schEDU: WEB-BASED PUBLIC SENIOR HIGH SCHOOL SCHEDULING SYSTEM USING ANT COLONY OPTIMIZATION

A Thesis
Presented to the Faculty of the
Computer Studies Department
College of Science
Technological University of the Philippines
Ayala Boulevard, Manila

In Partial Fulfilment of the
Requirements for the Degree
Bachelor of Science in Computer Science

By
SOPHIA J. DALUMPINES
CHRISTINE G. DELOS REYES
LEA O. FLOR
SELWYN AUDREY C. PEREZ
RENZ L. SARMIENTO

June 2024

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines | Tel No. +632-5301-3001 local 608 Fax No. +632-8521-4063 | Email: cos@tup.edu.ph | Website: www.tup.edu.ph	Index No.	TUPM-F-COS-16-TAU
		Revision No.	00
	THESIS APPROVAL SHEET FOR THE UNDERGRADUATE PROGRAMS OF THE COS	Date	07012022
		Page	1 / 1

This thesis hereto entitled:

schEDU: WEB-BASED PUBLIC SENIOR HIGH SCHOOL SCHEDULING SYSTEM USING ANT COLONY OPTIMIZATION

Prepared and submitted by **SOPHIA J. DALUMPINES, CHRISTINE G. DELOS REYES, LEA O. FLOR, SELWYN AUDREY C. PEREZ, RENZ L. SARMIENTO** in partial fulfillment of the requirements for the degree **BACHELOR OF SCIENCE IN COMPUTER SCIENCE** has been examined and is recommended for approval and acceptance.

PROF. DOLORES L. MONTESINES
Adviser

Approved by the Committee on Oral Examination with a grade of **PASSED** on JUNE 5, 2024.

PROF. MAY GARCIA
Member

PROF. PRISCILLA BATOR
Member

PROF. PERGARINO AMADOR JR.
Member

PROF. DOLORES L. MONTESINES
Department Head/Chair

Accepted in partial fulfillment of the requirements for the degree **BACHELOR OF SCIENCE IN COMPUTER SCIENCE**.

Date: June 13, 2024

DR. JOSHUA T. SORIANO
Dean

Transaction ID	TUPM-COS-TAU-ELS-07012022 -0258PM
Signature	

Transaction ID Legend: TUPX-AAA (Office Code)-BBB (Type of Transaction)-CCC (Initial of employee)-MMDDYYYY (month day year)-HHMMAM/PM (hourminutesAM/PM)

ACKNOWLEDGEMENT

With great appreciation, the researchers acknowledge the support, guidance and contributions of every individual who had crucial roles in the completion of this thesis.

First and foremost, the researchers extend their deepest gratitude to Prof. Dolores L. Montesines, their dedicated thesis adviser. Her profound wisdom, unwavering guidance throughout the entire process, and careful attention to every concern from the initial topic proposal to the final defense have been invaluable and are deeply appreciated.

This study was also made possible by the invaluable contributions of the panelists, Prof. Amador Peregrino Jr., Prof. Priscilla Bator, and Prof. May Garcia. Their thoughtful feedback, insightful suggestions, generous commitment in final defense were instrumental to the success of this work.

The researchers extend their heartfelt gratitude to the participants of this study, who generously offered their time, insights, and ideas. Without their contributions, their research would not have been possible. Their input was vital in understanding the underlying issues and developing effective solutions. The researchers also wish to thank their classmates, friends, and family for their unwavering support and encouragement, which kept them motivated and committed throughout the completion of this thesis.

Lastly, the researchers would like to express their heartfelt thanks to all the individuals who have served as sources of inspiration and influence. Whether directly or indirectly, their contributions have been instrumental not only to the researchers' academic development but also to their personal and interpersonal growth.

ABSTRACT

Every academic year, academic institutions face the challenging and time-consuming task of creating a schedule that accommodates the preferences and needs of their teachers and students. This challenge is no different for senior high schools. This paper aims to introduce schEDU, a web-based scheduling system specifically targeted for public senior high school. This system will utilize Ant Colony Optimization (ACO) to generate a seamless schedule for one semester: room availability, teachers' subject preferences, and the required subjects for each section throughout the semester. The system employed fixed parameters: a 6-hour maximum teaching load per day, school days from Monday to Friday, time slots from 7:00 am to 3:00 pm, and 30-minure break times at 9:30 am and 12:30 pm. The developed web-app make used of Flask for backend operations, React for frontend development, and MySQL as a database manager. The evaluation results revealed highly acceptable ratings for the system's functional suitability (3.85), performance efficiency (3.71), usability (3.78), and maintainability (3.75). Overall, the average rating was 3.77, indicating a highly acceptable performance. The study's findings shows that the ACO algorithm was successfully implemented to construct efficient schedule solutions based on the required inputs. With this scheduling approach, the issues associated with manual scheduling are reduced. The schEDU web application ensured that each generated schedule was free of conflicts and adhered to senior high school scheduling constraints. Using the ACO algorithm in the system's development proved efficient and effective, producing non-conflicting schedules for all sections and teachers. Future enhancements could focus on expanding the scope and number of user roles, accommodating multi-shift scheduling, and extending the scheduling process to generate schedules for an entire academic year.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

INTRODUCTION

Background of the Study

In the modern era of internet and information technology, cyberspace breaks conventional lines between different domain (Rita, 2013). For schools, it is necessary to handle in a more convenient and methodical way of questioning how to work with information. The number of students studying at large universities have increased and deeper changes are shown in education due to the fast expansion of the economy recently. One of the most challenging problems arise from the growing size of school's teaching facilities that can't accommodate more students is its daily teaching schedule. It must show where exactly to find a classroom, what courses are going to be taught there, some student names, who will teach them and how long each class should take. Manual scheduling is ineffective and laborious (Minna & Yanxiang, 2015). To improve productivity, reduce labour costs and relieve academic staff members' workloads at colleges and universities, an automated scheduling system must be implemented (Chen et al., 2021).

The task of arranging courses is extremely complex, requiring careful consideration of numerous parameters and data dimensions throughout the entire process. It also calls for a lot of endurance. In order to schedule classes, it is required to resolve a variety of conflicting issues and ascertain a number of conditions, such as different teaching facilities, teachers and students, and other elements. For example, a course cannot be present in two classrooms simultaneously, a teacher cannot attend two courses simultaneously, and a course cannot have two or more teachers assigned to it simultaneously. A class that is taught again in a single day cannot have fewer students than the total number of classes in the classroom (Tao, 2015). On a given day, a single teacher ought to teach several classes in the same location. Artificial efficiency during issue solving is low. This work requires the use of a computer to be completed in order to increase efficiency and save money. An effective software system can be created as long as the programming language is chosen, and the software algorithm is optimized. The system must meet the robust and safe performance criteria based on network applications and handle scheduling requirements such as diversification, frequent schedule adjustments, massive amounts of data, etc. (Chen et al., 2021).

Ant colony optimization (ACO) is a popular metaheuristic for combinatorial optimization issues that draws inspiration from actual ant colonies' foraging practices. Ants can determine the fastest route connecting their colony and a food supply. They create a pheromone trail by leaving pheromones—a chemical compound made by ants—on the ground to indicate their route. They also tend to follow trails with high concentrations of pheromones. The ants can communicate with their nest companions in this way.

A colony of (artificial) ants collaborate to create superior solutions to challenging combinatorial optimization issues in ACO. Solutions are built using (artificial) pheromone trails as guidance. The pheromone trace leaving and following activities of actual ants served as the model for the pheromone model, which was utilized to update the pheromone trails. In essence, the ants have altered a parametrized probabilistic model to account for their experiences in solving a specific challenge (Mavrovouniotis et al., 2020).

Real ant colonies have intriguing behaviour that are appropriate for network traversal and, consequently, scheduling. Specifically, an ant can assist in determining the quickest path between food sources and a nest by randomly exploring potential courses. In the process, it deposits a chemical called pheromone, creating "pheromone trails" that other ants in the colony can follow. Ants utilize their sense of smell to determine which path routes are possible and they often choose those that are indicated by stronger smell signals. As a result, an ant in isolation wanders practically at random, but an ant that comes across a path that has already been travelled with a pheromone-laid trail can recognize such a thing, decide to follow it, and then strengthen the track with its particular pheromone.

The shared behaviour is further described by a positive (reinforcing) response loop in which the likelihood that any one ant would select the correct way to follow rises as more ants select the same path in the previous steps. The path-traversing path eventually converges to the shortest path quite quickly (Computational Technology Resources - CCP - Paper, 2023).

In this study, the researchers will use Ant Colony Optimization to develop a system that would generate a schedule for senior high schools. Considering constraints such as subject requirement, preferred subjects of teachers, room allocation, definite class hours, and etc., the system will be able to produce a schedule of an entire academic year for public senior high school under the Department of Education curriculum.

Objectives of the Study

The main objective of this study is to develop a web-based scheduling system for Public Senior High Schools employing the metaheuristic Ant Colony Optimization approach. The system is designed to produce an optimized schedule for a single semester within the senior high school academic year. The system takes the following inputs to construct an optimize schedule: room availability, teachers' subject preferences, and the required subjects for each section throughout the semester. The researchers employed fixed parameters: a 6-hour maximum teaching load per day, school days from Monday to Friday, time slots from 7:00 am to 3:00 pm, and 30-minure break times at 9:30 am and 12:30 pm.

Specifically, this study aims to achieve the following objectives:

1. Develop a web-based class scheduling application with the following features:
    a. Allows the user to add subjects, teachers and their preferred subjects, rooms, and sections with their required subjects to take.
    b. Allows the user to modify or delete the data input.
    c. Generate a schedule based on the input of the user.
    d. View the generated schedule per section or per teacher.
    e. Export the generated schedule to a PDF File.
2. Implement Ant Colony Optimization as an algorithm in generating schedules.

Scope and Limitations of the Study

This study aims to build an automated scheduling system, schEDU, tailored for senior high school using the Ant Colony Optimization (ACO). schEDU is intended to enhance the scheduling process of senior high schools, offering efficiency for both faculty and students. By connecting with the school's resource management system, schEDU can ensure that classrooms, labs, and other facilities are used efficiently.

The main objective is to create a generated system that would benefit the senior high school focal person, teachers, and students. It includes the teacher’s preferred time accessibility, preferred subject areas, and the time allotted for each class. While teachers' preferences for subjects are taken into account, each teacher is restricted to a maximum of three subject preparations.

Technologically, schEDU is built using Python as the primary programming language. The frontend is configured with React, and Flask is used for the system's framework. MySQL is utilized for database configuration.

Limitations

Because of the varied and complex availability of part-time teachers in private institutions, the system's current functionality is limited to serving only public senior high schools.

schEDU can only cater system input for a single-shift schedule and can only generate schedules for one semester at a time. To use the system, the academic year's enrollment should already be conducted to determine the number of students and sections. The school registrar or scheduling personnel will then enter the finalized sections, the needed subjects to take for the semester, the list of teachers and their subject preferences for their workload, and the classroom availability. Lastly, while these inputs are used to design a schedule solution, additional factors influencing the school calendar, such as holidays, are not considered by the algorithm.

Chapter 2

REVIEW OF RELATED LITERATURE

This chapter lays out a review of related literature and studies, as well as the study's conceptual model and operational definitions of words pertinent to this study.

K TO 12 Curriculum

The K-12 Program includes Kindergarten and 12 years of basic education (six years of primary education, four years of Junior High School, and two years of Senior High School [SHS]) to allow for sufficient time to master concepts and skills, develop lifelong learners, and prepare graduates for tertiary education, middle-level skill development, employment, and business (The Department of Education, Republic of the Philippines, 2018).

The K-12 program was implemented in 2012. Through this reform, the Philippines is catching up with worldwide secondary education standards and emphasizing the importance of early childhood (The Department of Education, Republic of the Philippines, 2018).

Senior high school was implemented as a part of the K-12 reforms in 2016. The Grade 12/Senior High School Diploma was first granted in 2018.

Public School Teacher Working Hours and Teaching Load

The Republic Act No. 4670, occasionally referred to as the Magna Carta for Public School Teachers, gives teachers' teaching hours shall be a maximum of six hours. The Department of Education (DepEd) published Memorandum 291. 2008 permitting instructors to allocate six hours each day for class teaching. The remaining two hours would then be spent in other teaching-related activities outside of the classroom. Thus, teachers must offer a total of eight hours for the day (Department of Education, Republic of the Philippines, 2008; Republic Act No. 4670, 1966).

DepEd School Calendar

According to Republic Act (RA) No. 11480, also known as An Act to Lengthen the School Calendar from Two Hundred (200) Days to No More Than Two Hundred Twenty (220) Class Days. The DepEd school calendar is more than a schedule; it serves as a framework for the Philippines' educational ecology. Its evolution demonstrates the country's responsiveness to external challenges and dedication to improving educational performance. To accommodate the changing demands of all stakeholders involved in the educational process, the school calendar must be reviewed and adapted on a regular basis.

Senior High School Curriculum

Based on the DepEd Senior High School Curriculum (n.d.), these are the SHS tracks that the students can select from: (1) Academic, (2) Arts and Design, (3) Sports, and (4) Technical-Vocational-Livelihood.

The Academic Track has four strands: (1) Accountancy, Business and Management (ABM), (2) Science, Technology, Engineering, and Mathematics (STEM), (3) Humanities and Social Science (HUMSS), and (4) General Academic Strand (GAS).

The Technical-Vocational-Livelihood has the following strands: (1) Agricultural-Fishery Arts (AFA), (2) Home Economics (HE), (3) Industrial Arts (IA), (4) Information and Communications Technology (ICT).

The Arts and Design Track encompasses a variety of art forms such as Theater, Music, Dance, Creative Writing, Visual Arts, and Media Arts.

The Sports Track is tailored for students aiming for careers related to sports, including athlete development, fitness education, and mentoring.

