TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

COLLEGE OF SCIENCE

Developing a College of Science Scheduling System Using Genetic Algorithms

A research proposal presented to the Technological University of the Philippines,

College of Science

In Partial Fulfillment Of the Requirements for the Degree

Bachelor of Science in Computer Science

Submitted by:

Elmer II B. Absalon

Ivan Joshua C. Gepanayao

Ronne Eduard P. Reyes

Carl Matthew R. Tagapia

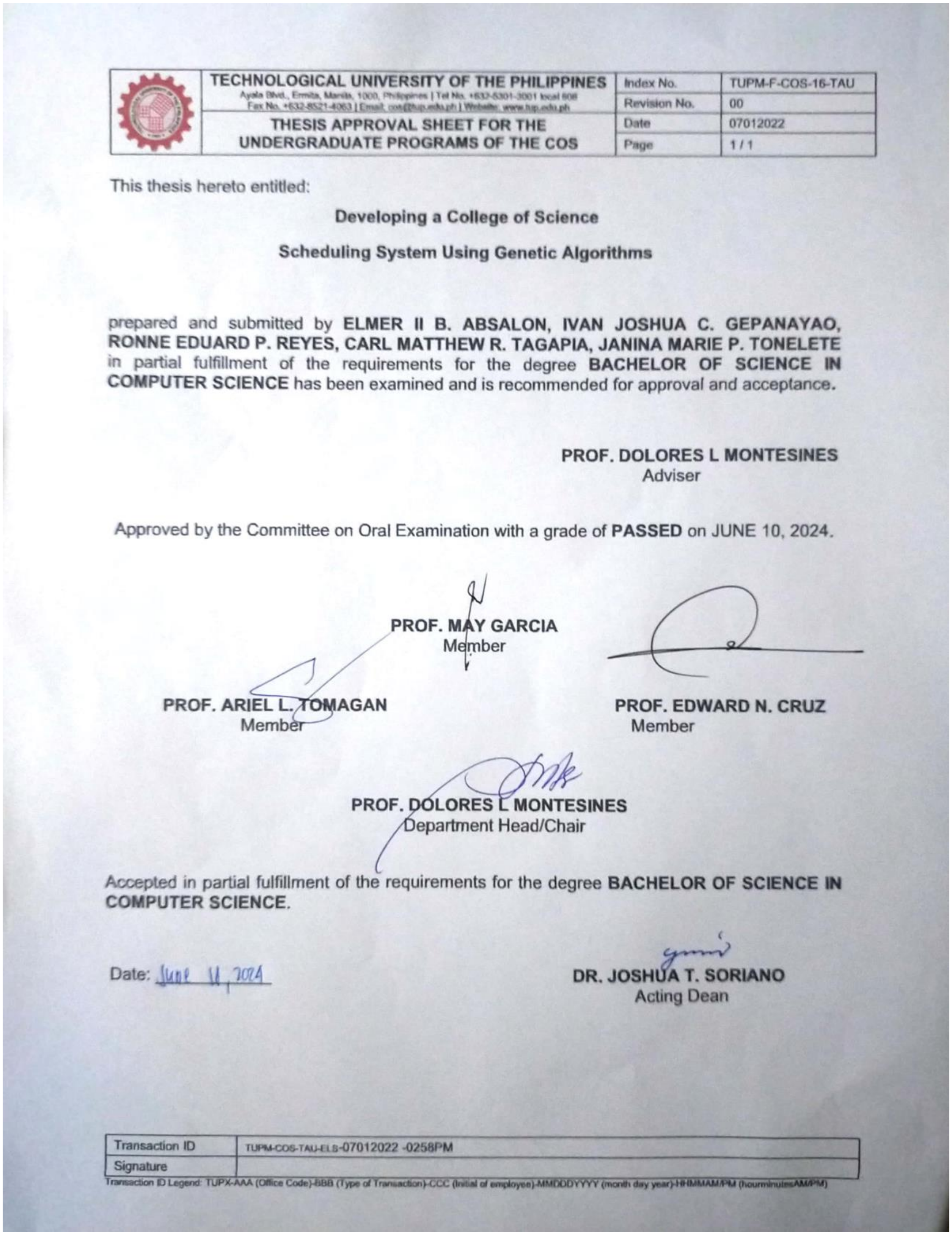
Janina Marie P. Tonelete

Submitted to:

Prof. Dolores L. Montesines

2024

TABLE OF CONTENTS



	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:

Developing a College of Science

Scheduling System Using Genetic Algorithms

prepared and submitted by **ELMER II B. ABSALON, IVAN JOSHUA C. GEPANAYAO, RONNE EDUARD P. REYES, CARL MATTHEW R. TAGAPIA, JANINA MARIE P. TONELETE** 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 10, 2024.

