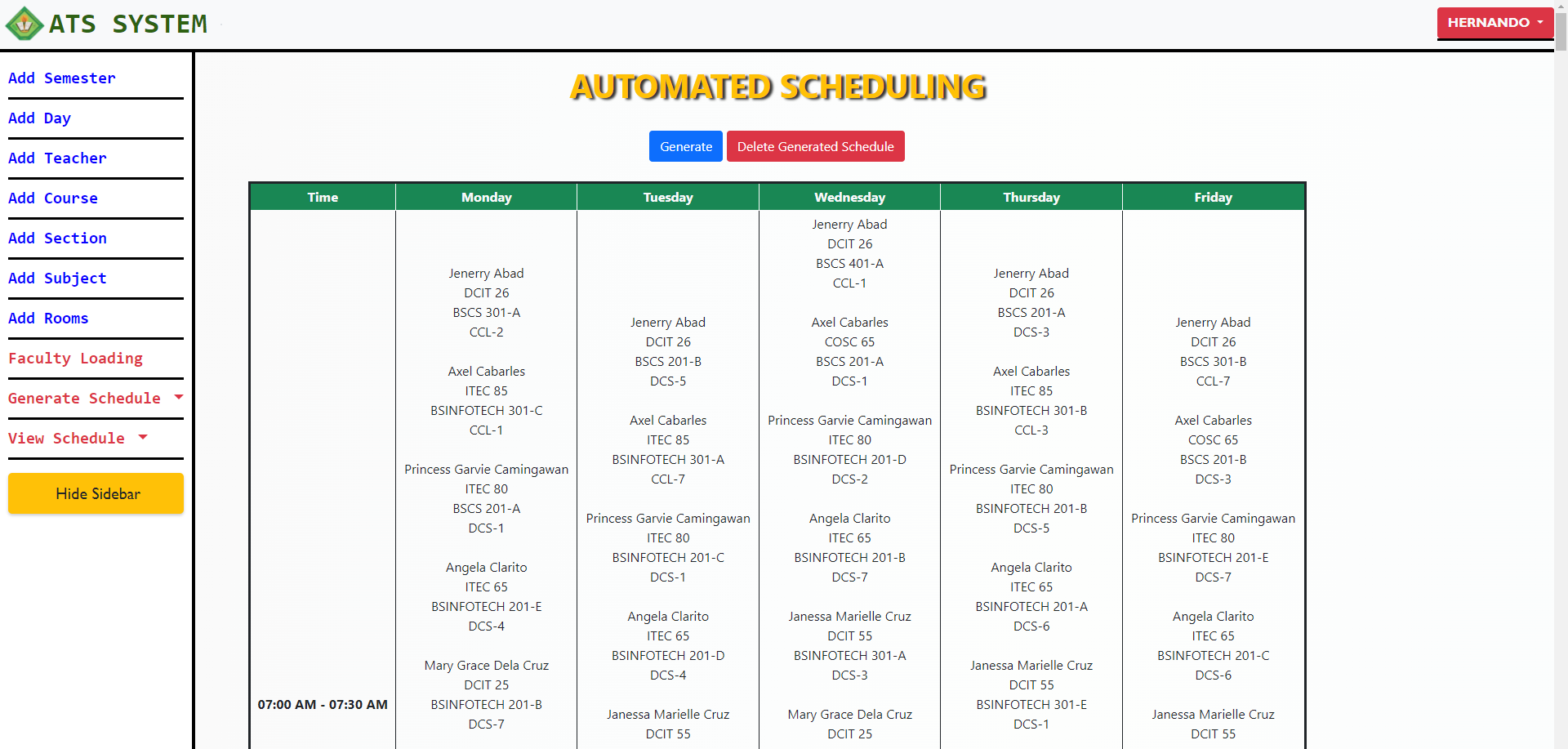


Figure 31: Automated Timetable Page with Generated Schedule

Figure 32 shows the page for adding a manual schedule to the generated schedules. The user must manually check if the teacher, subject, course, year and section, day, start time and end time is not overlap with the generated schedules. If the user will add timeslot that was already assigned the system will not allow the schedule to be added to the generated schedules.

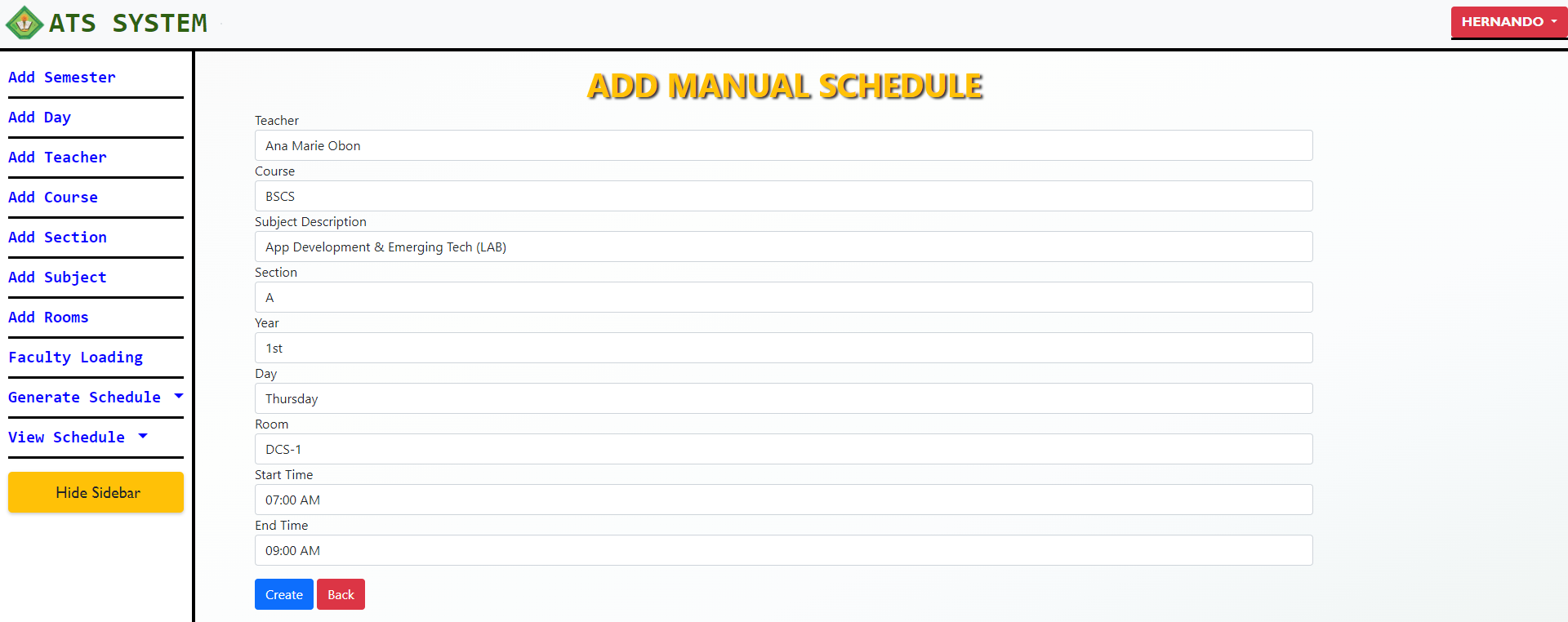


Figure 32: Adding of Manual Schedule

Figure 33 shows the generated room schedule, the user can see the separated schedules for room in table format.