Every student enrolled in the DepEd Senior High School Curriculum is mandated to complete the academic regimen encompassing fifteen Core subjects and sixteen Applied Track Subjects and Specialized Subjects specific to their chosen strand. Each subject is structured with 80 hours allocated per semester, except for Physical Education (PE), which is allotted 20 hours.

Class Scheduling Problem

When dealing with academic institutions, generating a competent schedule that considers instructors' preferences and needs as well as those of the administrative staffs and students can be daunting and time consuming. This is despite recent developments in software and hardware technologies that have not achieved much efficiency considering that many educational institutions still create charts manually (Aslan, 2018).

Scheduling is the process of coordinating, directing, and optimizing work. The goal of scheduling is to reduce time and expenses by instructing a institute on when and where to process. Scheduling improves process efficiency and reduces costs. It is a technique for allocating critical computing resources such as bandwidth, memory, and processing time. Scheduling is used to allocate resources evenly, balance system load, and assign priority according to a predetermined set of rules. This ensures that the computer system can handle all requests while maintaining a particular level of service. The scheduling in a system is done by a scheduler, who is concerned with numerous factors, including throughput, latency, and reaction time. A process's throughput defines how quickly it will perform a particular number of tasks from the beginning of the process to the end (Gawali, 2018).

Scheduling, as defined by Liu (1973) and quoted in New Study by Sarathambekai (2022), varies according to the system and usage, as well as the user's preferences and goals. It is primarily based on the elements listed above. The scheduler controls demand on modern resources with high processing power and the ability to multitask by running multiple threads at the same time to provide the best answer to problems. Scheduling examples for optimization problems include job shop scheduling, open shop scheduling, and flow shop scheduling. The scheduling of optimization problems is determined by time constraints and problem minimization. Which deals with jobs completed with diverse features and order of execution by machines.

Scheduling is the process of selecting how to distribute resources among the many job classifications. The goal is to distribute assets across time to execute a sequence of tasks. Many articles in the field of operational research have addressed this issue. This is a key decision-making process in most manufacturing and service industries. It can also be defined as a decision-making process for optimizing goals such as accomplishing and lowering makespan (Hakim, Rekiek, & Reklaoui, 2022). Scheduling challenges can be represented as allocation concerns, which are a vast class of combinatorial optimization problems. In most such circumstances, determining the best option is extremely difficult.

Educational institutions have to deal with class scheduling issues, which sometimes prove difficult due to a number of constraints and complexities. According to Jansen Wiratama et al., (2023) this problem is NP-hard and thus requires efficient solution procedures such as genetic algorithms. In addition, various universities are grappling with problems like manual schedule distribution that leads redundancy of data and poor efficiency hence necessitating new alternatives like mobile based applications (Chen et al., 2022). Another research that has been conducted also focused on advanced optimization techniques including multi-class teaching-learning-based optimization method for reducing make span and tardiness in distributed hybrid flow shop scheduling problems (Lei & Su, 2023). Besides, the scheduling issue continues to job allocation on parallel batch machines involving models such as mixed integer linear programming or constraint programming with special algorithms such as iterated greedy among others showing different ways in which scheduling challenges can be adequately addressed (Jiang et al., 2021).

Solving Scheduling Problem using Various Swarm Intelligence Algorithms

The coming field of swarm intelligence (SI) takes its cue from the dispersed, prearranged systems like ant colonies, bird flocks and fish schools. This approach has increasingly been used to solve difficult optimization problems especially scheduling problem. Efficient allocation of resources aimed at optimal performances is a crucial aspect in scheduling within different domains such as manufacturing, project management and education.

A number of swarm intelligence techniques: Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), have had promising results in solving scheduling problems. These techniques mimic natural processes to find near-optimal solutions through iterative improvement.

Particle Swarm Optimization (PSO): The PSU is founded on the social of birds and fishes where these animals move in flocks. Every point in the swarm is a potential solution and moved according to personal and global experience (Kennedy& Eberhart, 1995) PSO has been applied to different scheduling problems including job-shop scheduling and project scheduling as the presence of PSO identified efficient solutions in case of different scheduling problems in reasonable response time (Marinakis et al., 2008). Genetic algorithms as a part of evolutionary algorithms begin with a populace of possible keys – particles and try to improve the solutions with the help of a quality factor known as a fitness function. The improvised is done with the movement of the particles across the search space by applying certain mathematical equations which demonstrate inter-particle interaction in a certain minimalistic way. To the simplest of these equations, these mathematical formulas advise that each particle should move near its personal best experienced and also towards the best experienced by the whole swarm, all with some randomness. There are many models that have other rules for update, so now it can be considered with variations (Kaveh, 2014).

Bee Colony Optimization (BCO): BCO mimics the food finding process observed in bees through the waggle dance, where bees move in a specific pattern to indicate the location of food sources (Karaboga & Basturk, 2008). Users of BCO include job scheduling, tasks assigning and other combinatorial optimization problems. For this reason, its ability to indulge in both the exploitation and the exploration of new areas of interest makes it most suited to dynamic and complex scheduling (Pham et al., 2006). BCO is a wellover algorithmic reasoning relatively new metaheuristic for handling challenging combinatorial optimization problems. It is one of the approaches that honey bees use to try and understand working intelligent in the process of, for example, nectar foraging (Branka Dimitrijević et al., 2012).

Ant Colony Optimization

Another SI algorithm is the Ant Colony Optimization technique or ACO which mimics how ants search for a source of food starting from their nests by following the shortest routes while lacking sight or any form of vision (Old Study 2007 cited in New Study, 2022). Although ACO is a population optimization approach that Dorigo proposed in 1992, the technique has been effectively applied to solve a number of NP-hard combinatorial problems, which can be heuristically solved to provide approximate solutions to the specified problem (Old Study, cited in New Study, 2022). To avoid reaching a locally optimal solution systematically, the process of evaporating pheromones plays a role.

Captioning from Dorigo & Stützle, (2018), the Ant colony optimization (ACO), is a solution technique inspired by the foraging behaviour of certain species of ants on the ground where they use a process known as pheromonal marking on the ground to highlight the favourable paths. This method of depositing pheromones is used by performing an economic analogy of the biological procedure in ant colony optimization for solving optimization dilemmas.

In the early of the 1990's, Marco Dorigo, as a PhD thesis, proposed the ant colony system as a population-based metaheuristic, inspired by the foraging behaviour of ants searching for the nearest straight path between their nest and a food origin. Originally, ACO was developed for the TSP, a particularly difficult task; However, due to its versatility and high performance, it has been successfully applied in various other fields of complex combinatorial optimization problems (Soofastaei, 2022).

There are many optimization problems where you can use ACO for finding the optimal solution, some of them are: Capacitated vehicle routing problem, Probability based vehicle routing problem, Vehicle routing problem with pickups and deliveries, Group scheduling problem, Scheduling of nurses, Permutation flow shop problem, Frequency assignment, Redundancy allocation problem.

In a real ant colony, the worker ants use chemical signals called 'Pheromone' to achieve the Co-ordination. When a receiving ant recognizes this odor, its actions develop in such a manner that the ant is now more presumably to track this trail. Initially a colonial cycle is founded with ants collected together with a main goal of exploring for food and hauling it back to the hive. At the beginning they just move around in a disorderly fashion in order to locate some food. However, as they do it, they lay down pheromone trail on the surfaces hence mapping the area. The pheromone acts as an attractant for other ants: in their movements, when they come across such a pheromone, they follow the path it defines rather than being arbitrarily mobile. The amount of the pheromone is pumped out in proportion to the level of its concentration which gradually denatures making it less and less appealing to the ants. After the execution of some period of time, the path which has shown better performance will be further strengthened while the path which has weak performance will be eliminated. And over time, the ants will identify the best route (Von Thienen et al. 2014).

As mentioned above, ACO is derived from the natural system of ants; strategies in this model such as collective behavior and collaborative planning and organization as well as integration for seeking and finding a solution are all derived from ants (Nayar, 2021). This is where the idea of ACO is more closely related to pheromones. Real ants in an ant hive use pheromone gradient to show other ants where food sources that are more satisfactory might be found. In ACO this natural phenomenon is extended into a computational model, where instead of real ants the algorithm utilizes “artificial ants” that move over a graph that models the given problem.

At first these approaches propose that each path within the graph is unique in terms of the intensity of discovered pheromones. While moving, ants create pheromone signals by chance and blindly choose their next spot based off it. This trail becomes a memory depiction of their journey, holding considerable sway for the future development of these ants. Most importantly while long paths are strengthened only by a small pheromone amount, better solutions such as short paths have richer pheromone traces deposited on them. This positive feedback loop gradually results in more ants visiting the good solutions hence, exploring and exploiting the search space efficiency.

Two key mechanisms govern the dynamics of pheromones in ACO: update and evaporite. Once the ants have finished foraging, they leave behind chemical cues in the borders discovered in proportion to the potential of their solutions. This reinforcement increases the value of routes with positive outcomes, making them more attractive in future loops. However, the pheromone trails also disappear slowly with time; this helps to break bad loops and search actively for other better solutions.

While other deterministic optimization algorithms work and sometimes get trapped in local optima, ACO has an adaptive model which comprises pheromone trails. Furthermore, the ACO algorithm is an emergent computation paradigm used for solving complex combinatorial optimization problems due to its high flexibility of the framework that depends on the choice of the representation of the optimization problem and the rules in constructing the solutions. Because of the described advantages, it can be used in multiple subjects and domains such as routing problems and scheduling problems (Dorigo & Stützle, 2018).

Scheduling using Ant Colony Optimization

ACO is a bio-inspired complex metaheuristic algorithm which is used to simulate the behaviour of ants while foraging food. Marco Dorigo in early 1990 introduced it and applies to numerous combinatorial optimization problems such as scheduling (Dorigo, 1992). ACO algorithms are like processes in ant colonies where real ants drop pheromones to express themselves and set the best paths; it thus holds the potential for developing approximate or best-solution solutions for challenging scheduling issues (Dorigo & Stützle, 2004).

As it has been highlighted, ACO has been applied in solving different scheduling problem because of its flexibility and robustness. The nature of learning via pheromones and the way that the algorithm improves upon solutions by imitating the laying and following behaviour of ant colonies makes the Ants algorithm particularly effective for dynamic, complex environments (Blum, 2005). Studies have also demonstrated that ACO is capable of generating competitive schedules across numerous scheduling problems such as job shop scheduling, flow shop scheduling, and course timetable scheduling to mention but a few (López-Ibáñez & Blum, 2010).

The Ant Colony Optimization (ACO) method has been effectively used in complicated scheduling problems, including school scheduling (Ge & Chen, 2022). This algorithm uses the behaviour of ant colonies to pick optimal pathways. In a study on an ACO-based solution for class scheduling, the algorithm's effectiveness in addressing the University Class Scheduling Problem (UCSP) versus Genetic Algorithms (GA) was demonstrated (Li et al., 2023). Furthermore, integrating ACO with selective probability (ACOSP) has resulted in enhanced performance by lower computing costs when solving severely limited UCSPs.

Python for Backend Scripting

Python is an interactive and high-level programming language that supports several programming paradigms, procedural, object-oriented, and functional programming paradigms as pointed by Lutz (2013). This language is highly readable, which is an essential factor when working on programs because it saves a lot of time and prevents possible errors. Furthermore, we would get to know that Python supports a standard library and a large number of third-party libraries, which includes web development, data analysis, machine learning and several other tools or modules. Since Python is supported by a large and vibrant community, changes and updates are constant and simple to acquire, resources, libraries, and solutions are readily available (Lutz, 2013). However, as an interpreted language, Python will generally run faster than C or Java due to slower execution times which may take their toll in high usage applications. Moreover, the Python's incorporated capability, known as Global Interpreter Lock (GIL) can hamper the execution of programs based on multi-threading but it can be alleviated by use of multi-processing or one can opt for implementations like Jython or IronPython (Beazley, 2013).

Flask for Backend Web Framework Development

As you already know, Flask is a lightweight web structure implemented in Python. It is compact and easily extensible and by this reason offers the developers the freedom to choose only the parts and subroutines they need while at the same enabling them to develop intricate web applications if required. They attribute Flask's simplicity and extensibility for development; thus the framework is to suit for both the novices and the professional developers (Grinberg, 2018). Flask also has a simple architecture and is lightweight, which is one of the strong sides of the Framework since it offers a simple core moment which can be easily extended with help of existed extensions. Further, Flask gives freedom from the strict project-based framework and dependency which enable the developer to structure his work in the most suitable way. In addition, Flutter on the hand enjoys extensive documentation and the presence of a large community which means that it is easy to get help and a lot of resources on the web (Ronacher, 2010). However, some downside to Flask is also notable. It is not bundled with as many tools as more advanced systems like Django, and some additional tweaking is necessary for large-scale SARGE applications. Further, the lightweight nature of Flask can be disadvantageous for very large applications, as requirements and application add-ons must be incorporated correctly (Grinberg, 2018).

React for Frontend Development

React is a front-end library for building reusable UI components created by Facebook for its web applications including single-page apps. It allows the developers to construct the UI components, which is inherently reusable and can take proper care of the state besides rendering the views whenever there is a change in data (if needed) (Krause, 2016). The concept of building up the application using components is a concept of React application development because it enables decoupling of the application and also makes the modular structure easier to maintain. Virtual DOM can be utilised for the purpose of limiting direct communication with the DOM, as well as adapting it for efficient states modification and rendering when the state is changed. Same like it has a big community and around it there are services which can offer some tools, libraries or extensions to React (Krause, 2016). However, it is important to point out that React also has its drawbacks – the first one of which is the fact that a person who is developing an application with React needs to comprehend relatively complex language to utilize such notions as JSX, the life cycle of the component, and state. However, due to the nature of fast app development using React js, there are always some features that maybe added or changed frequently and this makes developers continuously adopt to new features and possibly better way of doing things considering the fact that the applications being developed are usually complex (Griffiths, 2017).