PROF. MAY GARCIA
Member

PROF. ARIEL L. TOMAGAN
Member

PROF. EDWARD N. CRUZ
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 11, 2024

DR. JOSHUA T. SORIANO
Acting 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)

Abstract

This study focuses on the College of Science at the Technological University of the Philippines in order to address the urgent need for effective scheduling systems in higher education institutions. Conventional manual scheduling approaches are commonly used labor-intensive, error-prone, and ineffective, which results in the misallocation of resources and conflicts. The objective of the study is to design a schedule that manages different restrictions and preferences through the use of genetic algorithms to optimize course scheduling. Natural selection processes serve as the inspiration for genetic algorithms, which provide a reliable solution for challenging issues like scheduling. To develop effective schedules, they can strike a balance between a number of restrictions, including student schedules, classroom sizes, faculty availability, and prerequisites for the courses. Genetic algorithms can be used in dynamic educational environments because of their flexibility and adaptability, which minimizes scheduling conflicts and ensures efficient resource use. Despite limitations such as restricted access to real-world data and practical implementation environments, the study demonstrates the viability and advantages of using genetic algorithms for course scheduling. The developed system successfully passed functionality, compatibility, and artificial intelligence tests, showing robustness and accuracy in generating optimized schedules.

Acknowledgements

First and foremost, we’d like to convey our heartfelt gratitude to God Almighty for His Blessings, wisdom, and strength throughout this trip. Without His favor, this achievement would not have been possible. This thesis would not have been possible without the help and encouragement of numerous persons.

We convey our sincerest gratitude to our advisor, Prof. Dolores L. Montesines, for their unwavering direction, insightful input, and persistent support during the course of this project. Their experience and effort have been essential.

A special appreciation to our colleagues and friends, who created a fascinating and exciting atmosphere in which to learn and grow. Their friendship and encouragement have been vital in overcoming the hurdles faced during this research.

Finally, we are deeply grateful to our families for their constant support, patience, and understanding. Their love and faith in us have been an ongoing source of inspiration.

Thank you all for making this journey possible.

Chapter 1

THE PROBLEM AND ITS BACKGROUND

This chapter provides the background, problem statement, and significance of the study. The general and specific goals are outlined, as well as the scope and limitations. This chapter will present an overview of the research, its social and personal impacts, and a brief description of the concepts discussed in the study.

Introduction

In all colleges and universities, scheduling systems has been evidently crucial. In fact, it influences faculty efficiency, student satisfaction, and the general performance of the institution. For instance, traditional methods of scheduling which are often done manually, can be inefficient, prone to mistakes, and time-consuming. This leads to wasted resources and unsatisfied users. In order to resolve this problem, recent advancements have introduced modern techniques like genetic algorithms. This study explores the development of a scheduling system for College of Science using genetic algorithms, aiming to improve the scheduling process by managing various constraints and preferences effectively.

Natural selection and evolution serve as the inspiration for genetic algorithms. Which provides a strong method for solving complex problems. By mimicking biological process like selection, crossover, and mutation, these algorithms can search through many possible solutions to find the best ones. For scheduling systems, genetic algorithms can lever the complex balance of various limitations, like course prerequisites, teacher availability, classroom sizes, and student schedules, to create practical and efficient timetables.

Utilizing genetic algorithms into scheduling process guarantees several advantages. It is important to acknowledge that these algorithms can easily adapt to changing conditions and restrictions resulting to a more flexible and resilient scheduling system. In addition, it can also provide an accessible solution that can handle increasing student numbers and course offerings without compromising the administrative workloads. By automating the scheduling process, genetic algorithms can considerably reduce the time and effort required, allowing the administrators to focus on other important tasks.

Background of the Study

In the Philippine setting, institutions on higher education face challenges in handling course scheduling due to varieties of factors such as varying student needs, faculty availability, and limited resources. In many colleges and universities in the Philippines, traditional manual scheduling is often used which are truly labor-intensive and susceptible to errors. This can lead to conflicting schedules, inefficient use of classrooms, and overall drop in the quality of education provided.