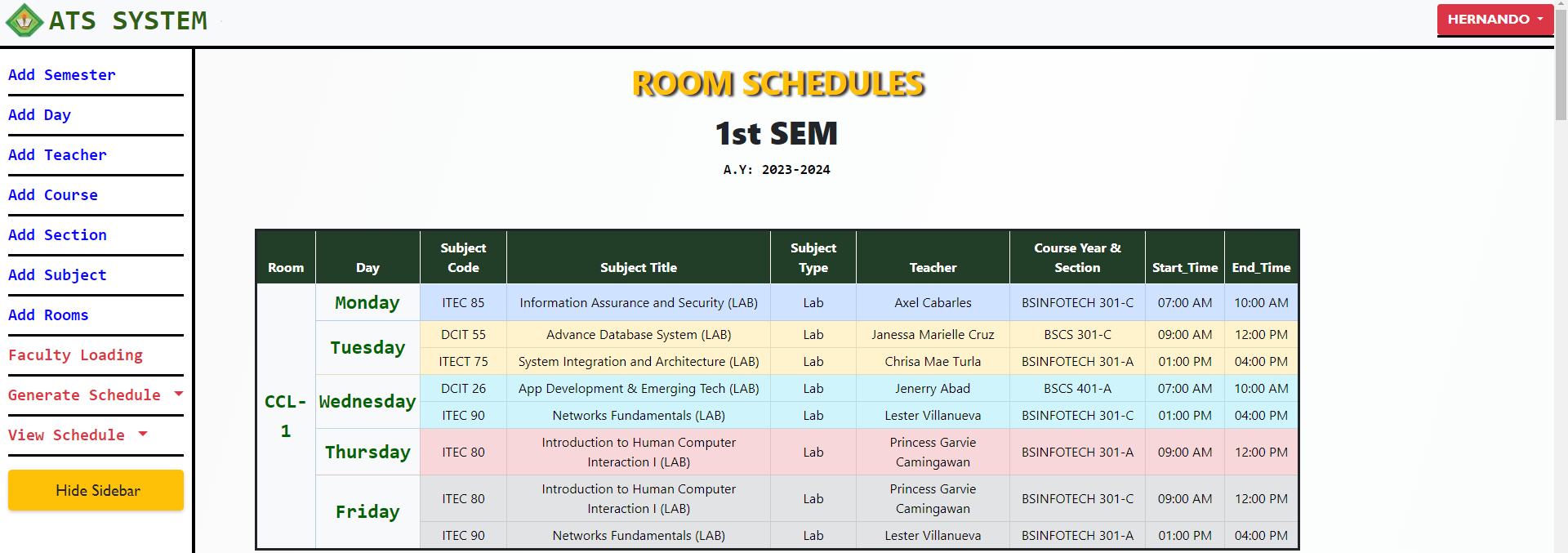


Figure 33: Automated Room Schedules

Figure 34 shows the pdf format of the generated schedule, the user needs to click the CTR+P for printing the pdf generated schedules.

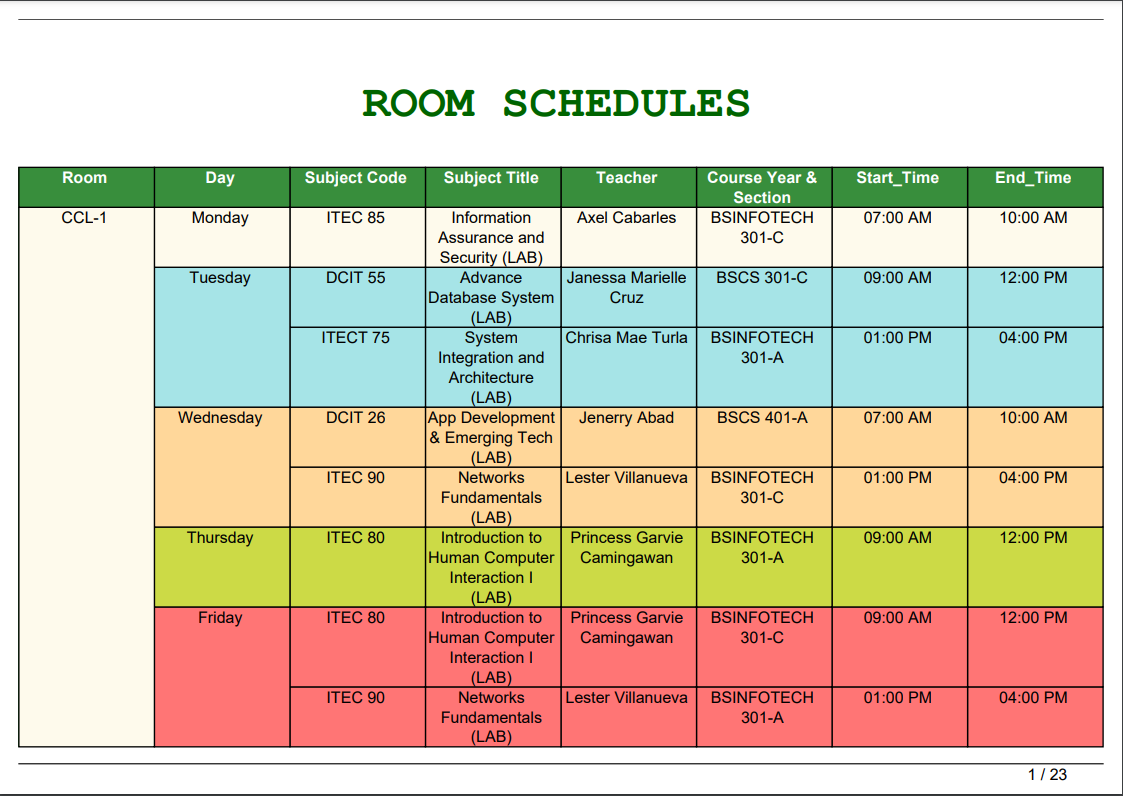


Figure 34: PDF Schedule for Room

Figure 35 shows the generated teacher schedules in separate tables. The user can convert the generated schedule in pdf format by clicking CTRL + P.

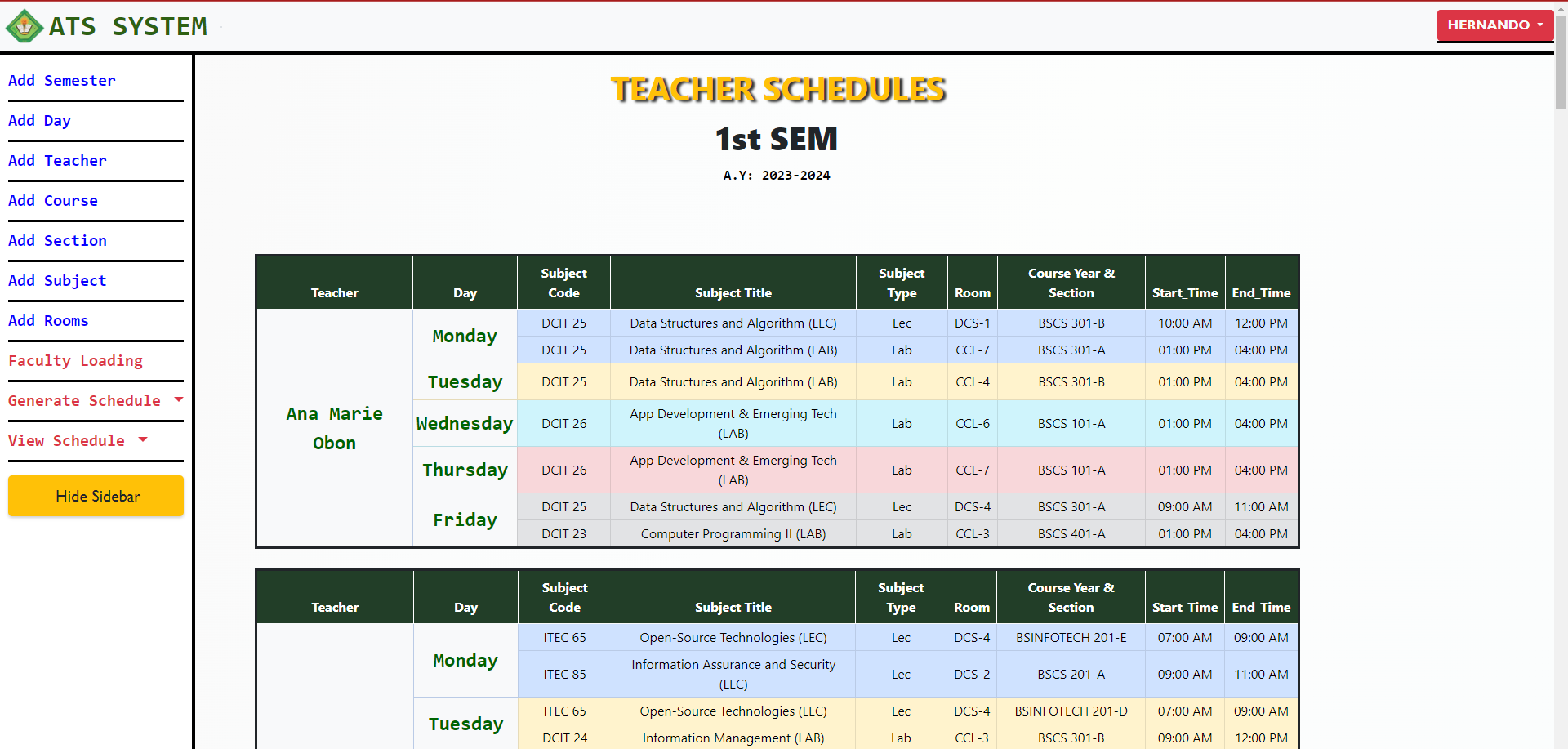


Figure 35: Automated Teachers Schedule

Figure 36 shows the pdf format of the generated schedules for teachers. The user can print the generated pdf schedules.