SQLAlchemy for Database ORM

For Python, SQLAlchemy is an ORM and SQL toolkit. Designed for efficient and high-performing database accesses, it comes with a complete set of well-known enterprise-level persistence patterns (Bayer, 2012). For varied abstraction levels, developers can code in different levels of low-level SQL expression language or at the high level using ORM. It works with different database backends because it is not dependent on any specific one making it easy to move from one database to another when required. Moreover, SQLAlchemy incorporates advanced characteristics like connection pooling as well as lazy loading and sophisticated query construction (Bayer, 2012). However, its extensive feature set makes it complex hence hard for novices to comprehend and use effectively. On the other side, compared to raw SQL queries which are faster; ORM abstraction introduces some performance overhead hence need optimization (Bayer, 2012).

MySQL for Database Management

MySQL is a relational DBMS that is supported by Oracle and is structured under the OSI model. Web application has embraced it largely due to its reliability, easy installation-launching, and efficiency as highlighted by DuBois, (2013). MySQL is especially designed to provide high intensity of SELECT operations and thus it is a perfect match to web-based applications. It is easy to use and comes with tons of documents and forums for users to consult and is flexible capable of managing large databases and supporting high traffic website (DuBois, 2013). However, MySQL has some sort of drawbacks: for example, not all the complex functions, which are provided by other RDBMS like PostgreSQL, are supported; MySQL also does not fully conform to SQL standards in many cases. Moreover, there are also some disadvantages of using

MySQL, including that MySQL tends to slow down notably when facing high concurrent write operations (DuBois, 2013).

Conceptual Model of the Study

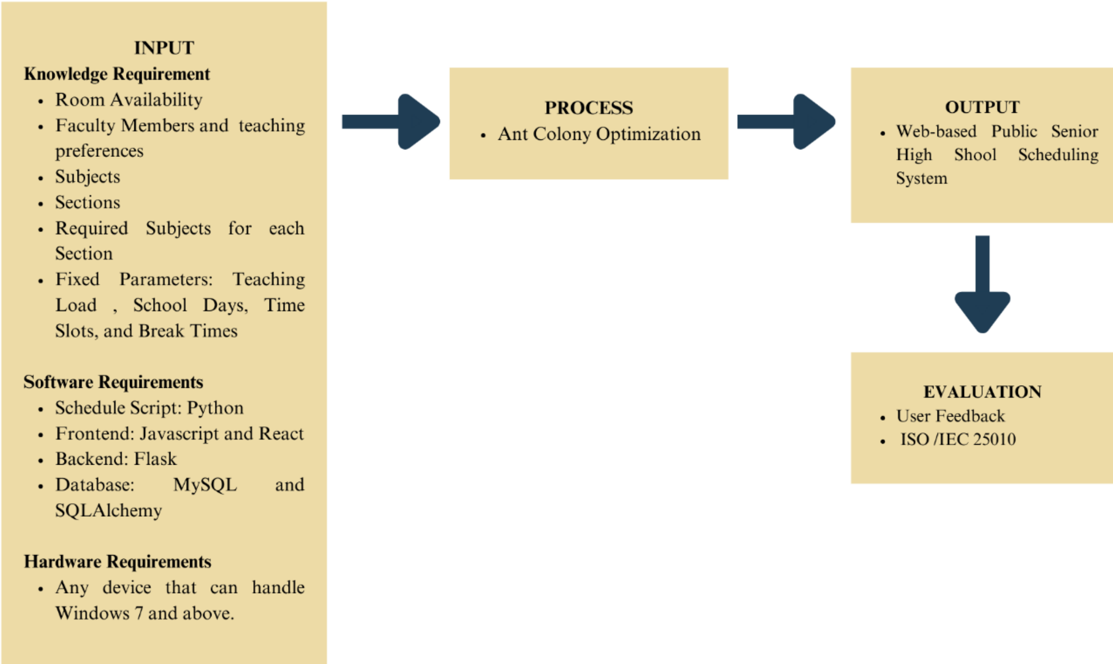


Figure 1. *Conceptual Model of schEDU*

Input

The input section includes three main categories: Knowledge Requirements, an understanding of the field; Software Requirements, the specific software the bot will need to handle; Hardware Requirements, the specific physical equipment the bot must be built with. The Knowledge Requirements include: The Relationship between the Arts & Humanities and Science & Technology; The Nature of Science; The Scientific Method; Advances in Technology; The Internet; The Global Economy; Multiculturalism; Exploration; Problematology; Biodiversity; Intellectual Property; Education and Technology; Education; Human Resources Development; Information Technology; The Future. The availability of the rooms and the schedules of each of the rooms are well detailed in the room availability. From here contain all the information of all faculty members including their available time and modality preferred in teaching the students. tents, a chronological list of subjects that have to be scheduled taking into account the curricular hour allocation is provided in the subjects' part. Student sections are defined in terms that cover the number of students enrolled in under the individual needs and learning ability. The regular courses that every section is supposed to cover within the one semester is known as specified subjects for each section. Academic teaching may not allow a teacher to teach for more than six hours a day due to tight teaching schedules. The days that can be scheduled are weekdays, which are from Monday through Friday if the system is to represent school days. The available schedule slots are called time slots, and they run from 7: This means that the restaurant will operate from 00:00 AM to 1500:00 or 03:00 PM on a particular day of the week. Break times are set times between 9: Half past 3:00 pm and 10:00 am, 12:30 pm and 1:00 pm are some of the times where classes cannot be planned down to the minute.

Process

This is the strategy that has been used in developing the schedules and it is referred to as the Ant Colony Optimization (ACO). Through systematic local search and by means of pheromone trails, which point out the desirability of certain assignments, this program constructs schedules and refines these schedules successively. ACO at the time of scheduling evaluates multiple aspects like instructor's preference, curriculum, and the least number of conflicts. Finally, the optimization procedure ensures that only efficient and effective schedules have been created out of the numerous possible schedules, not negating any restriction or preference.

Output

The remodelled social arrangements are: A web-based system of scheduling for all the public senior high schools. This technique makes an ideal timetable where subject-teacher--class allocations per section are made. Effectively, the schedule generated by an ACO model direction takes the instructor preferences into consideration while at the same time seeking to compose the teaching loads under equal teaching space resources while perusing students need on education as well.

Evaluation.

In the Evaluation tab, there is information of the usage of the objective function defined to assess the quality of the final schedule. Opinions from the users—IT specialists, administrators, teachers, and students—are incorporated to assess the created schedule's appropriateness and significance. From this, it can be seen that for the purpose of being objective in analysing how efficient the scheduling system is, it also collects and provides quantitative results such as the average of student travel time, the percentage of room usage and number of conflicts. It shows how initial results or real-time data comparing the ACO-based scheduling system to other scheduling approaches or prior manual scheduling systems look like. Last, flexibility measures how the created system copes with changes in inputs, be it static or dynamic, such as changes in availability of rooms or the courses within the learning institution.

Operational Definition of Terms

Ant Colony Optimization (ACO) - Ant Colony Optimization is a nature-inspired optimization process grounded on the foraging ways of ants. It is a metaheuristic algorithm that aims to find the optimal solution to a given problem by simulating the collective intelligence and cooperation observed in ant colonies. ACO is commonly applied to combinatorial optimization problems, such as the Traveling Salesman Problem.

Break Times - Fixed break times where no classes can be scheduled (9:30 am – 10:00 am and 12:30 pm – 1:00 pm).

Foraging - Foraging concerns the behaviour of seeking for and acquiring food or resources in the environment. This behaviour is commonly observed in various animal species, including insects, birds, and mammals. Foraging activities are essential for an organism's survival and reproduction, as they ensure the acquisition of necessary nutrients and energy.

Pheromone - Pheromones are chemical substances produced by animals, including insects like ants, to communicate with others of the same species. In the context of optimization algorithms like Ant Colony Optimization, pheromones are virtual chemical signals left by agents (simulating ants) to influence the way of other agents in the system. The intensity of pheromones affects the possibility of selecting a specific path or solution.

Preferred Subjects – Faculty members preferred subject to teach.

Room Availability - List of all available rooms

Required Subjects - Mandatory subjects that each section needs to take throughout the semester.

Senior High School - Senior High School refers to the last two or three years of secondary education, typically for students aged 15 to 18 years old. It follows junior high school and precedes tertiary education. Senior high school is where students have the option to choose specific field of concentration that is aligned with their interests and career goals.

Teaching Load – The number of hours a teacher teaches a class daily.

Time Slots – Available short time spans in a day.

Chapter 3

METHODOLOGY

Project Design

This study proposes the usage of the ant colony optimization (ACO) algorithm in a class scheduling system, for an efficient and optimized solution generation while adhering to the specified constraints inputted by the user. The system takes the following inputs to construct an optimize schedule: room availability, teachers' subject preferences, and the required subjects for each section throughout the semester. The researchers employed fixed parameters: a 6-hour maximum teaching load per day, school days from Monday to Friday, time slots from 7:00 am to 3:00 pm, and 30-minure break times at 9:30 am and 12:30 pm.

ACO is a metaheuristic optimization technique influenced by how ants seek food. Ants leave pheromone trails as they explore paths, and these trails will be the guide for other ants to follow. In the context of class scheduling, ants represent potential solutions, and pheromone trails represent the desirability of a particular scheduling decision.

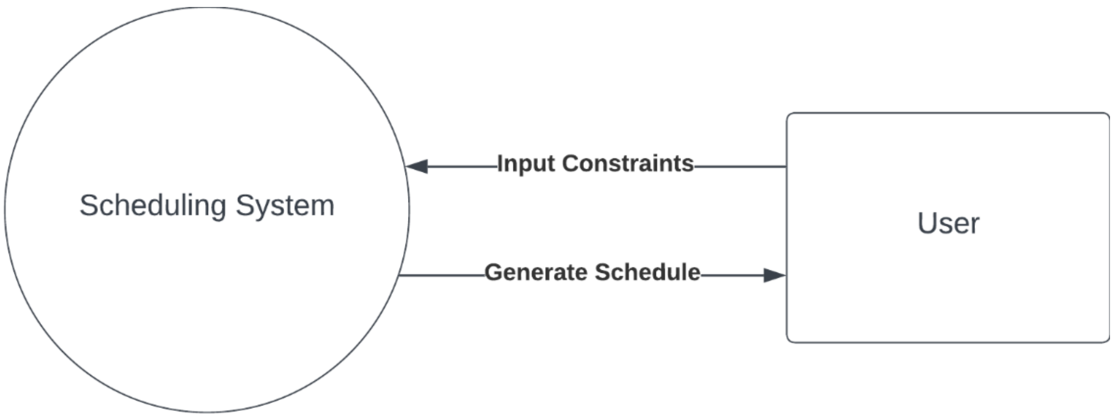


Figure 2. *Context Diagram*

The high-level communication between a user and the scheduling system is depicted in Figure 2. Constraints, such as topic requirements, classroom sizes, and instructor availability, are entered by the user into the scheduling system. The scheduling system analyses the data and creates a school schedule based on these limitations. After that, the user receives the generated schedule back, giving them the opportunity to examine and adjust it. The essential information flows between the user and the scheduling system are shown in this figure.

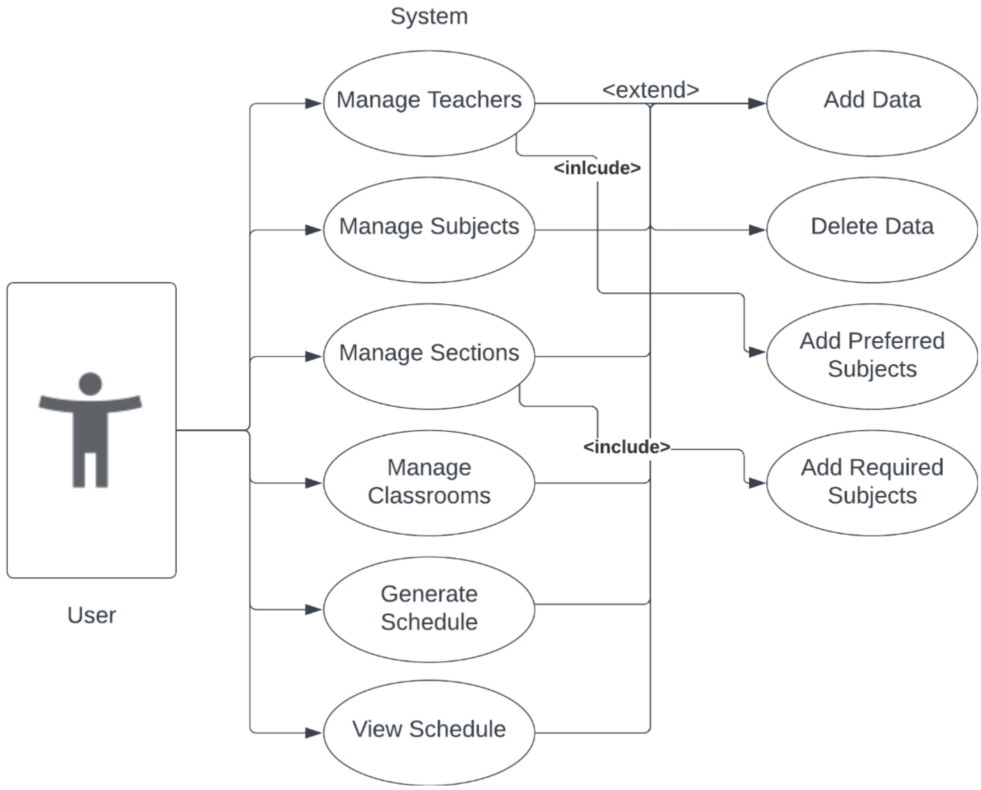


Figure 3. *Use Case Diagram*

Figure 3 highlights important features while emphasizing user interactions with a school scheduling system. Teachers, subjects, sections, and classrooms can all be managed by the user. While managing subjects is adding desired subjects and removing data, managing teachers also requires adding and deleting data. Adding necessary and desired subjects is part of managing sections. The user creating and viewing the schedule is also depicted in the diagram. Certain functionalities, such as deleting data while managing instructors or subjects, are always present or can be extended to other tasks, as shown by the relationships between use cases (e.g., adding data when managing teachers). This arrangement guarantees thorough resource management and efficient schedule creation.