The rapid increase of student populations in the Philippines including *Technological University of the Philippines* further complicates the scheduling process. Colleges, particularly with huge departments need to accommodate numerous course sections, laboratory classes, and other specialized courses. The complexity of coordinating these elements increases the likelihood of scheduling conflicts and resource misallocation.

Furthermore, the educational system in the Philippines is undertaking significant changes with the implementation of the K-12 program and other educational reforms aimed at improving the quality of education. These reforms add another layer of complexity to the scheduling process, as institutions need to align their schedules with new curricula and educational standards.

Given these problems as discussed previously, we could say that there is an urgent need. For more dependable and effective scheduling systems. Developing a genetic algorithm-based scheduling system can be a good provider of viable solution by automating the scheduling process, reducing conflicts, and ensuring optimal use of resources. This method can help Philippine colleges and universities like (TUP) improve their scheduling efficiency, better serve their students and faculty, and enhance overall educational outcomes.

Objective of the Study

The main objective of the study is to develop and implement a scheduling system for a College of Science using genetic algorithms to optimize the course scheduling process. The specific goals of the study are as follows.

- To develop an algorithm that is efficient and can handle the complexities and constraints of course scheduling in College of Science.
- Ensure that the system can manage to adapt to various scheduling limitations such as course prerequisites, instructor availability, classroom capacities, and student schedules.
- Maximize the use of available classrooms, laboratories, and other resources to avoid overlapping.
- Guarantee that the scheduling system aligns with institutional policies and maximizes the efficiency of resource allocation.
- Lessen the occurrence of schedule conflicts for students and faculty, guaranteeing that courses do not overlap unnecessarily and that required courses are available at suitable times.
- Assess the impact of the scheduling system on resource utilization, time management, and overall efficiency within the College of Science building.

- Provide suggestions for implementing and improving the scheduling system depending on the research findings and user feedback.

By accomplishing these research objectives, this study aims to demonstrate the viability and advantages of using genetic algorithms for course scheduling in College of Science, ultimately contributing to the enhancement of academic administration and the overall educational experience.

Significance of the Study

Developing a College of Science scheduling system using genetic algorithms imparts a significant consequence for various stakeholders within the academic community of Technological University of the Philippines. By systematizing the scheduling process, it lessens the time and effort required by administrative staffs, which leads to a more efficient operations, allowing staff to focus on other important tasks, thereby enhancing overall administrative productivity.

Students can also benefit from a well-constructed schedule system that minimizes conflicts and guarantees access to necessary courses, which may lead to higher satisfaction, better academic performance, and improved retention rates as they can complete their programs in a timely manner without unnecessary delays.

Faculty members will also experience benefits with this program. This will visualize instance of overlapping classes and overbooking, allowing instructors to plan their time better, engage more fully in their teaching and research responsibilities, and maintain a balanced workload. Moreover, the system ensures optimal use of classrooms, laboratories, and other institutional resources, preventing the underutilization or overbooking of facilities, which leads to cost savings and more efficient management of physical resources.

Scope and Delimitations

Scope

The purpose of this work is to design a scheduling system, specifically for the College of Science and other universities that provide science-related courses. The College of Science scheduling system will consider coursework and available resources, such as labs and classrooms. The project's main optimization method for creating class schedules will be the genetic algorithm. It will examine how genetic algorithms can be modified and used to address the college's scheduling problem.

Limitations

Significant limitations of this research are the restricted access to real-world data and practical implementation environments for student researchers investigating genetic algorithms and scheduling systems. The performance of scheduling algorithms relies greatly on the detailed and dynamic nature of educational institution operations. However, due to factors such as institutional policies, data privacy concerns, and limited resource availability, obtaining comprehensive and authentic datasets posed a considerable challenge for this research.

Chapter 2

Conceptual Framework

This chapter provides a more profound discussion of the review of related literature, related studies, the conceptual model of the study, and the operational definition of terms relevant to this study. Chapter II expounds the concepts discussed in the paper by providing supporting materials that will enable the researchers to further understand the study.

Review of Related Literature

This section contains an in-depth examination of previously published scholarly works, encompassing academic articles, books, conference papers, theses, and relevant research materials.