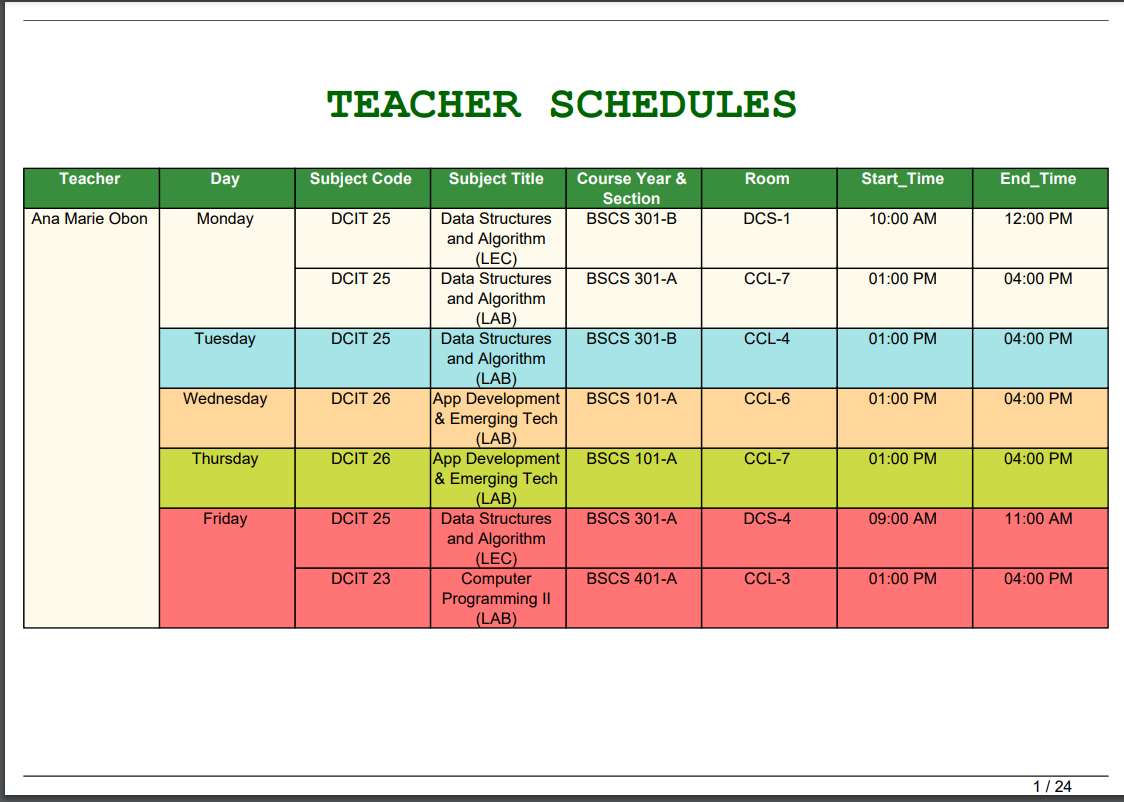


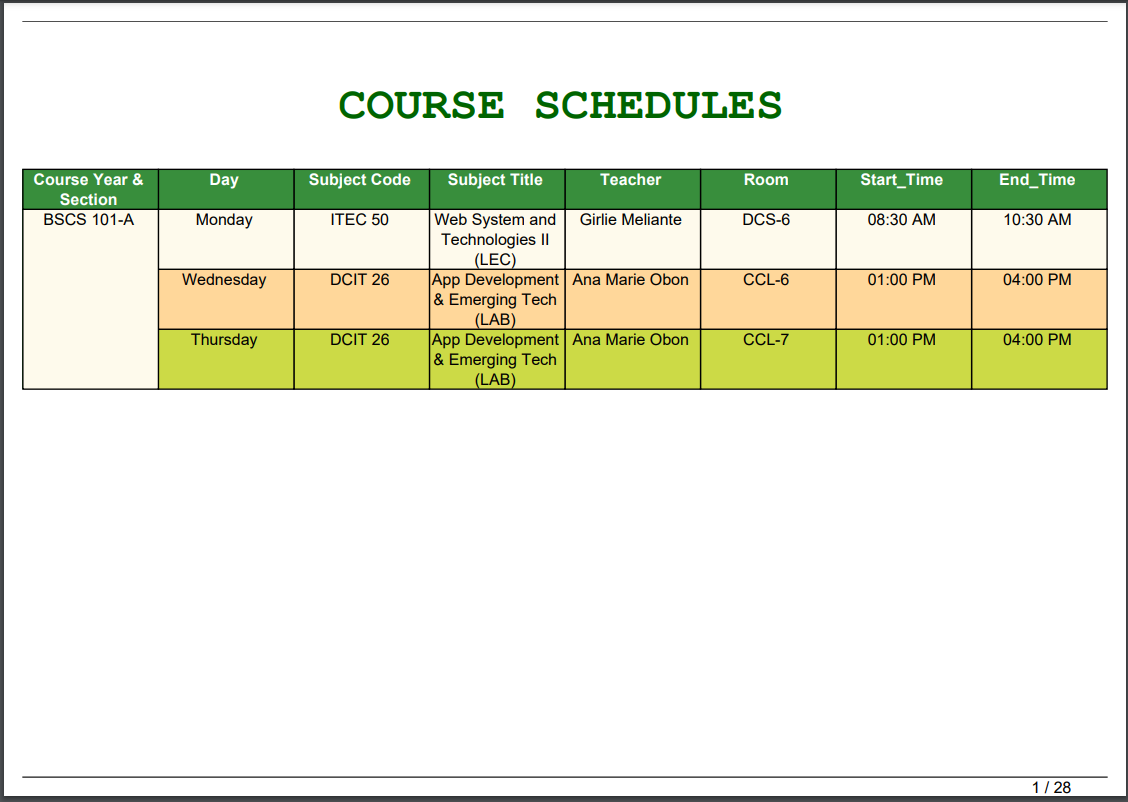
Figure 36: PDF Teachers Schedule

Figure 37 shows the course generated schedules in a separate table. The user can also convert it into pdf format.



Figure 37: PDF Automated Schedule for Course

Figure 38 shows the converted PDF format of course generated schedules. It was used for printing purposes.

Figure 38: PDF Schedule for Course

**Testing Results**

The developers conducted the test to ensure the reliability of the generated schedules to make sure that the criteria were met such as the teacher, rooms and courses must have a conflict-free schedules.

Table 4 shows the testing conditions to check the generated output and the testing results meet the condition number 1 to 5 while the last condition was failed due to some rooms that have a subject type of Lecture assigned in lab rooms.

**Table 4. Testing the reliability of generated outputs.**

|  |  |
| --- | --- |
| **TESTING CONDITION** | **REMARK** |
| 1. **Teachers, Students and Rooms must have a conflict-free generated schedules.** | Passed |
| 1. **The timeslot duration must align with the subject hours.** | Passed |
| 1. **The generated output should be free of conflicts.** | Passed |
| 1. **The Manual added schedule should not overlap with the existing schedules.** | Passed |
| 1. **The room type should be corresponded to the subject type.** | Passed |

Table 5 shows the results for system's CRUD functionality, which has been successfully executed. The test results validate the proper functioning of the CRUD operations and their alignment with the expected requirements.

**Table 5. Testing done by the developers to test the CRUD functionality of system.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **OBJECTIVE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **NUMBER OF DEBUGGED** |
| T-001 | Test the CRUD functionality for Semester | Users should be able to create, view, update, and delete semesters. | Users have successfully handled semester management, including creation, viewing, updating, and deletion. | 1 |
| T-002 | Test the CRUD functionality for Day | Users should be able to create, view, update, and delete Day. | Users have effectively performed day management tasks, which encompassed creating, viewing, updating, and deleting operations. | 1 |
| T-003 | Test the CRUD functionality for Teacher | Users should be able to create, view, update, and delete teacher. | Users have successfully managed teachers’ details including creation, viewing, updating, and deletion. | 1 |
| T-004 | Test the CRUD functionality for Course | Users should be able to create, view, update, and delete course. | Users have successfully handled course management, including creation, viewing, updating, and deletion. | 1 |
| T-005 | Test the CRUD functionality for Section | Users should be able to create, view, update, and delete section. | Users have successfully handled section management, including creation, viewing, updating, and deletion. | 1 |
| T-006 | Test the CRUD functionality for Subject | Users should be able to create, view, update, and delete subject. | Users have successfully handled subject management, including creation, viewing, updating, and deletion. | 1 |
| T-007 | Test the CRUD functionality for Room | Users should be able to create, view, update, and delete room. | Users efficiently managed room operations, covering creation, viewing, updating, and deletion. | 1 |

Table 6. shows that the system effectively blocked users with unregistered accounts from accessing the system. Furthermore, it was observed that registering a new account requires the use of a unique email address.

**Table 6. Testing done by the developers to test the register and login module.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **OBJECTIVE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **NUMBER OF DEBUGGED** |
| T-008 | Test the login module | The system must not allow to login the unregistered user, | Unregistered login was blocked the system. | 1 |
| T-009 | Test the register module | If the user registers a new account with a same email, The system must not allow the registration of the account, | Same email used in registration process was blocked by the system. | 1 |

Table 7 illustrates the results for testing the faculty loading module. Based on the table results the teacher with a same assigned subject to a same course was block by the user and display an error prompt.