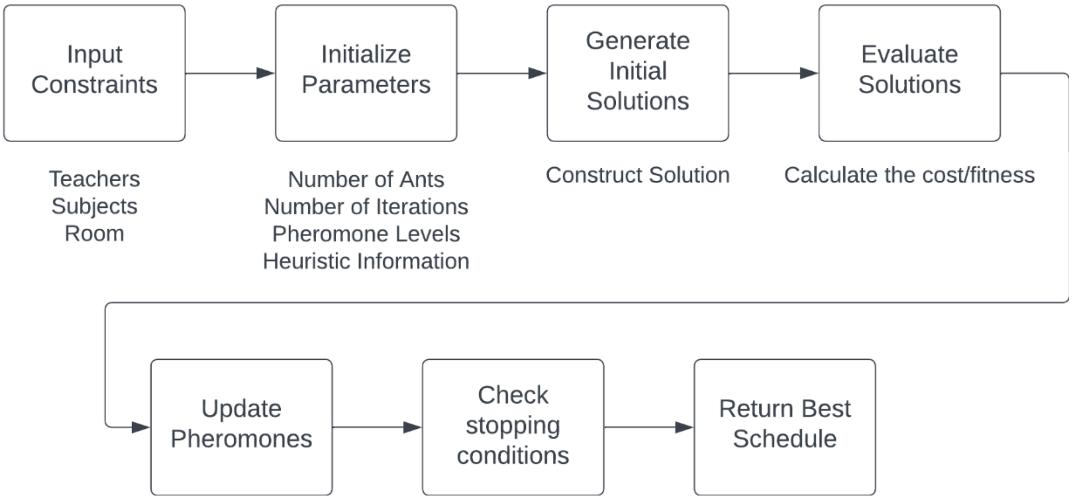


Figure 4. *Ant Colony Optimization Flowchart*

Figure 4 describes a scheduling system that considers constraints such as teachers' availability, subjects, and rooms, in order to produce optimal schedules using Ant Colony Optimization (ACO). First, these limitations must be entered. Next, the ACO algorithm's parameters must be initialized. Ants then create preliminary schedules, which are assessed for overall quality and conformance to the limitations. Future iterations are guided by updated pheromone levels, whereby superior schedules are assigned higher levels and previous pathways' effect is diminished through evaporation. The optimal schedule is produced once the procedure is repeated until a stopping condition is satisfied, such as completing the allotted number of repetitions or arriving at a workable solution.

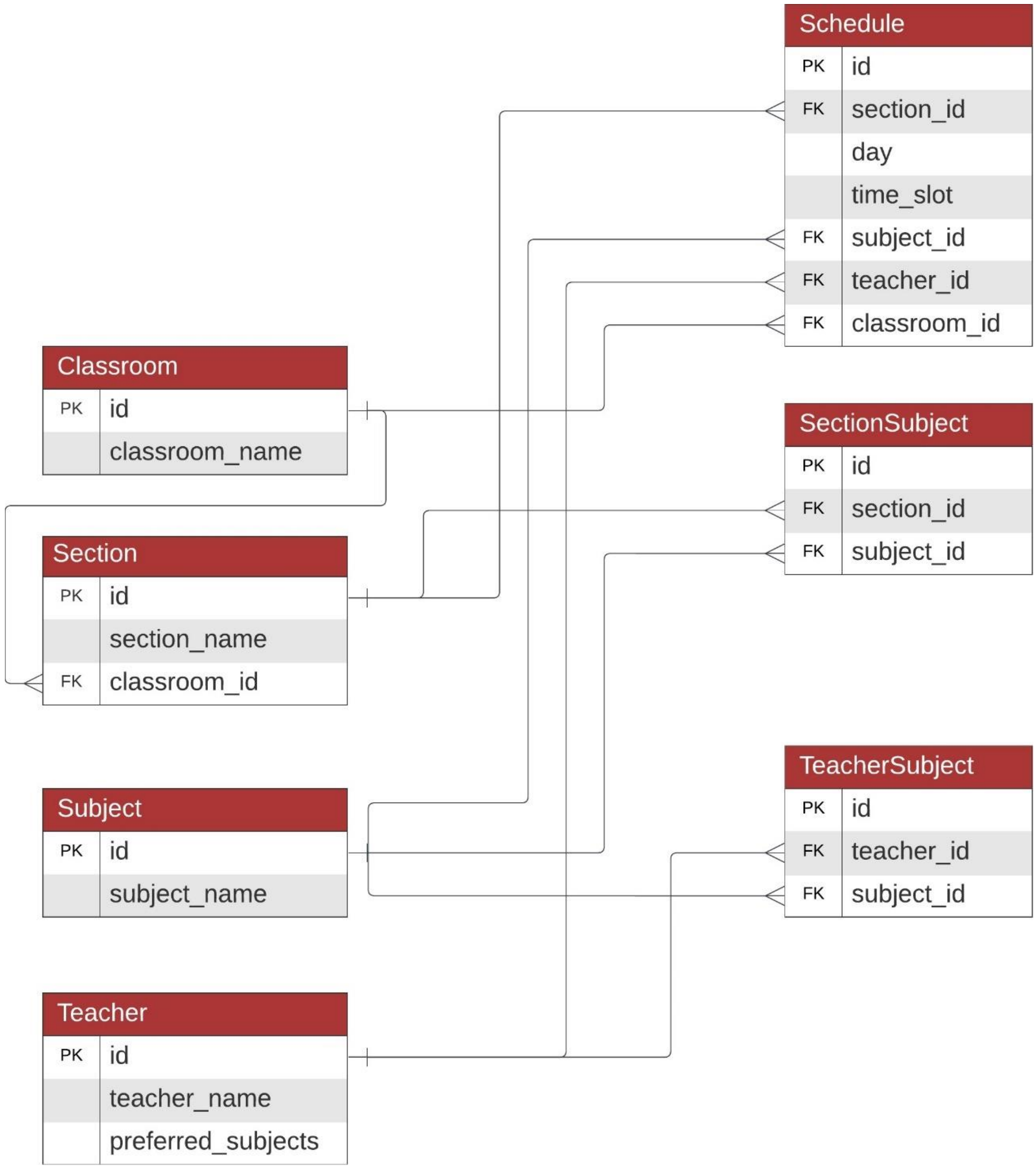


Figure 5. *Entity-Relationship Diagram*

Figure 5 illustrates the Entity-Relationship Diagram (ERD) for the schEDU system. It lists seven interconnected entities that represent the essential components of the scheduling process: Classroom, Section, Subject, Teacher, TeacherSubject, SectionSubject, and Schedule.

Table 1

Classroom

Field Name	Data Type
id	int
classroom_name	string

The Classroom entity contains unique identifiers, names of classrooms, and has a one-to-one relationship with the Section entity.

Table 2

Subject

Field Name	Data Type
id	int
subject_name	string

The Subject entity represents the various courses offered in the school curriculum. Each subject includes attributes such as the subject name and duration. This entity is crucial to the scheduling process, as the SectionSubject and TeacherSubject associative entities form the relationships between the Section and Teacher entities.

Table 3

Section

Field Name	Data Type
id	int

section_name	string
classroom_id	int

The Section entity represents attributes like section names and references to their assigned classrooms. The SectionSubject associative entity establishes a many-to-many relationship with the Subject entity, as each section requires multiple subjects.

Table 4

Teacher

Field Name	Data Type
id	int
teacher_name	string

The Teacher entity holds unique identifiers for each teacher. Teachers can have a maximum of three preferred subjects, with the TeacherSubject associative entity establishing a many-to-many relationship with the Subject entity.

Table 5

SectionSubject

Field Name	Data Type
id	int
section_id	int
subject_id	Int

The SectionSubject entity shows the section-subject relationship – how the sections are linked to their required subjects.

Table 6

TeacherSubject

Field Name	Data Type
id	int
teacher_id	int
subject_id	Int

The TeacherSubject entity facilitates managing the relationships between subjects and teachers, linking teachers to their preferred subjects, respectively.

Table 7

Schedule

Field Name	Data Type
id	int
section_id	int
day	string
time_slot	string
subject_id	int
teacher_id	int

classroom_id	int

The Schedule entity displays a detailed scheduling information. This comprehensive structure ensures that all aspects of the school scheduling system are interconnected and can be managed effectively.

Project Development

The development of schEDU follows the Agile Software Development model, focusing on continuous improvement. User involvement is crucial for refining the system, as users provide feedback and ideas to enhance performance.

Requirements

To initiate the process, essential user requirements for the scheduling system, such as devices and features, will be gathered. The emphasis is on prioritizing frequently used aspects, while less critical details can be addressed post-implementation. Basic features are incorporated initially.

Design

After identifying requirements, the software design involves two key operations: interface and architectural. The optimal framework, algorithm, and programming language are chosen for accuracy and user experience. A preliminary model showcases the user interface and architectural design, with data storage considerations.

Development

Coding and software design take place in this phase, aligning with approved requirements and prototypes. The development phase continues until the program adheres to conceptual plans, with errors corrected.

Testing

Prior to deployment, a thorough testing phase ensures an error-free and debugged system. Various tests are conducted to verify consistent performance and accurate results.

Deployment

Following testing, the system undergoes beta testing with real users to gather feedback. This stage addresses any issues discovered during beta testing.

Review

Before commercialization, a re-evaluation ensures the system's functionality and readiness for commercial use.

Delivery

Upon completing the re-evaluation, the system is ready for market use.

Feedback

Once in commercial use, user feedback shapes future versions of the system.

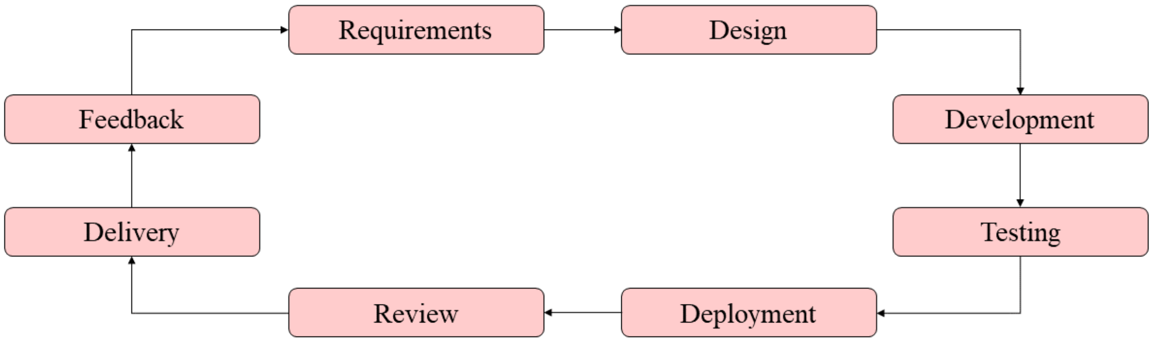


Figure 6. *Agile Software Development*

Operation and Testing Procedure

The scheduling system should be cleared for any errors that could harm both the software itself and the users' experience. To ensure that such problems would not arise, a testing phase will be conducted:

1. The system functions smoothly as a stand-alone application.
2. Upon successful registration, users input information to create a schedule.
    a. Select scheduling system type.

b. Enter curriculum, faculty availability, classroom availability, and capacity.

4. The system generates an optimized schedule using Ant Colony Optimization.
5. The generated schedule is displayed and available for download.

Evaluation Procedure

The evaluation tool, based on ISO/IEC 25010 standards, is employed to assess system performance in quality management, acquisition, and maintenance. The evaluation process involves:

1. Creating a demonstration video of the system's functionality.
2. Gathering knowledgeable respondents in software development.
3. Distributing the demonstration video and evaluation form to respondents.
4. Respondents evaluate the scheduling application using provided sheets and a 4-point Likert scale.
5. Data is processed to calculate mean ratings for comprehensive evaluation.

Table 8

Scale	Descriptive Rating	Range
4	Highly Acceptable	3.4 - 4.0
3	Very Acceptable	2.6 - 3.3
2	Acceptable	1.8 - 2.5
1	Not Acceptable	1.0 - 1.7

Chapter 4

RESULTS AND DISCUSSION

In this chapter, the project's description, structure, capabilities, and limitations are discussed. Also included are the analyses of test and evaluation results.

Project Description

This study focused on developing schEDU, which utilizes Ant Colony Optimization (ACO) algorithm in automating class scheduling processes. The web application is intended to benefit scheduling personnel of public senior high schools. The researchers were able to develop schEDU using Python, JavaScript, Flask, React, and Git.

The system's interface presents several modules including *Subject Module, Teacher Module, Classroom Module, Section Module, Summary Module, and View Modules*. These modules enable the user to integrate the required data to utilize the web application – the total number of sections enrolled, the current semester's subjects, each teacher's preferred subjects, and the list of available classrooms.