**Table 7. Testing done by the developers to test the faculty loading module**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **OBJECTIVE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **NUMBER OF DEBUGGED** |
| T-010 | Test faculty loading module | The same teacher should not have a same assigned subject with same sections. | The system prompts a message “The subject was already assigned with the same teacher!” | 1 |

**Evaluation Results**

The system's performance was assessed using the ISO-IEC 25010. evaluation criteria, encompassing functional suitability, performance efficiency, usability, security, maintainability, and portability. A total of 40 individuals participated in the evaluation process, including 10 IT experts and 30 end users.

***Narrative Discussion of the Performance Evaluation of the System based on IT Experts***

Table 8 reflects the assessment of the IT expert respondents in the functional sustainability of the system when it comes to Functional Completeness, Functional Correctness, Functional Appropriateness. The indicator was rated as follows; Functional Completeness received, 3.7; Functional Correctness, 4.3 and Functional Appropriateness, 4.5. The mean of 4.1 indicates that the system is functioned well, provides correct results with precision and facilitates the accomplishment of the tasks and objectives.

**Table 8. Perception of IT Experts in the functional suitability of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Functional Completeness | 3.7 | Satisfactory |
| 1. Functional Correctness | 4.3 | Very Satisfactory |
| 1. Functional Appropriateness | 4.5 | Excellent |
| Average Mean | 4.1 | Very Satisfactory |

Table 9 reflects the assessment of the IT expert respondents in the performance efficiency of the system when it comes to Time Behavior, resource utilization, capacity. The indicator was rated as follows; Time behavior received, 4.5; Resource Utilization, 4.6 and Capacity, 4.4. The mean of 4.56 indicates that the system performed well as it met the specified requirements of the system.

**Table 9. Perception of IT Experts in the Performance Efficiency of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Time behavior | 4..5 | Excellent |
| 1. Resource utilization | 4.6 | Excellent |
| 1. Capacity | 4.6 | Excellent |
| Average Mean | 4.56 | Excellent |

Table 10 reflects the assessment of the IT Expert respondents in the Appropriateness Recognizability, Learnability, Operability, User Error Protection and User Interface Aesthetics of the system. Appropriateness Recognizability received, 4.9, Learnability received 4.5, Operability received 4.5, User Error Protection received 4.4, User Interface Aesthetics received 3.8 and Accessibility received 4.4. This result indicates that the system is Very Satisfactory in exchanging information and make use of information.

**Table 10. Perception of IT Experts in the Usability of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Appropriateness Recognizability | 4.5 | Excellent |
| 1. Learnability | 4.5 | Excellent |
| 1. Operability | 4.5 | Excellent |
| 1. User Error Protection | 4.4 | Satisfactory |
| 1. User Interface Aesthetics | 3.8 | Very Satisfactory |
| 1. Accessibility | 4.4 | Very Satisfactory |
| Average Mean | 4.35 | Very Satisfactory |

Table 11 reflects the assessment of the IT Expert respondents in the security of the system when it comes confidentiality and integrity, The indicator was rated as follows, confidentiality received, 4.2; and integrity received, 3.9. The mean of 4.05 indicates very satisfactory, and the system can provide strong security for the user.

**Table 11. Perception of IT Experts in the Security of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Confidentiality | 4.2 | Very Satisfactory |
| 1. Integrity | 3.9 | Satisfactory |
| Average Mean | 4.05 | Very Satisfactory |

Table 12 shows system maintainability evaluation results with ratings for each indicator: modularity 4.3, reusability 4.4, analyzability 4.2, modifiability 4.3, and testability 4. The mean average of 4.24 indicates high user satisfaction and ease of modification and maintenance.

**Table 12. Perception of IT Experts in the Maintainability of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Modularity | 4.3 | Very Satisfactory |
| 1. Reusability | 4.4 | Very Satisfactory |
| 1. Analyzability | 4.2 | Very Satisfactory |
| 1. Modifiability | 4.3 | Very Satisfactory |
| 1. Testability | 4 | Very Satisfactory |
| Average Mean | 4.24 | Very Satisfactory |

Table 13 reflects the assessment of the IT Expert respondents in the portability of the system when it comes Adaptability. The indicator was rated as follows, Adaptability received, 3.6. The mean of 3.6 indicates satisfactory, and the system can provide enough portability for the user.

**Table 13. Perception of IT Experts in the Portability of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Adaptability | 3.6 | Satisfactory |
| Average Mean | 3.6 | Satisfactory |

The summary of assessments provided by the IT Experts is presented in Table 14 showcasing the ratings for various aspects. Functional Suitability achieved a score of 4.16, Performance received 4.56, Performance Efficiency received 4.56, Usability obtained 4.35, Security received 4.05, Maintainability achieved 4.24, and lastly, the portability of the system received a rating of 3.6. The average mean of these scores is 4.16, indicating a Very Satisfactory performance overall

**Table 14. Summary of IT Experts Evaluation Results**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Functional Suitability | 4.16 | Very Satisfactory |
| 1. Performance Efficiency | 4.56 | Very Satisfactory |
| 1. Usability | 4.35 | Very Satisfactory |
| 1. Security | 4.05 | Very Satisfactory |
| 1. Maintainability | 4.24 | Very Satisfactory |
| 1. Portability | 3.6 | Satisfactory |
| Average Mean | 4.16 | Very Satisfactory |

***Narrative Discussion of the Performance Evaluation of the System based on End Users***

The evaluation of the system involved 30 participants, comprising students from Department of Computer Studies. These respondents assessed the system's usability, effectiveness, efficiency, satisfaction, and coverage of relevant contexts.

Table 15 reflects the assessment of the end user respondents in the usability of the system when it comes to appropriateness recognizability, learnability, operability, user error protection, user interface aesthetics. The indicator was rated as follows; appropriateness recognizability received, 4.7; learnability, 4.6; operability, 4.7; user error protection, 4.4; user error protection, 4.4; and User Interface Aesthetics, 4.5. The mean of 4.58 indicates that the system is significantly user-friendly.

**Table 15 Perception of End-users in the Usability of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Appropriateness Recognizability | 4.7 | Excellent |
| 1. Learnability | 4.6 | Excellent |
| 1. Operability | 4.7 | Excellent |
| 1. User Error Protection | 4.4 | Very Satisfactory |
| 1. User Interface Aesthetics | 4.5 | Very Satisfactory |
| Average Mean | 4.58 | Excellent |

Table 16 reflects the assessment of the end user respondents in terms of effectiveness of the system. The indicator was rated as follows; Effectiveness, 4.8. This result indicates that the system is significantly effective in terms of achieving specified goals when used in specified contexts.

**Table 17. Perception of End-users in the Effectiveness of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Effectiveness | 4.8 | Excellent |
| Average Mean | 4.8 | Excellent |

Table 18 reflects the assessment of the end user respondents in terms of the efficiency of the system. The indicator was rated as follows; Efficiency, 4.7. This result suggests that the system demonstrates a commendable level of efficiency by effectively utilizing resources to assist users in achieving their objectives accurately and comprehensively.

**Table 18 Perception of End-users in the Efficiency of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Efficiency | 4.7 | Very Satisfactory |
| Average Mean | 4.7 | Very Satisfactory |

Table 19 reflects the assessment of the end user respondents in terms of user satisfaction of the system. The indicator was rated as follows; Usefulness, 4.8; Trust, 4.7; Pleasure, 4.6, and Comfort, 4.7. The mean of 4.7 indicates that the system is relevantly satisfied the user in terms of trust, pleasure, comfort, and usefulness.

**Table 19 Perception of End-users in the Satisfaction of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Usefulness | 4.8 | Excellent |
| 1. Trust | 4.7 | Excellent |
| 1. Pleasure | 4.6 | Excellent |
| 1. Comfort | 4.7 | Excellent |
| Average Mean | 4.7 | Excellent |

Table 20 reflects the assessment of the end user respondents in terms of context coverage of the system. The indicator was rated as follows; Content Completeness, 4.7; and Flexibility, 4.8. The mean of 4.75 indicates that the system is significantly complete and flexible in order for the users to achieve specified goals effectively, efficiently, and satisfactorily through different complexity.

**Table 20. Perception of End-users in the Context Coverage of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Content Completeness | 4.7 | Excellent |
| 1. Flexibility | 4.8 | Excellent |
| Average Mean | 4.75 | Excellent |

The evaluation results of the system for the end user respondents are summarized in Table 21. Overall, the end users found the system to be user-friendly, accessible, trustworthy, adaptable, useful, and comfortable. The average mean score for these evaluations is 4.70, indicating a highly satisfactory level of performance.

**Table 21. Perception of End users in the Summary of End-user of the system**

|  |  |  |
| --- | --- | --- |
| Characteristics | Average Score | Descriptive Interpretation |
| 1. Usability | 4.58 | Excellent |
| 1. Effectiveness | 4.8 | Excellent |
| 1. Efficiency | 4.7 | Excellent |
| 1. Satisfaction | 4.7 | Excellent |
| 1. Context Coverage | 4.75 | Excellent |
| Average Mean | 4.70 | Excellent |

**SUMMARY, CONCLUSIONS AND RECOMMENDATION**

**Summary**

The research project entitled "Automated Timetable Scheduling System using Backtracking Algorithm" was conducted from May 2022 to May 2023. The objectives of the study were as follows: 1) document and analyze the existing process of scheduling classes, 2) design and develop a web application that automate the generation of class schedules. 3) test the reliability of generated schedules based on testing conditions and test the functionality of system modules. b) Evaluate the system using modified ISO-IEC 25010, 4) Create an implementation plan for the deployment 5) to evaluate the system using the ISO/IEC 25010. During the study, the developers adopted the Rapid Application Development (RAD) methodology, which involved four phases: Requirement Analysis Phase, Prototype Cycle Phase, Developing and Testing Phase, and Deployment Phase.

During the Requirement analysis phase, the developers collected and analyzed information about the existing scheduling process, as well as studied related articles, books, studies, and systems to identify the criteria for assigning timeslots and rooms. Once the requirements and necessary data were gathered, the developers created flow charts and prototypes (or wireframes) for the proposed system. Subsequently, the prototype was transformed into a functioning system, which underwent refinements whenever errors or bugs were identified.

**Conclusion**

The following conclusion were made based on the results of the study:

* The developers were able to develop a dynamic web-based system to automates the generation of schedules using backtracking algorithm.
* The researchers were able to document and analyzed the current scheduling process at Cavite State University CCAT Campus and used it for designing the prototype or wireframes.
* The researchers were able to test the reliability of generated outputs based on the testing conditions. The generated output passed all the testing condition which means that the generated schedules were free of conflict.
* The developers were able to test the functionality of each system modules using the test cased provided in the manuscript.
* The researchers were able to use the modified ISO-IEC 25101 for evaluation. Collectively, these evaluation results highlight the exceptional performance and characteristics of our developed system, demonstrating its effectiveness, efficiency, usability, security, maintainability, and contextual coverage. The positive ratings from both IT experts and end users affirm its high quality and ability to meet user requirements, making it a highly satisfactory and excellent system overall.

**Recommendation**

Based on the results the following recommendation was made:

1. Printing Specific Schedules: Allow users to easily print specific schedules directly from the system.
2. User and Teacher Management: Introduce a user management system that enables administrators to add and manage user accounts. Additionally, provide teachers with access to view schedules online.
3. Teacher Swap Timeslots: Add a functionality where instructor could swap timeslots with another instructor.
4. Loading Screen Animation: Add a loading screen so the user knows what is happening during generation of schedules.

**REFERENCES**

Ahmed Gad (2018), Introduction to Optimization with Genetic Algorithm. <https://www.kdnuggets.com/2018/03/introduction-optimization-with-genetic-algorithm.html>

Al-Jarrah, M. A., Al-Sawalqah, A. A., & Al-Hamdan, S. F. (2017). Developing a course timetable system for academic departments using genetic algorithm https://www.researchgate.net/profile/Sami-Hamdan-4/publication/313487752\_Developing\_A\_Course\_Timetable\_System\_for\_Academic\_Departments\_Using\_Genetic\_Algorithm/links/601b7a55299bf1cc26a00768/

Ander Fernández Jauregui (2022), Genetic Algorithm in Python. <https://anderfernandez.com/en/blog/genetic-algorithm-in-python/>

“Simplelearn” (2022), What Is Data Collection: Methods, Types, Tools, and Techniques <https://www.simplilearn.com/what-is-data-collection-article>

Ben Moreland, (2015), Hard and Soft timetabling constraints. <https://educationspaceconsultancy.com/hard-and-soft-timetabling-constraints-not-only-a-working-timetable-but-a-great-timetable/>

“CQL” (2022), 12 Reasons Why Data is Important. https://www.c-q-l.org/resources/guides/12-reasons-why-data-is important/

Edmund Kieran Burke, Sanja Petrovic (2002), Recent research   directions in automated timetabling <https://doi.org/10.1016/S0377-2217(02)00069-3>

Manoj Kumar, Mohammad Husian, Naveen Upreti & Deepti Gupta (2010), Genetic Algorithm: Review and Application. https://deliverypdf.ssrn.com/delivery.php

Paymon Khamooshi (2019), The Benefits of using a web-based application. <https://www.geeks.ltd.uk/insights/the-benefits-of-using-web-based-applications>

Soni Upadhyay (2023), Introdcution Backtracking algorithm <https://www.simplilearn.com/>

“Stanford University (2017), Recursive Backtracking and Optimization. https://web.stanford.edu/class/cs106b-8/lectures/backtracking optimization/Lecture13.pdf

Subhan Datta (2022), Backtracking Algorithms. https://www.baeldung.com/cs/backtracking-algorithms

Zina Houhamdi, Belkacem Athamena, Rani Abuzaineddin, Mohammad Muhairat (2019), A Multi-Agent System for Course Timetable Generation. <https://www.researchgate.net/publication/331995150_A_multi-agent_system_for_course_timetable_generation>

**Related System**

Abhik Mitra (2020), Intelligent Timetable Generator using Genetic Algorithm. <http://abhik.pythonanywhere.com/>

Gamale, Jeralyn M., Anuta, Edward C., Sayson, Zina D. (2011), Automated Scheduling System Capstone Project. <https://capstoneguide.com/automated-class-scheduling-system-capstone-project-document/>

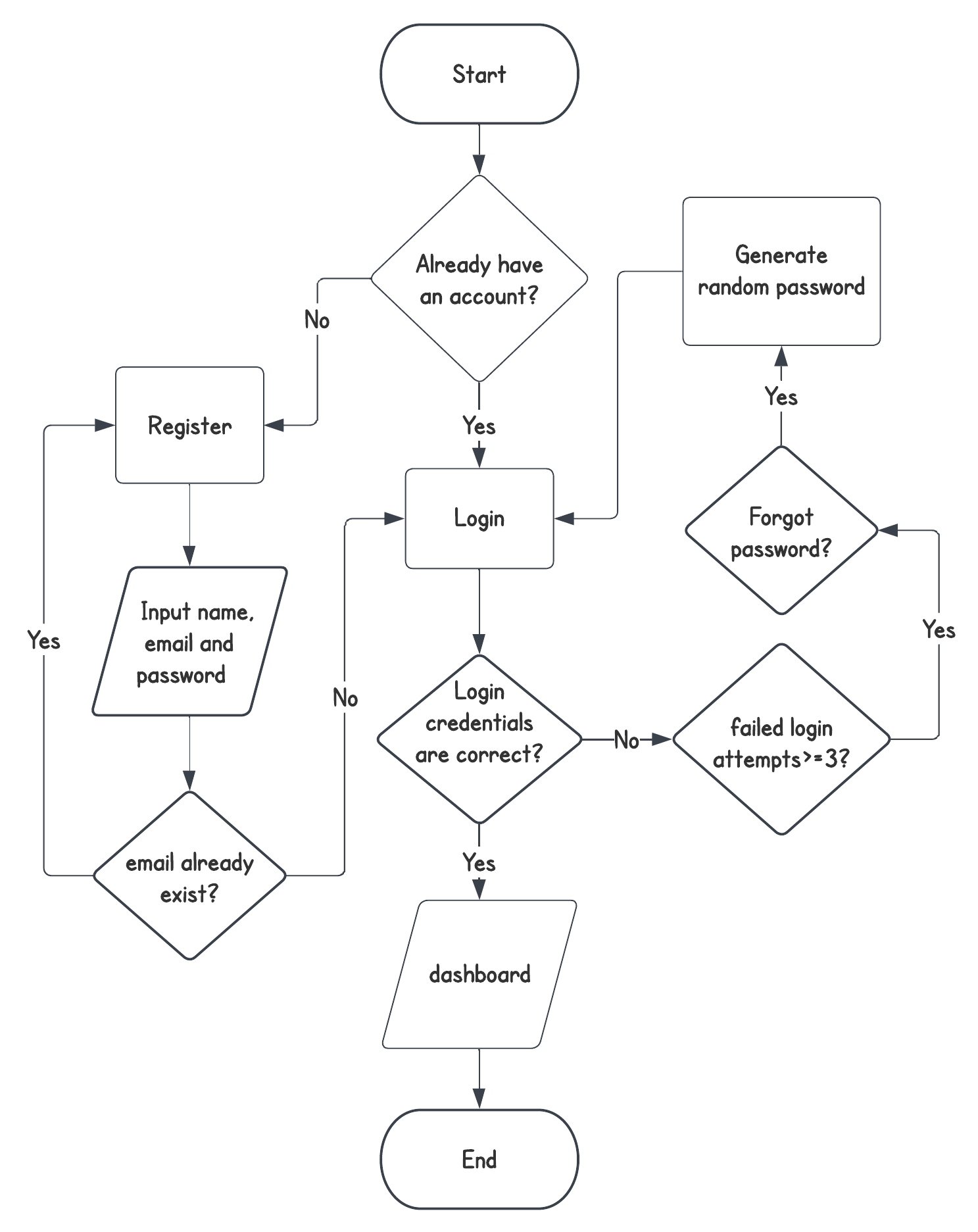
John Benedict C. Legaspi, Roman M. De Angel, Ace C. Lagman, John Heland Jasper C. Ortega (n.d), Web based Course Scheduling System using Greedy Algorithm. <https://ijssst.info/Vol-20/No-S2/paper14.pdf>