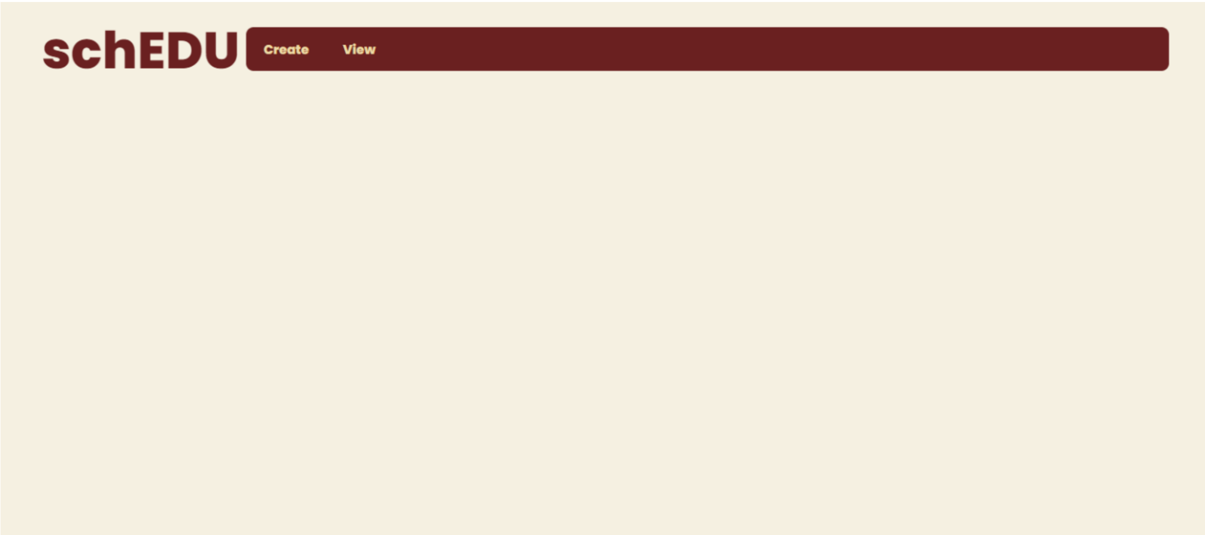


***Figure** 7* schEDU Main Screen

Project Structure

schEDU focuses on the following modules: *Subject Module, Teacher Module, Classroom Module, Section Module, Summary Module, and View Modules*.

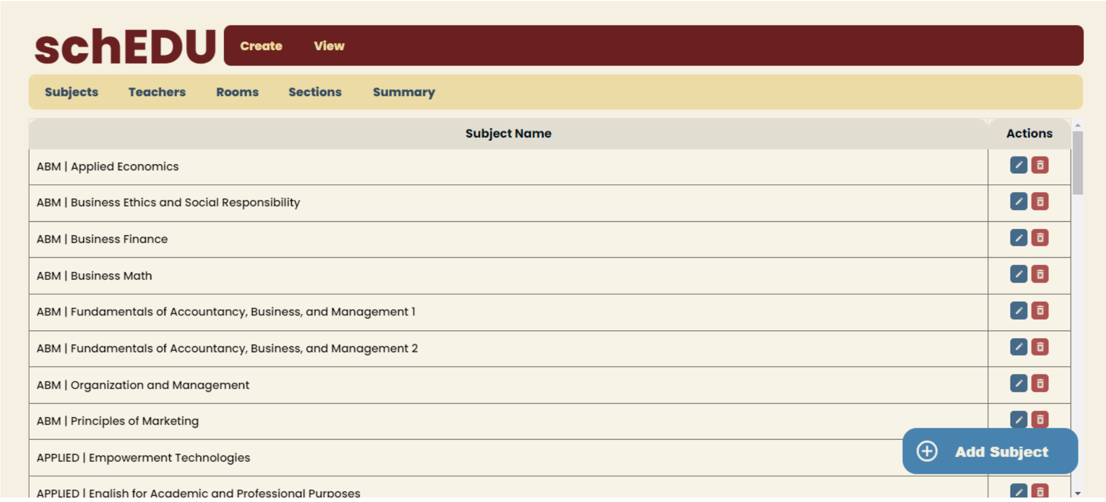


Figure 8 Subjects Module

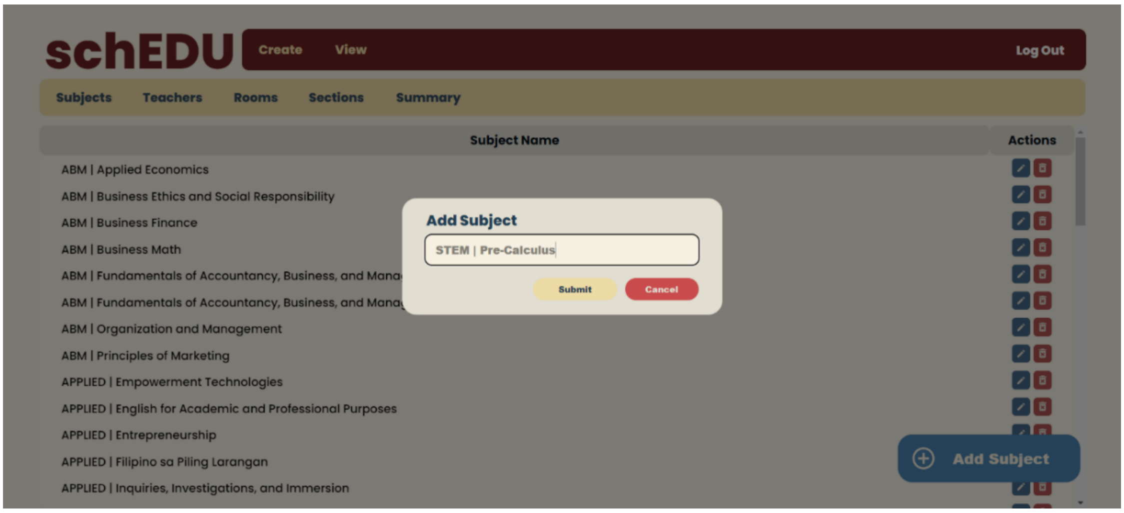


Figure 9 Subject Module (Adding subjects)

Figures 10 and 11 show schEDU's subject module, where all subjects under the DepEd's senior high school curriculum are initially displayed. The subjects are shown based on three categories: specialized subjects for each strand, core subjects that apply to all strands, and applied subjects that apply to all tracks. The user can update and delete these pre-existing subjects, as well as add new subjects in case of changes in the DepEd curriculum.

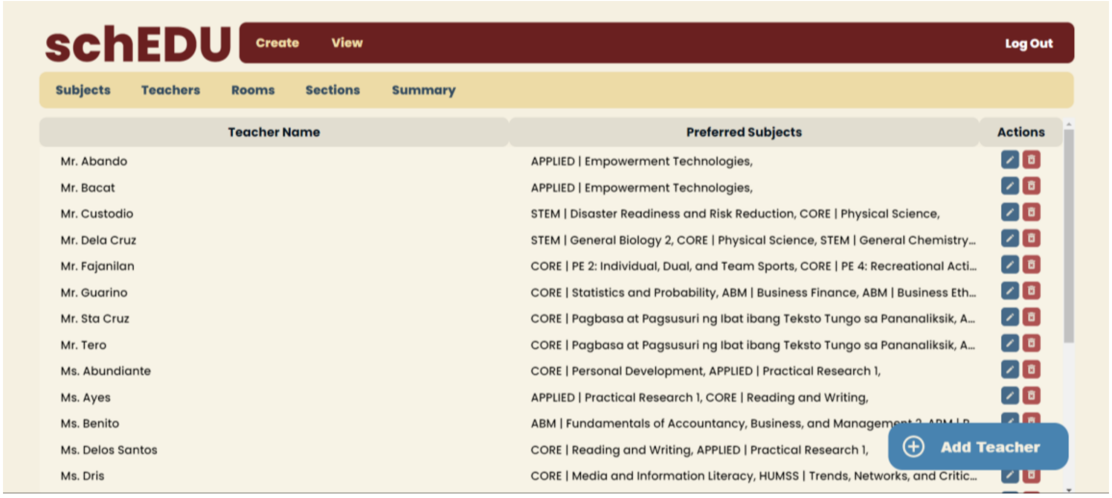


Figure 10 Teachers Module

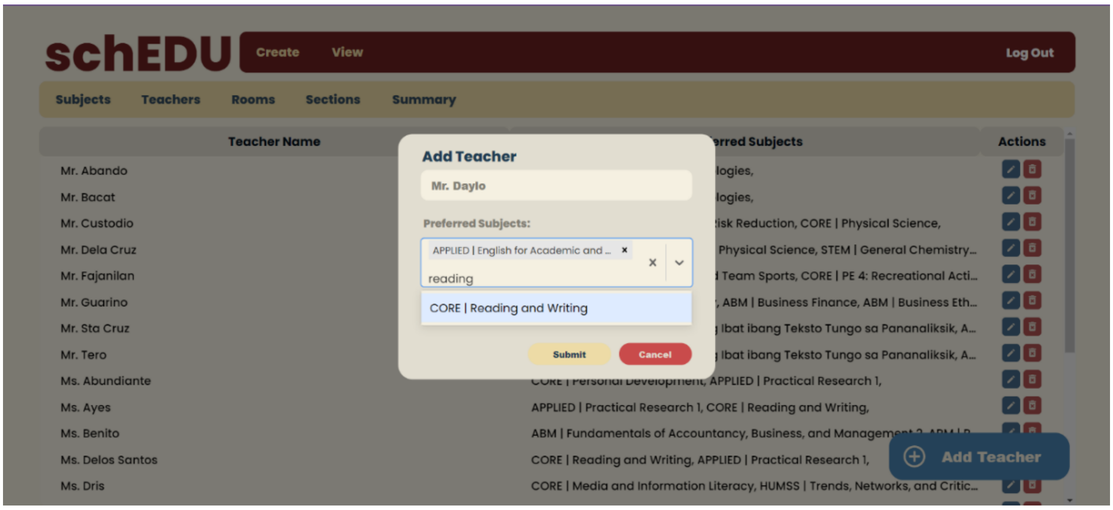


Figure 11 Teachers Module (Adding teachers)

Figures 12 and 13 show schEDU's teacher module, where the user will input teachers' names along with their preferred subjects. The user can only select up to three preferred subjects for each teacher, as the mandate allows a maximum of only three preps per semester. Also available in the teacher module are the add, update, and delete features.

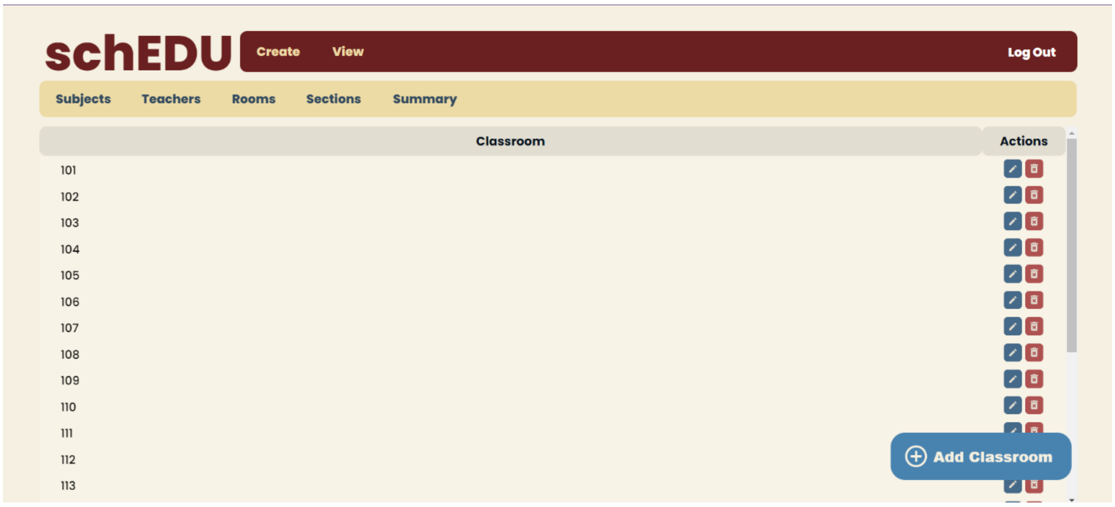


Figure 12 Classrooms Module

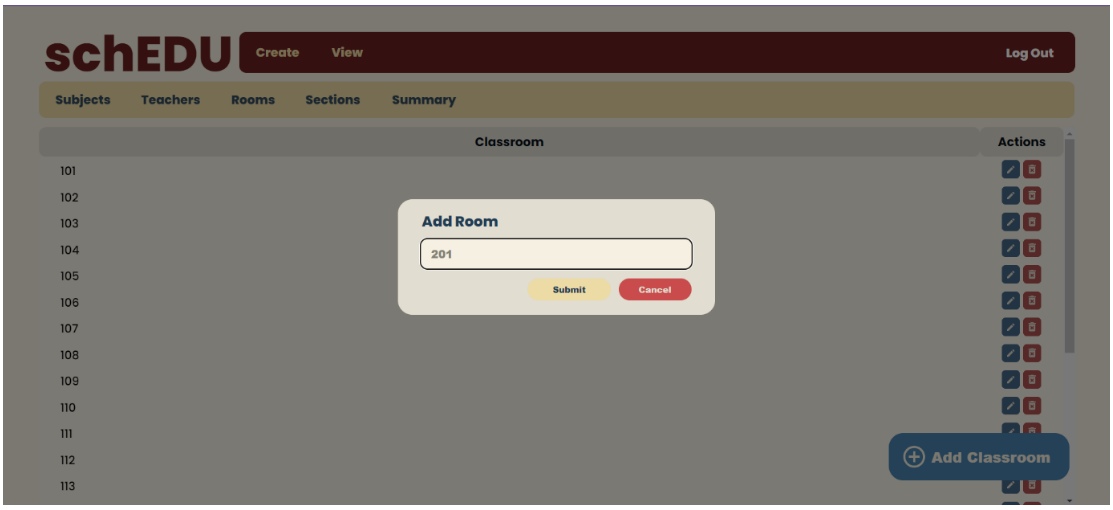


Figure 13 Classrooms Module (Adding classrooms)

Figures 14 and 15 show schEDU's classrooms module, where all classrooms available for class designation are inputted. The add, update, and delete features are also available in this module.