Loic Ngou (2019), Timetable Generator using Genetic Algoritmh. https://Timetable Generator

“Neon Projects” (2015), Automated College Timetable Generator. <https://nevonprojects.com/automated-college-timetable-generator/>

**APPENDIX TABLES**

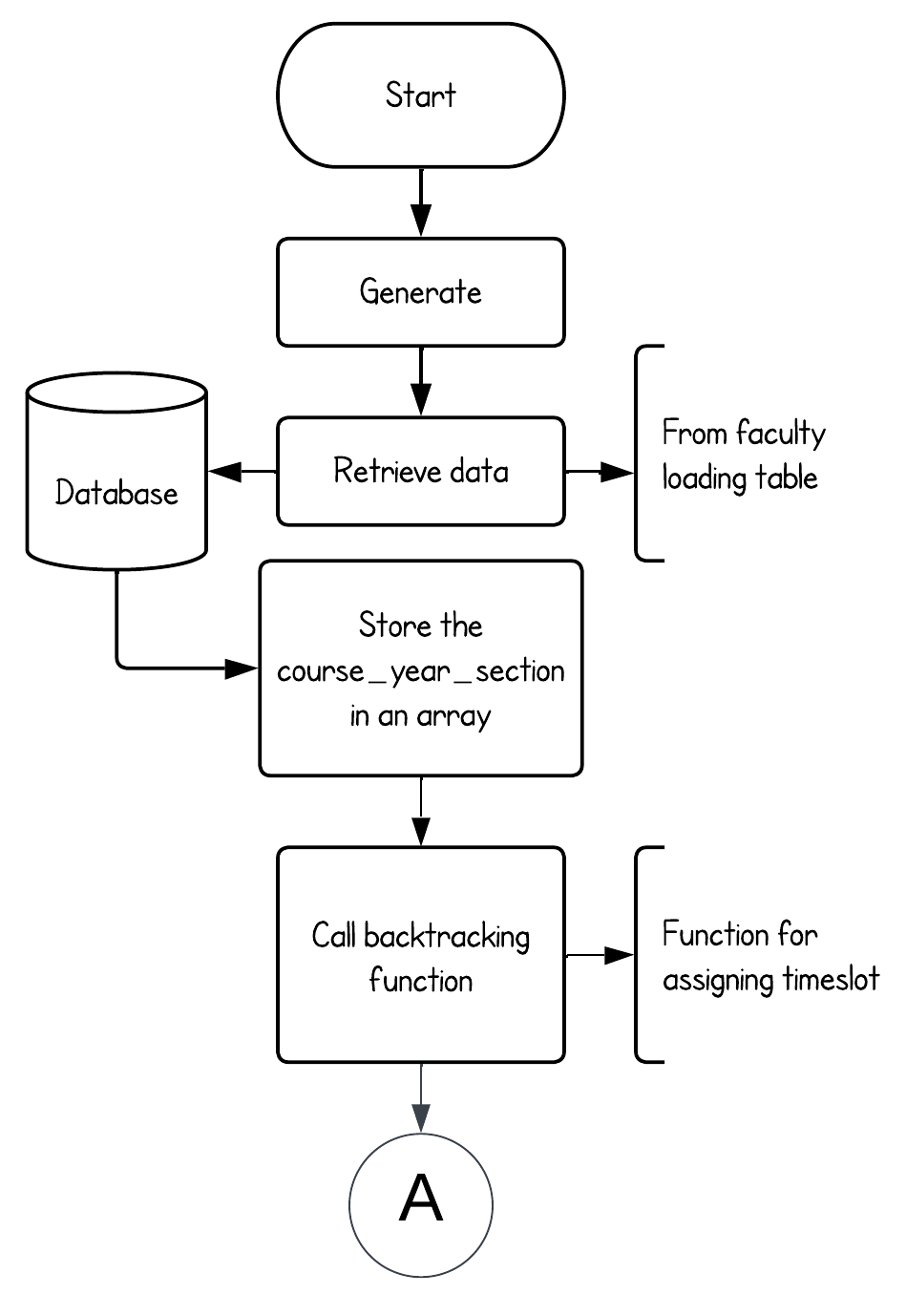
**APPENDIX FIGURES**

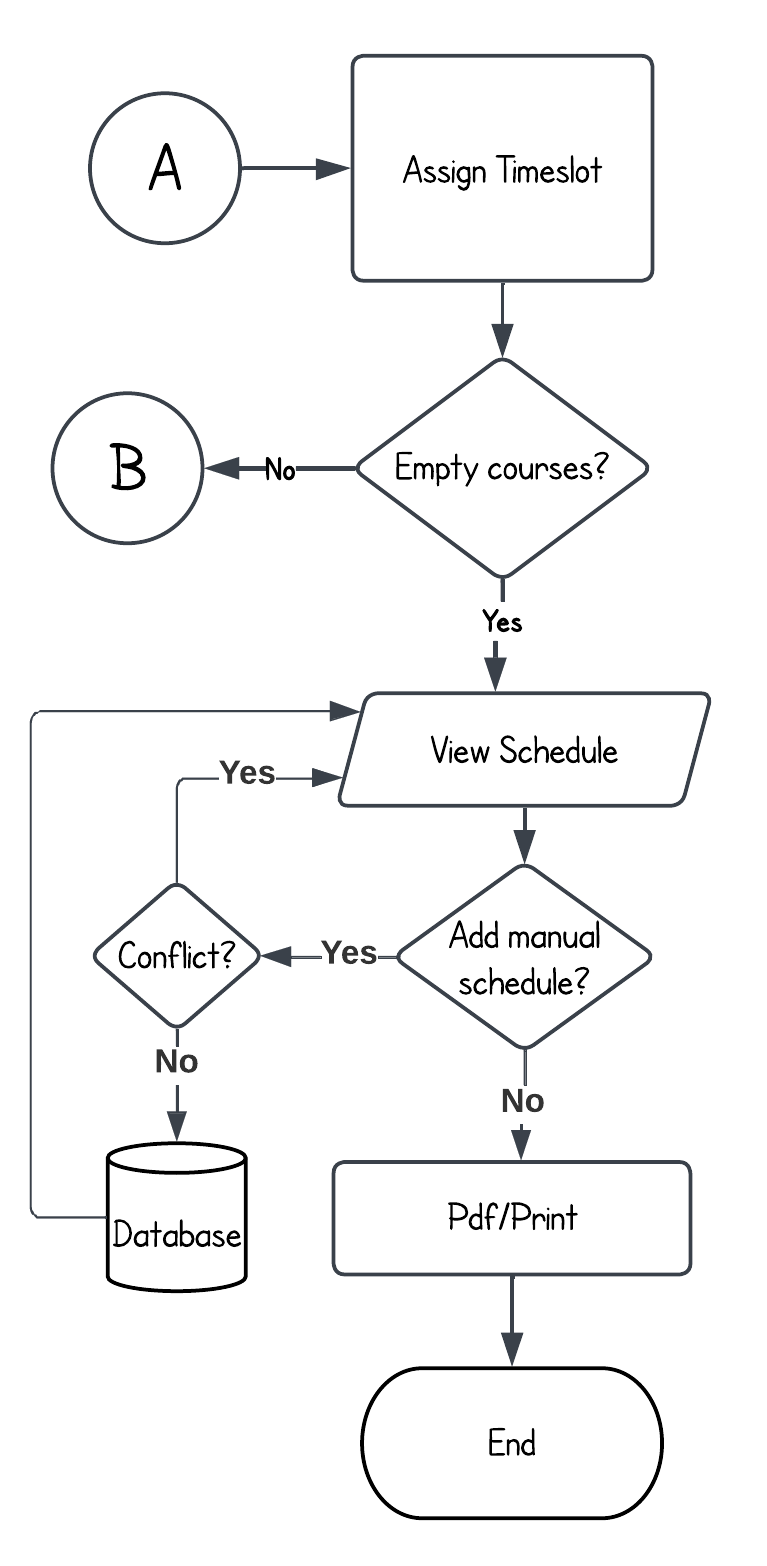


Appendix Figure 1. Login Flowchart

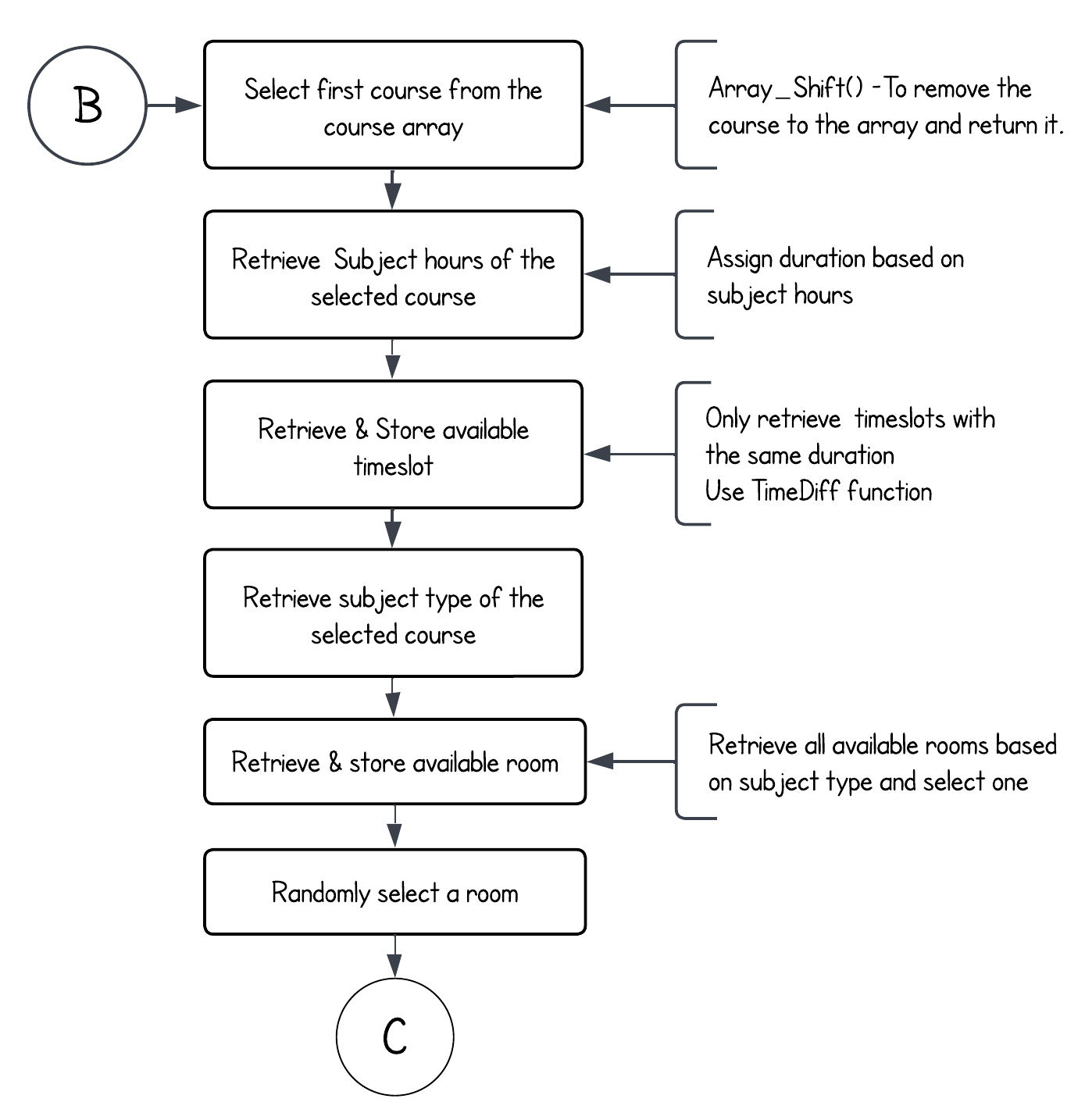


Appendix Figure 2: Faculty loading Flowchart

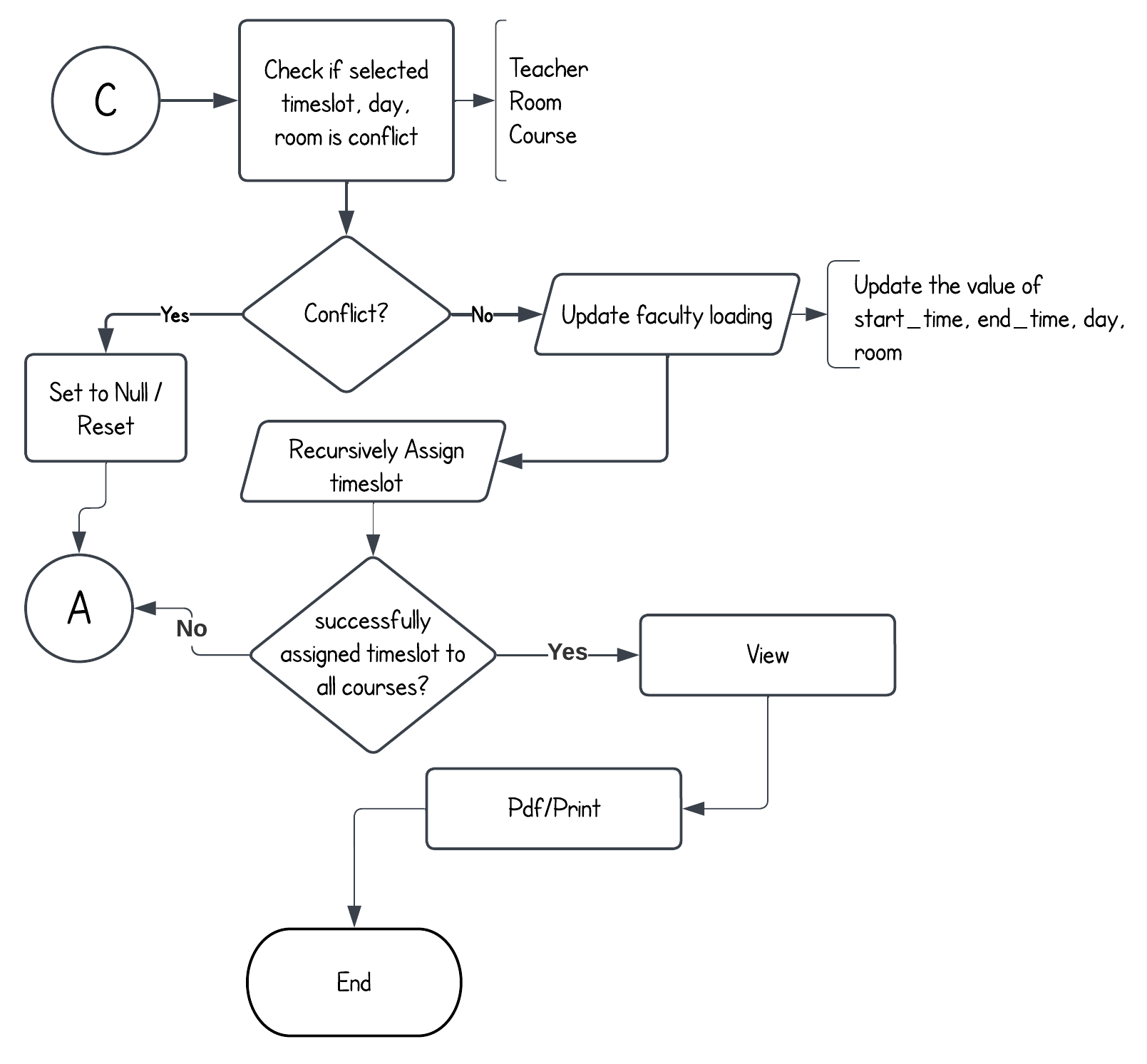
****Appendix Figure 3: Automated Scheduling Flowchart