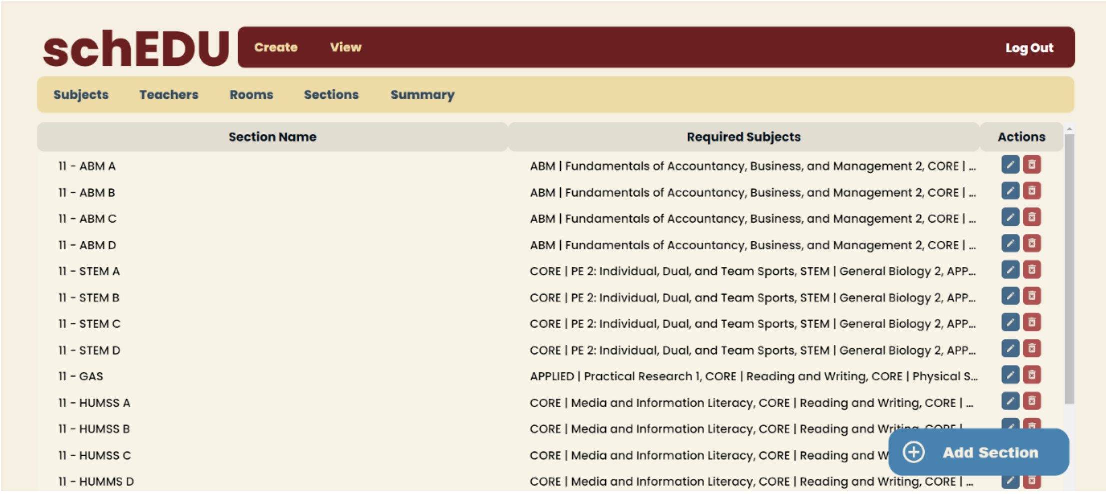
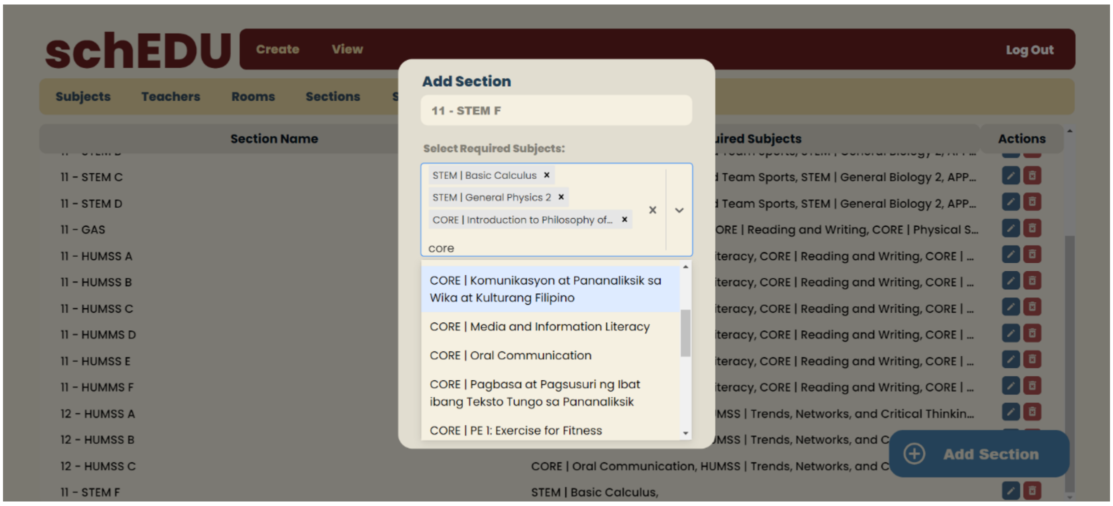


Figure 14 Sections Module

Figure 15 Sections Module (Adding Sections)



Figures 16 and 17 show schEDU's section module, where the user will input the finalized sections along with their required subjects. The user can only select up to nine required subjects for each teacher, as the curriculum allows a maximum of only nine subjects per semester. Also available in this module are the add, update, and delete features.

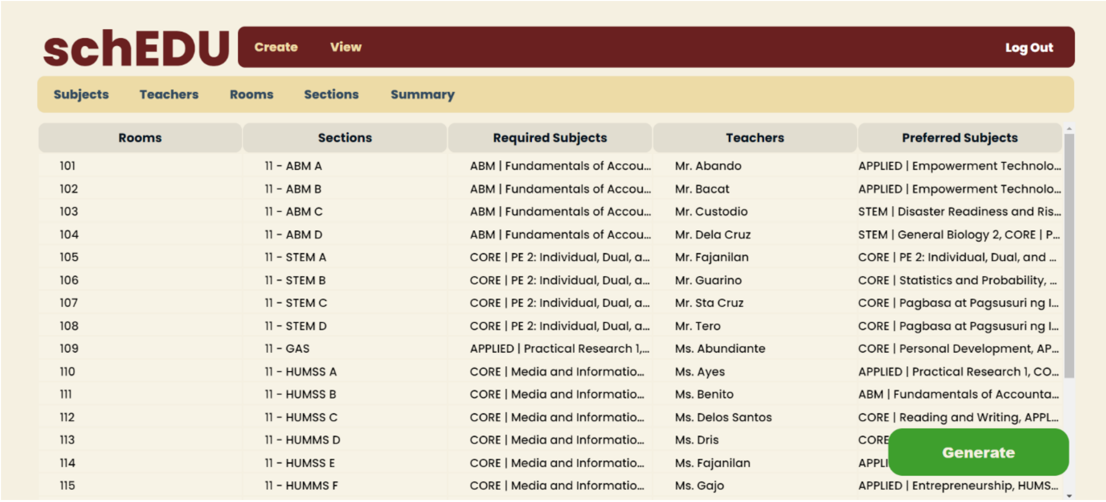


Figure 16 Summary Module

Figure 18 shows schEDU's summary module, where all previous inputs are displayed systematically. This allows the user to review the information before proceeding to generate the schedule. After reviewing, the user will click the 'generate' button to start the scheduling process, redirecting them to the view tab.

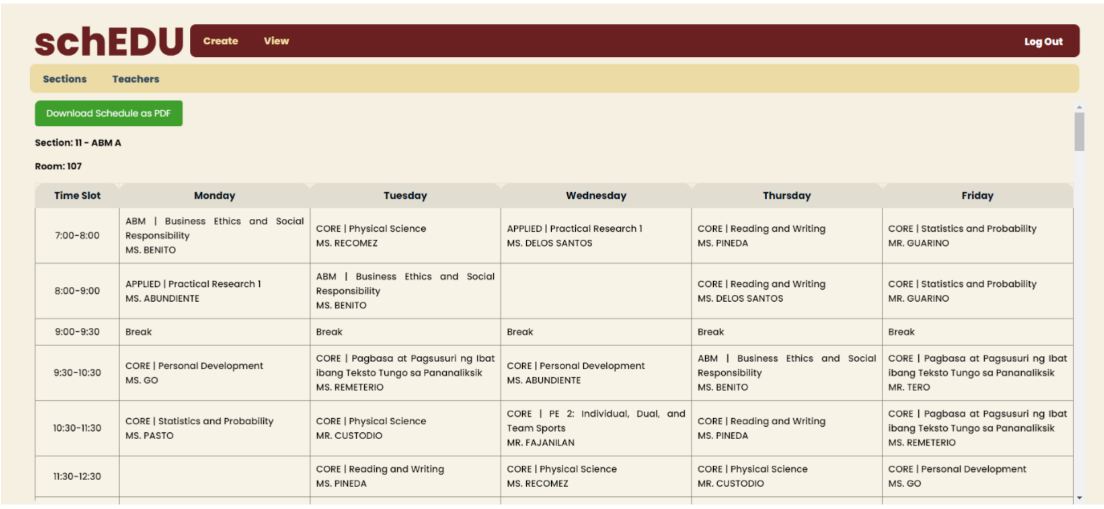


Figure 17 View Module (Section Tab)

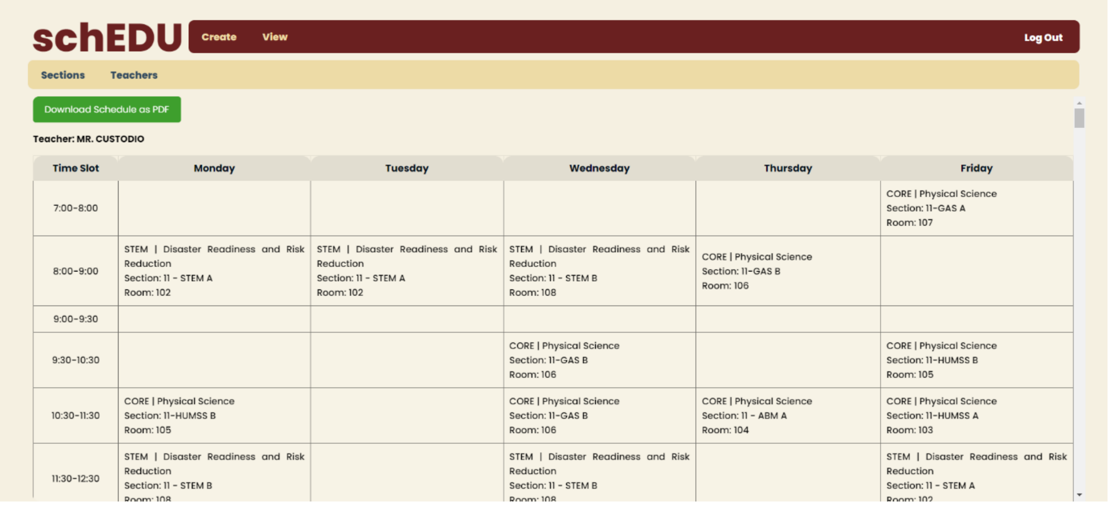


Figure 18 View Module (Teacher Tab)

Figures 19 and 20 show schEDU's view modules, where the generated schedules (based on the user input) are displayed. It can be observed that no classes have conflicting issues, and no teacher or classroom are double-booked. There are two ways to view the output. Under the *sections* tab are the generated class schedules for each section. The generated schedules are presented in a tabular format, with the school days in the column and time slots in rows. Plotted are the classes and teachers assigned in specified time slots. Meanwhile, under the *teacher's* tab are the generated schedules for each teacher. The user can then download pdf version of the schedules afterwards by clicking the download button in their respective tabs.

System Capabilities and Limitations

The following are the system capabilities of schEDU:

a. The web application automates the class scheduling process through manual input of required data: subjects, sections, teachers, and classrooms.

b. The web application allows the user to add, modify, and delete input before proceeding to schedule generation.

c. The web application produces a conflict-free schedule every section, teacher, and classroom input.

d. The web application can display the output in two ways: in the section tab where the schedules generated per section are shown, and in the teacher tab where the teachers' individual schedules are displayed.

e. The web application allows the user to download a pdf version of generated schedules, both available in the *sections* and *teachers'* tabs.

The following are the current system limitations of schEDU:

a. The current system capabilities only support a one-shift class format.

b. The system is currently limited by fixed time slots and break times.

Test Results

The tables below present the steps undertaken to test each module, along with the corresponding results.

Table 9.

Functional Suitability Test Results

Test on modules	Steps undertaken	Test results
Subject Module	1. The user reviews and modifies the pre-defined list of available subjects under the curriculum.	The user successfully updated the subject pool.
Teacher Module	1. The user inputs the teachers' names, along with their three.	The user successfully entered the teachers' names with their corresponding preferred subjects.

Classroom Module	1. The user inputs the classrooms available for class assignment.	The user successfully inputted available classrooms.
Section Module	1. The user creates sections and selects each section’s required subjects from the input subjects.	The user successfully entered up to nine required subjects for each section.
Summary Module	1. After completing and reviewing all the required input, the user will click the generate button.	The system redirects the user to the view tab where the generated schedules are displayed.
View Module	1. The user views the output in two ways: a class view of the schedule in the *sections* tab, and the teacher’s schedule in the *teachers* tab. These schedules are available download in their respective tabs (section and teacher).	The user successfully downloads, in pdf format, the generated schedule.

Table 9 presents the testing procedure and results for the web application's modules. The test results show that the steps undertaken achieved the system's functionality goal for each module.

Evaluation Results

The system was evaluated by 30 respondents, composed of Senior High School scheduling personnel and IT Professionals. The evaluation was conducted following the ISO 25010 standard instrument with the following criteria: functional suitability, performance efficiency, usability, and maintainability.

Table 10.

Summary of Evaluation

Criteria	Mean	Rating
A. Functional Suitability		
Functional Completeness	3.89	Highly Acceptable
Functional Correctness	3.78	Highly Acceptable
Functional Appropriateness	3.89	Highly Acceptable
Criterion Weighted Mean	**3.85**	**Highly Acceptable**
B. Performance Efficiency		
Time Behavior	3.67	Highly Acceptable
Resource Utilization	3.89	Highly Acceptable
Capacity	3.56	Highly Acceptable
Criterion Weighted Mean	**3.71**	**Highly Acceptable**
C. Usability		

Appropriateness Recognizability	3.78	Highly Acceptable
Learnability	3.89	Highly Acceptable
Operability	3.78	Highly Acceptable
User Error Protection	3.67	Highly Acceptable
User Interface Aesthetics	3.78	Highly Acceptable
Accessibility	3.78	Highly Acceptable
Criterion Weighted Mean	**3.78**	**Highly Acceptable**
D. Maintainability		
Modularity	3.78	Highly Acceptable
Analysability	3.67	Highly Acceptable
Modifiability	3.78	Highly Acceptable
Testability	3.78	Highly Acceptable
Criterion Weighted Mean	**3.75**	**Highly Acceptable**

Table 11.

Summary of Responses

Criteria	Total Mean	Rating
Functional Suitability	3.85	Highly Acceptable
Performance Efficiency	3.71	Highly Acceptable
Usability	3.78	Highly Acceptable
Maintainability	3.75	Highly Acceptable
Average	**3.77**	**Highly Acceptable**

Table 11 demonstrates that the system received a Highly Acceptable rating across all criteria. The Functional Suitability criteria got the highest rating, with a weighted mean of 3.85. This means that the system successfully met all the specified objectives and delivered the expected results. On the other hand, even if the Performance Efficiency criteria got the lowest rating among all criteria, it got a weighted mean of 3.71 which was still considered Highly Acceptable.

CHAPTER 5

SUMMARY OF FINDINGS, CONCLUSIONS, AND RECOMMENDATIONS

In this chapter, the summary of findings, conclusions, and recommendations of the study are presented.

Summary of Findings

This study has able to develop a web-based scheduling system for public senior high schools. The system successfully utilized Ant Colony Optimization (ACO) to generate a seamless schedule for one semester. The study's findings shows that the ACO algorithm was successfully implemented to construct efficient schedule solutions based on the required inputs. With this scheduling approach, the issues associated with manual scheduling are reduced. The schEDU web application ensured that each generated schedule was free of conflicts and adhered to senior high school scheduling constraints. Using the ACO algorithm in the system's development proved efficient and effective, producing non-conflicting schedules for all sections and teachers.

Conclusion

The following conclusions were derived from the evaluation of the developed scheduling system using ant colony optimization algorithm:

1. The developed system has the following functionalities:
    - A comprehensive display of all the inputs: subjects, rooms, teachers, and sections.
    - Users can conveniently view, create, add, update, and delete the scheduling inputs.
    - A view for the generated schedule of each teacher.
    - A view for the generated schedule of each section.
    - Ability to export generated schedule in pdf format.

    - The system can create efficient schedule based on the inputs: classrooms, sections, required subjects, teachers and their preferred subjects.

2. The system was developed using the following tools:
    a. Frontend:
        i. HTML
        ii. CSS
        iii. JavaScript
        iv. ReactJS
    b. Back-end:
        i. Python
        ii. Flask
    c. Database Management System:
        i. MySQL
3. The system was evaluated using the ISO/IEC 25010 evaluation tool:
    - Functional Suitability. The respondents rated the system's functional suitability as highly acceptable after meeting all specified objectives and deliver accurate, expected results.
    - Performance Efficiency. The respondents rated the system's functional suitability as highly acceptable after successfully performing within the specified parameters and conditions.
    - Usability. The respondents rated the system's functional suitability as highly acceptable after achieving the study's goal with effectiveness and efficiency.

- Maintainability. The respondents rated the system's functional suitability as highly acceptable after successfully adapting to the environment changes.

Recommendations

Based on the results of this study, the researchers recommend the following for future enhancements:

1. Expand the number of user roles. Include a superuser for management efficiency, as well as faculty and student roles for security. Having these separate user roles would allow more security, for the teachers would only be able to view their schedules and students to view only theirs.
2. Modify the system to have multi-shift, flexible time slot and dynamic break time assignments.
3. Adjust the system's scope to generate schedules for an entire academic year.
4. Modify the system to consider external factors affecting the school calendar, such as holidays.

References

Alkallak, I. N., Shaban, R. Z., & Alnema, Y. H. S. (2022). Intelligent university timetable scheduling system using sudoku grid with magic square. Bulletin of Electrical Engineering and Informatics, 11(3), 1526–1534. https://doi.org/10.11591/eei.v11i3.3694

Austero, L. D., Sison, A. M., Matias, J. B., & Medina, R. P. (2022). Optimizing Faculty Workloads and Room Utilization using Heuristically Enhanced WOA. *International Journal of Advanced Computer Science and Applications/International Journal of Advanced Computer Science & Applications*, *13*(11). https://doi.org/10.14569/ijacsa.2022.0131164

Awan-Ur-Rahman. (2021, December 14). Introduction to Ant colony optimization (ACO) - Towards Data Science. *Medium*. https://towardsdatascience.com/the-inspiration-of-an-ant-colony-optimization-f377568ea03f

Babaei, Hamed, et al. "A Survey of Approaches for University Course Timetabling Problem." *Computers & Industrial Engineering*, vol. 86, 1 Aug. 2015, pp. 43–59, www.sciencedirect.com/science/article/pii/S0360835214003714, https://doi.org/10.1016/j.cie.2014.11.010. Accessed 1 Mar. 2020.

Bayer, M. (2012). SQLAlchemy. https://www.sqlalchemy.org

Beazley, D. M. (2013). *Python Essential Reference* (4th ed.). Addison-Wesley.

Bellardo, H. A. (2010). PREFERENCE DRIVEN UNIVERSITY COURSE SCHEDULING SYSTEM. Www.academia.edu. https://www.academia.edu/884266/PREFERENCE_DRIVEN_UNIVERSITY_COURSE_SCHEDULING_SYSTEM

Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*, 2(4), 353-373. https://doi.org/10.1016/j.plrev.2005.10.001

Branka Dimitrijević, Dušan Teodorović, Vladimir Simić, & Milica Šelmić. (2012). Bee Colony Optimization Approach to Solving the Anticovering Location Problem. *Journal of Computing in Civil Engineering*, *26*(6), 759–768. https://doi.org/10.1061/(asce)cp.1943-5487.0000175

Chen, X., Yue, X.-G., Yi, R., Ainur Zhumadillayeva, & Liu, R. (2021). Design and Application of an Improved Genetic Algorithm to a Class Scheduling System. *International Journal of Emerging Technologies in Learning/International Journal: Emerging Technologies in Learning*, *16*(01), 44–44. https://doi.org/10.3991/ijet.v16i01.18225

Chen, Y., Mahmonir Bayanati, Ebrahimi, M., & Sadaf Khalijian. (2022). A Novel Optimization Approach for Educational Class Scheduling with considering the Students and Teachers' Preferences. *Discrete Dynamics in Nature and Society*, *2022*, 1–11. https://doi.org/10.1155/2022/5505631

Childers, E. A. (2018). Effects of Class Scheduling and Student Achievement on State Testing. In *Effects of Class Scheduling and Student Achievement on State Testing*. https://eric.ed.gov/?id=ED592046

Colorni, A., Dorigo, M., & Maniezzo, V. (1994). Ant colony system for job-shop scheduling. *Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL)*, 34(1), 39-53.

Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C., & Stummer, C. (2004). Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection. *European Journal of Operational Research*, 171(3), 830-841. doi:10.1016/j.ejor.2004.09.006

Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: A new meta-heuristic. *Proceedings of the 1999 Congress on Evolutionary Computation*, 2, 1470-1477. doi:10.1109/CEC.1999.782657

Dorigo, M., & Stützle, T. (2018). Ant Colony Optimization: Overview and recent advances. In *International series in management science/operations research* (pp. 311–351). https://doi.org/10.1007/978-3-319-91086-4_10

Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*. MIT Press.

Duan, Y., & Lu, W. (2021). Automatic course scheduling system in universities based on hybrid Genetic-ANT Colony algorithm. *Journal of Physics, 2066*(1), 012079. https://doi.org/10.1088/1742-6596/2066/1/012079

DuBois, P. (2013). *MySQL*. Addison-Wesley.

Foong, W. K., Maier, H. R., & Simpson, A. R. (2008). Power plant maintenance scheduling using ant colony optimization: an improved formulation. *Engineering Optimization*, *40*(4), 309–329. https://doi.org/10.1080/03052150701775953

Gawali, M. B., & Shinde, S. K. (2018). Task scheduling and resource allocation in cloud computing using a heuristic approach. Journal of Cloud Computing, 7(1), 1-16

Ge, R., & Chen, J. (2022). Analysis of College Course Scheduling Problem Based on Ant Colony Algorithm. *Computational Intelligence and Neuroscience*, *2022*, 1–9. https://doi.org/10.1155/2022/7918323

Griffiths, J. (2017). *Learning React: Functional Web Development with React and Redux*. O'Reilly Media.

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

Hakim, J., Rekiek, S., & Reklaoui, K. (2022). Solving the Job Shop Scheduling Problem by the Multi-Hybridization of Swarm Intelligence Techniques. *International Journal of Advanced Computer Science and Applications*, *13*(7). https://doi.org/10.14569/ijacsa.2022.0130788

Hamed Babaei, Jaber Karimpour, & Amin Hadidi. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering, 86,* 43–59. https://doi.org/10.1016/j.cie.2014.11.010

Jagdish Chand Bansal. (2018). Particle Swarm Optimization. *Studies in Computational Intelligence*, 11–23. https://doi.org/10.1007/978-3-319-91341-4_2

Jansen Wiratama, Santoso, H., & None Clairence. (2023). Developing a Class Scheduling Mobile Application for Private Campus in Tangerang with the Extreme Programming (XP) Model. *G-Tech*, *7*(2), 484–493. https://doi.org/10.33379/gtech.v7i2.2288

Jayalakshmi, N., & Mantha, S. S. (2021). Intelligence methods for data mining task. In *Elsevier eBooks* (pp. 21–39). https://doi.org/10.1016/b978-0-12-820601-0.00007-0

Jiang, W., Shen, Y., Liu, L., Zhao, X., & Shi, L. (2021). A new method for a class of parallel batch machine scheduling problem. *Flexible Services and Manufacturing Journal*, *34*(2), 518–550. https://doi.org/10.1007/s10696-021-09415-w

Kaveh, A. (2014). Particle Swarm Optimization. *Springer EBooks*, 9–40. https://doi.org/10.1007/978-3-319-05549-7_2

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8(1), 687-697 doi:10.1016/j.asoc.2007.05.007

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942-1948. doi:10.1109/ICNN.1995.488968

Krause, J. (2016). *React Quickly: Painless web apps with React, JSX, Redux, and GraphQL*. Manning Publications.

Lei, D., & Su, B. (2023). A multi-class teaching–learning-based optimization for multi-objective distributed hybrid flow shop scheduling. *Knowledge-Based Systems*, *263*, 110252–110252. https://doi.org/10.1016/j.knosys.2023.110252

Li, Y., Chen, D., & Hao, J. (2023). Cloud workflow scheduling optimization research based on ant colony algorithm. *Academic Journal of Computing & Information Science*, *6*(5). https://doi.org/10.25236/ajcis.2023.060502

Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.

López-Ibáñez, M., & Blum, C. (2010). Beam-ACO for the travelling salesman problem with time windows. *Computers & Operations Research*, 37(9), 1570-1583. https://doi.org/10.1016/j.cor.2009.09.004

Marinakis, Y., Marinaki, M., & Douligeris, C. (2008). Honey bees mating optimization algorithm for the project scheduling problem. *IEEE*

Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 38(5), 786-797. doi:10.1109/TSMCA.2008.2001660

Mavrovouniotis, M., Yang, S., Van, M., Li, C., & Polycarpou, M. (2020). Ant Colony Optimization Algorithms for Dynamic Optimization: A Case Study of the Dynamic Travelling Salesperson Problem [Research Frontier]. *IEEE Computational Intelligence Magazine*, *15*(1), 52–63. https://doi.org/10.1109/mci.2019.2954644

Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant colony optimization for resource- constrained project scheduling. *IEEE Transactions on Evolutionary Computation*, 6(4), 333-346. doi:10.1109/TEVC.2002.802460

Minna, L., & Yanxiang, L. (2015). Design and Implementation of Course Scheduling System Based on B/S. *Computer Knowledge and Technology*, *06*(05), 62–63.

Mushi, A. (2010). Tabu search heuristic for university course timetabling problem. *African Journal of Science and Technology*, *7*(1). https://doi.org/10.4314/ajst.v7i1.55191

Nayar, N., Gautam, S., Singh, P., & Mehta, G. (2021). Ant Colony Optimization: A review of literature and application in feature selection. In *Lecture notes in networks and systems* (pp. 285–297). https://doi.org/10.1007/978-981-33-4305-4_22

Paquete, L., & Stützle, T. (2003). An experimental investigation of iterated local search for coloring graphs. *The Journal of Heuristics*, 9(3), 251-292. doi:10.1023/A:1023621322657

Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2006). The Bees Algorithm – A novel tool for complex optimisation problems. *Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems*, 454-459. doi:10.1016/B978-008045157-2/50081-X

Pramanik, P. K. D., Pal, S., Mukhopadhyay, M., & Singh, S. (2021). Big Data classification: techniques and tools. In *Elsevier eBooks* (pp. 1–43). https://doi.org/10.1016/b978-0-12-820203-6.00002-3

Rajendran, C., & Ziegler, H. (2004). Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, 155(2), 426-438. doi:10.1016/S0377-2217(02)00966-2

Rita, L. (2013, August 5). *The Usage of Automation System in Smart Home to Provide a Sustainable Indoor Environment: A Content Analysis in Web 1.0*. Ssrn.com. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2306084

Ronacher, A. (2010). Flask. https://flask.palletsprojects.com/en/2.0.x/

Shobaki, G., Gordon, V. S., McHugh, P. R., Dubois, T., & Kerbow, A. (2022). Register-Pressure-Aware instruction scheduling using Ant colony optimization. *ACM Transactions on Architecture and Code Optimization*, *19*(2), 1–23. https://doi.org/10.1145/3505558

Socha, K., Knowles, J., & Sampels, M. (2002). A MAX-MIN ant system for the university course timetabling problem. *Proceedings of the 3rd International Workshop on Ant Algorithms*, 1-13. doi:10.1007/3-540-45724-0_1

Soofastaei, A. (2022). Introductory chapter: Ant Colony Optimization. In *IntechOpen eBooks*. https://doi.org/10.5772/intechopen.103801

Tao, W. (2015). Design and Implementation of Middle School Course Arrangement System Based on Genetic Algorithm. *Heilongjiang Education (High Education Research and Evaluation)*, *11*(09), 39–40.