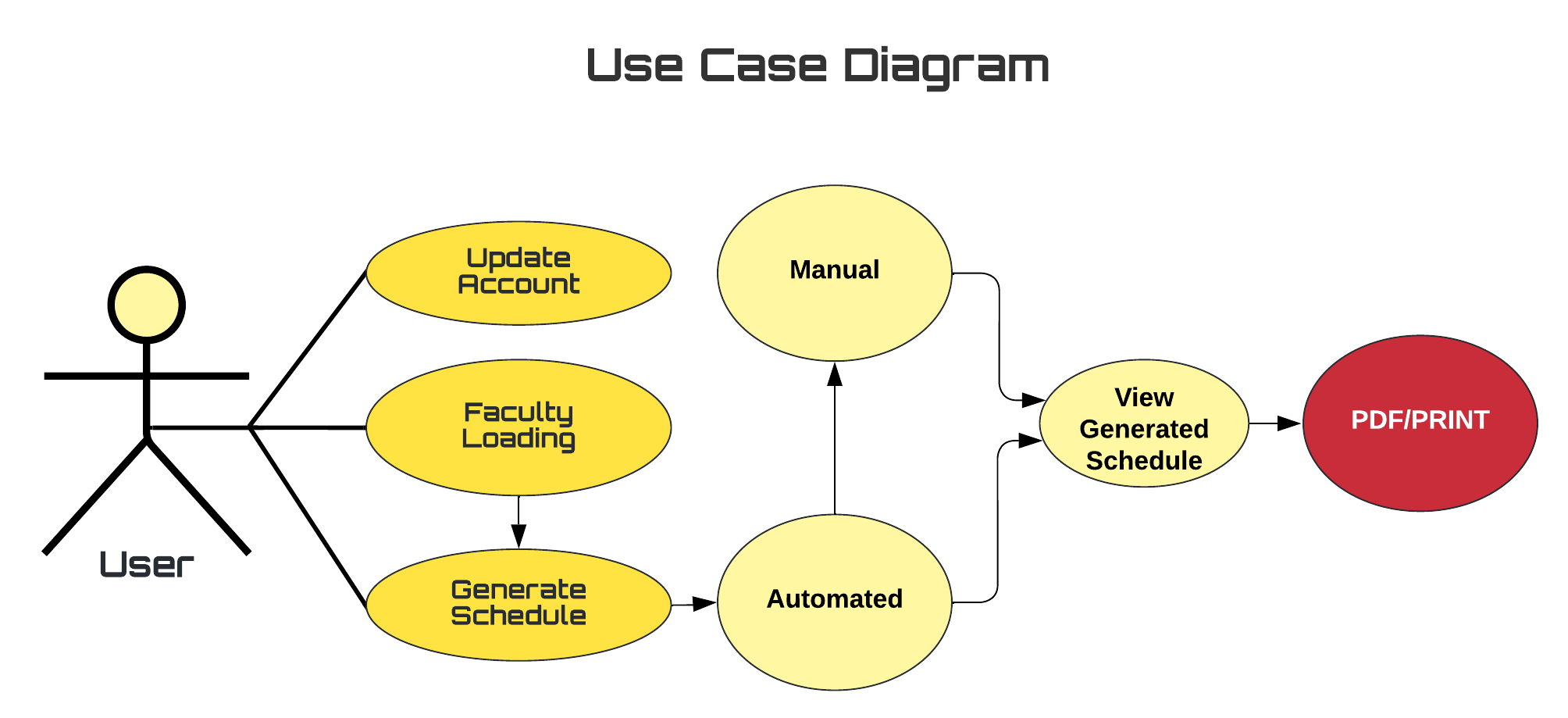
Appendix Figure 4: Automated Scheduling Flowchart Pt.2

****

Appendix Figure 5: Automated Scheduling Flowchart Pt.3

****

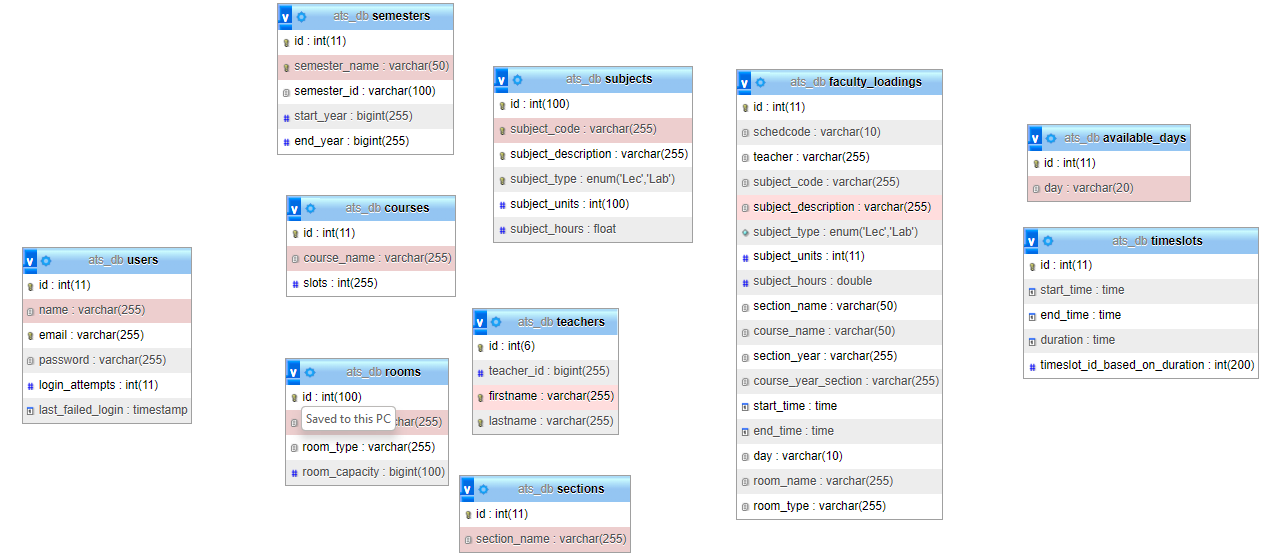
Appendix Figure 6: Automated Scheduling Flowchart Pt.4

****

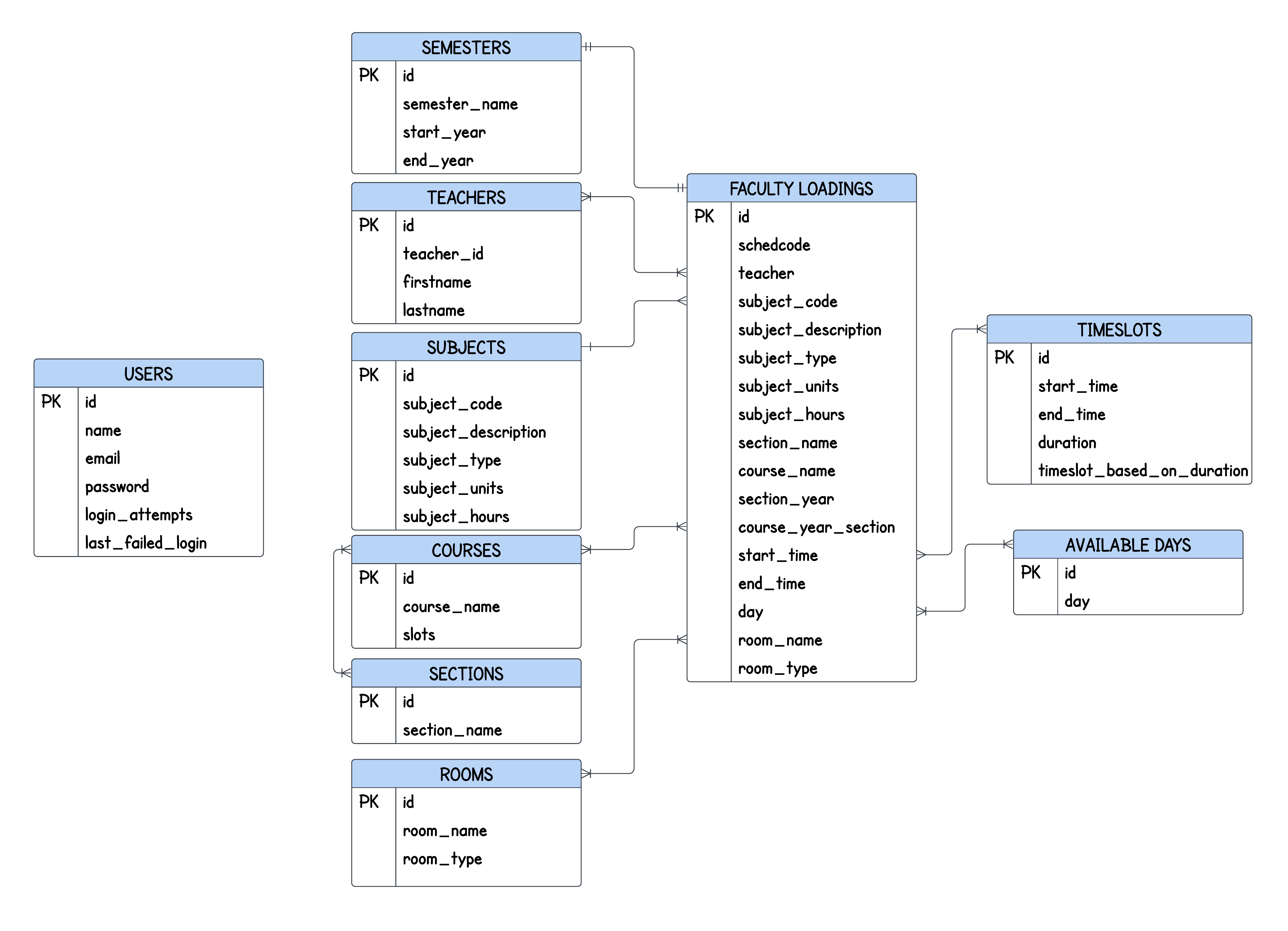
Appendix Figure 7. Use Case Diagram

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **TASK** | **2022-2023** | | | | | | | |
| **MAY** | **JUNE** | **JULY** | **AUG.** | **SEPT.** | **OCT.** | **NOV.** | **DEC.** |
| **1** | **Requirements Analysis and Design phase** |  | | |  |  |  |  |  |
| **2** | **Prototype Cycle Phase** |  |  |  |  |  |  |  |  |
| **3** | **Developing and Testing Phase** |  |  |  |  | | | | |
| **4** | **Deployment Phase** |  |  |  |  |  |  |  |  |

Appendix Figure 8. Gantt Chart

****

Appendix Figure 9 ATS Database Scheme

****

Appendix Figure 10 ATS Entity Relationship Diagram