Vancroonenburg, W., Petrovic, S., Kouroupetroglou, G., Kotsilieris, T., & Walsh, M. (2014). An ant colony system for production scheduling in the automotive industry. *Expert Systems with Applications*, 41(11), 4975-4988. doi:10.1016/j.eswa.2014.02.042

Von Thienen, W., Metzler, D., Choe, D., & Witte, V. (2014). Pheromone communication in ants: a detailed analysis of concentration-dependent decisions in three species. *Behavioral Ecology and Sociobiology*, *68*(10), 1611–1627. https://doi.org/10.1007/s00265-014-1770-3

Yue, Y., Han, J., Wang, S., & Liu, X. (2017). Integrated Train Timetabling and Rolling Stock Scheduling Model Based on Time-Dependent Demand for Urban Rail Transit. *Computer-Aided Civil and Infrastructure Engineering*, *32*(10), 856–873. https://doi.org/10.1111/mice.12300

Zahra Naji Azimi. (2005). Hybrid heuristics for Examination Timetabling problem. *Applied Mathematics and Computation*, *163*(2), 705–733. https://doi.org/10.1016/j.amc.2003.10.061

SOPHIA DALUMPINES

Jade Residences, Malagasang 1-G, Imus City,Cavite

09568952252

phiadalumpines@gmail.com

SKILLS

- Knowledgeable in frameworks such as React, Flask
- Knowledgeable in C, C++, Python, Javascript

PROJECT DONE

- Mushroom Classification Using Convolutional Neural Network
  - Developer
  - January 2023
- Signature Verification Using Siamese Neural Network
  - Developer
  - June 2023

CHRISTINE G. DELOS REYES

Blk. 5 Lot 6 Southville 8C, San Isidro, Rodriguez, Rizal

09951732651

christine.08delosreyes@gmail.com

SKILLS

- Knowledgeable in frameworks such as React, Django, Flask
- Knowledgeable in C, C++, Python, VB.net, Javascript

PROJECT DONE

- Mushroom Classification Using Convolutional Neural Network
  - Developer
  - January 2023
- Signature Verification Using Siamese Neural Network
  - Developer
  - June 2023

LEA O. FLOR

Blk. 36 Lot. 13 Leo St., San Roque, Navotas City

09218657046

leaotanes2014@gmail.com

SKILLS

- Knowledgeable in User interface and User Experience in Microsoft Power Apps and design logical workflow in Microsoft Power Automate
- Knowledgeable in software distribution, application development of cloud services, mailing lists and file sharing.
- Knowledgeable in Networks and Connectivity, copy equipment maintenance and troubleshooting.
- Knowledgeable in HTML, CSS, C, and C++

PROJECT DONE

- Competition Enforcement Office (CEO) Page and E-Complaint Portal Status Notification
  - Software Developer
  - April – May 2024
- PCC e-Log Management System (Digital Logbook)
  - Software Developer
  - June 2024

SELWYN AUDREY C. PEREZ

13 Bagong Lipunan St., BF International Village, Las Pinas City

09936537185

sydperez74@gmail.com

SKILLS

- Have an intermediate proficiency in programming languages such as C, C++ and Python
- Proficient in prototyping, wireframing, designing systems, and collaboration for interactive mockups using Figma
- Capable of creating basic edits and graphic design using Adobe Photoshop and Adobe Illustrator

PROJECT DONE

- Age and Gender Classification using OpenCV in Python
- Inventory Stock Control using Min-Max Analysis in Visual Basic
- Error Detection of APA 7 Citation using Regular Expression in Python
- Age Detection and Face Blurring for Privacy Protection of Minors using OpenCV in Python

RENZ L. SARMIENTO

8511 PUROK Laloma Brgy. 178 Area D. Camarin, Caloocan City

09069749200

sarmientorenz64@gmail.com

SKILLS

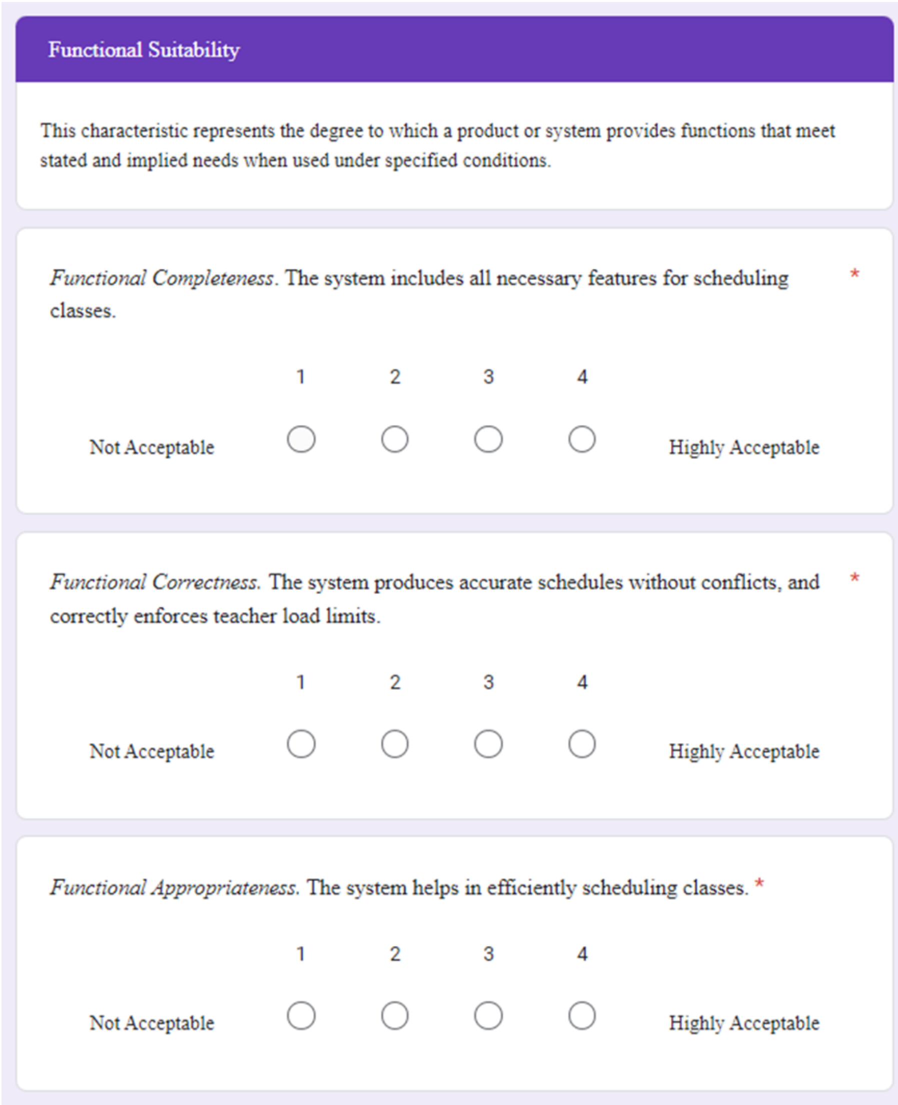
- Knowledgeable in HTML, CSS, and JAVASCRIPT
- Knowledgeable in Tailwind CSS, React JS and Next JS

PROJECT DONE

- Utilizing Student Information System to Assess Student Performance and Progress in Higher Education
  - Developer
  - June – July 2023

APPENDIX A

Survey Questionnaire Form



Functional Suitability

This characteristic represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.

Functional Completeness. The system includes all necessary features for scheduling classes. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Functional Correctness. The system produces accurate schedules without conflicts, and correctly enforces teacher load limits. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Functional Appropriateness. The system helps in efficiently scheduling classes. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Performance Efficiency

This characteristic represents the degree to which a product performs its functions within specified time and throughput parameters and is efficient in the use of resources under specified conditions.

Time Behavior: The system generates schedules within an acceptable time frame. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Resource Utilization. The system uses computational resources efficiently. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Capacity. The system can handle a large number of classes, teachers, and time slots without performance degradation. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Usability

Degree to which a product or a system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.

Appropriateness Recognizability. The system's purpose is clear and easy to understand. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Learnability. The system is easy to learn for new users. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Operability. The system is easy to use and the interface is user-friendly. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

User Error Protection. The system helps prevent errors during scheduling. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

User Interface Aesthetics. The system's interface is visually appealing. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Accessibility. The system is accessible to all intended users. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Maintainability

This characteristic represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements.

Modularity. The system's components are well-organized and modular. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Analyzability. The system makes it easy to diagnose and resolve issues. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Modifiability. The system is easy to modify to accommodate changes. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

Testability. The system's features can be tested easily. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-5301-3001 local 608| Fax No. +632-8521-4063 Email: cos@tup.edu.ph | Website: www.tup.edu.ph	Index No.	
		Revision No.	
		Effectivity Date	
VAA-COS	**THESIS GRAMMARIAN CERTIFICATION**	Page	

This is to certify that the thesis entitled,

schEDU: WEB-BASED PUBLIC SENIOR HIGH SCHOOL SCHEDULING SYSTEM USING ANT COLONY OPTIMIZATION

authored by

Sophia J. Dalumpines
Christine G. Delos Reyes
Lea O. Flor
Selwyn Audrey C. Perez
Renz L. Sarmiento

has undergone editing and proofreading by the undersigned.

This Certification is being issued upon the request of Sophia J. Dalumpines, Christine G. Delos Reyes, Lea O. Flor, Selwyn Audrey C. Perez, and Renz L. Sarmiento for whatever purposes it may serve them.

Franze Oroceo.

MS. FRANZE NAVARRO OROCEO
Grammarian

Technological University of the Philippines

June 13, 2024

Transaction ID	
Signature	

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-301-3001 local 711 Email: urds@tup.edu.ph | Website: www.tup.edu.ph	Index No.	F-URD-4.1-CSI
		Issue No.	01
		Revision No.	00
		Date	07102023
VRE-URD	CERTIFICATE OF SIMILARITY INDEX USING TURNITIN	Page	1 / 1
		QAC No.	CC-07102023

This is to certify that the manuscript entitled

"schEDU: WEB-BASED PUBLIC SENIOR HIGH SCHOOL SCHEDULING SYSTEM USING ANT COLONY OPTIMIZATION"

Authored by

SOPHIA J. DALUMPINES
CHRISTINE G. DELOS REYES
LEA O. FLOR
SELWYN AUDREY C. PEREZ
RENZ L. SARMIENTO

College of Science

has been subjected to similarity check on June 13, 2024
using Turnitin with generated similarity index of **7**%.

Processed by:

Assoc. Prof. FRANCIS A. ALFARO, Ed.D., LPT
Faculty, University Research
and Development Services

Certified correct by:

Assoc. Prof. FRANCISCO D. ESPONILLA II, Ed.D.
Director, University Research
and Development Services

Transaction ID	URD-TCERT-ELGB-10032022
Signature	

sophia dalumpines inbound7951924296280539622 - So... Switch to the new view Submission Details Help

WEB-BASED SCHEDULING SYSTEM USING ANT COLONY OPTIMIZATION

schEDU: WEB-BASED PUBLIC SENIOR HIGH SCHOOL SCHEDULING SYSTEM USING ANT COLONY OPTIMIZATION

A Thesis
Presented to the Faculty of the
Computer Studies Department
College of Science
Technological University of the Philippines
Ayala Boulevard, Manila

By
SOPHIA J. DALUMPINES
CHRISTINE G. DELOS REYES
LEA O. FLOR
SELWYN AUDREY C. PEREZ
RENZ L. SARMIENTO

In Partial Fulfilment of the
Requirements for the Degree
Bachelor of Science in Computer Science

June 2024

WEB-BASED SCHEDULING SYSTEM USING ANT COLONY OPTIMIZATION

ABSTRACT

Every academic year, academic institutions face the challenging and tim
consuming task of creating a schedule that accommodates the preferences and needs
their teachers and students. This challenge is no different for senior high schools. Th
paper aims to introduce schEDU, a web-based scheduling system specifically target
for public senior high school. This system will utilize Ant Colony Optimization (AC
to generate a seamless schedule for one semester: room availability, teachers' subje
preferences, and the required subjects for each section throughout the semester. Th
system employed fixed parameters: a 6-hour maximum teaching load per day, scho
days from Monday to Friday, time slots from 7:00 am to 3:00 pm, and 30-minute bre
times at 9:30 am and 12:30 pm. The developed web-app make used of Flask for backe
operations, React for frontend development, and MySQL as a database manager. Th
evaluation results revealed highly acceptable ratings for the system's function
suitability (3.55), performance efficiency (3.71), usability (3.78), and maintainabili
(3.75). Overall, the average rating was 3.77, indicating a highly acceptab
performance. The study's findings shows that the ACO algorithm was successful
implemented to construct efficient schedule solutions based on the required inpu
With this scheduling approach, the issues associated with manual scheduling a
reduced. The schEDU web application ensured that each generated schedule was fr
of conflicts and adhered to senior high school scheduling constraints. Using the AC
algorithm in the system's development proved efficient and effective, producing no
conflicting schedules for all sections and teachers. Future enhancements could focus
expanding the scope and number of user roles, accommodating multi-shift schedulin
and extending the scheduling process to generate schedules for an entire academic ye

Top sources | All Sources

0 Flags

7% Overall Similarity

1	www.coursehero.com	INTERNET	1%
2	tup on 2024-06-11	SUBMITTED WORKS	1%
3	tup on 2024-06-13	SUBMITTED WORKS	<1%
4	towardsdatascience.com	INTERNET	<1%
5	www.rappler.com	INTERNET	<1%
6	Silliman University on 2024-02-25	SUBMITTED WORKS	<1%
7	dokumen.tips	INTERNET	<1%
8	pureadmin.qub.ac.uk	INTERNET	<1%
9	CSU, Fullerton on 2021-05-17	SUBMITTED WORKS	<1%
10	Jose Rizal University on 2024-03-04	SUBMITTED WORKS	<1%
11	slsu on 2023-12-13	SUBMITTED WORKS	<1%
12	Charles University on 2024-05-09	SUBMITTED WORKS	<1%
13	dokumen.pub	INTERNET	<1%
14	www.globalscientificjournal.com	INTERNET	<1%
15	Jebari Hakim, Siham Rekiek, Ka...	CROSSREF	<1%
16	Study Group Australia on 2015-0...	SUBMITTED WORKS	<1%
17	eurchembull.com	INTERNET	<1%
18	www.congress.gov.ph	INTERNET	<1%
19	www.ijarp.org	INTERNET	<1%
20	123dok.com	INTERNET	<1%

7% Overall Similarity

Document Details

*3% AI