Introduction

Ghaemi et. al (2020) Using genetic algorithms to tackle university course scheduling challenges, focusing on reducing conflicts. Two methods, cooperative GA and modified GA, are used. Modified fundamental genetic operators improve the performance of the modified GA approach, while intelligent operators enhance overall algorithm behavior. [1]

Sigl et al. (2018) A description is given of a genetic algorithm that solves a scheduling problem for timetables. We tested the technique on both small and big examples of the given problem. Modifying the fundamental genetic operators greatly improved algorithm performance. As an algorithm's overall behavior is improved, intelligent operators prevent the emergence of new conflicts in the individual [2].

Febrita et. al (2017) Because of both hard and soft constraints, establishing timetables in educational institutions is a challenging challenge. Students in junior and senior high schools have a full schedule to adhere to, which can cause fatigue and distraction. A modified genetic algorithm with a fuzzy temporal window is suggested as a solution to this. By reducing time and eliminating repair mechanisms, this algorithm ensures offspring with higher fitness values than the parent, getting to the optimal solution more quickly [3].

Sapru et. al (2020) An approach for organizing university timetables that avoids faculty conflicts and empty classrooms is presented in this study using genetic algorithms. The method creates workable global optima by employing a single point crossover, an application-specific encoding structure, and a rank-based selection strategy. Fit and convergence metrics are improved by the guided mutation operator [4].

Bhaduri (2019) The issue of scheduling educational timetables using genetic algorithms and memetic hybrid algorithms—also known as genetic artificial immune networks—is reviewed in this paper (GAIN). In comparison to Genetic Algorithms (GAs), GAIN finds the best possible answer more quickly. The study emphasizes the difficulties in resolving intricate scheduling issues, including those involving people, manufacturing, and educational timetable scheduling, as well as the possible applications of evolutionary methodologies [5].

P. Tormos et. al (2018) In particular, the Train Timetabling Problem is one of the difficult real-world issues that this study uses evolutionary algorithms to tackle. Train timetable optimization on a single line track, which is NP-hard owing to scheduling conflicts, is the goal of the Genetic Algorithm (GA). Real-world examples from the Spanish Manager of Railway Infrastructure (ADIF) are used to evaluate the GA, and the results show that it is a suitable approach for investigating complicated issues and arriving at sound answers as soon as possible [6].

Abduljabbar I. & Abdullah S. (2022) The timetabling issue in university departments is resolved by this study by utilizing genetic theories and an evolutionary algorithm. It creates a multi-solution schedule without redundancy by producing random and complete optimal timetables for every step. The primary contributions are raising the likelihood that the best schedule will be provided at each stage, allowing for replacement when necessary, and improving flexibility in creating optimal schedules with many copies. Courses, professors, and time are combined using the Genetic Algorithm (GA) to create a flexible scheduling system [7].

Burke E.K et. al (2016) In higher education institutions, creating exam schedules is a typical problem that is frequently handled manually or with little management. Students now have more course options thanks to the advent of modular degrees. Due to the increase in students, timetables are becoming more limited and new ones must be created to accommodate staff, student, and course changes, which results in a large amount of administrative labor [8].

Oh J. & Wu C. (2014) A novel approach to real-time job scheduling in multiprocessor systems is presented and evaluated in this work. Its goals are to reduce the total amount of task latency and the number of processors needed. Due to the problem's non-commensurable and competing objectives, a multi-objective genetic algorithm (GA) is used to carry out the minimization. The testing results demonstrated that the performance of our approach was equivalent or superior for 178 out of 180 randomly created task graphs when compared to five previously utilized methods, such as list-scheduling algorithms and a specific GA. The effect of a task graph's sparsity on our algorithm's performance is also displayed [9].

Abbaszadeh M. & Saeedvand S. (2016) The nondeterministic polynomial (NP) problem of university course scheduling is addressed in this study. Conventional techniques of solving this problem are laborious and time-consuming. The algorithm incorporates lecturer, class, and course information along with improved genetic techniques and a new chromosomal structure. When compared to other algorithms, the algorithm's efficiency in creating an optimum scheduling table for the weekly university program is high [10].

Pahlevan M. & Obermaisser R. (2018) An Ethernet extension called Time-Sensitive Networking (TSN) was created to handle temporal restrictions in mission-critical applications. Nevertheless, the interconnectedness of routing and scheduling issues is frequently ignored by current scheduling methods, which restricts their applicability to real-time systems. A novel heuristic scheduling method creates static global schedules in a single step by combining scheduling and routing restrictions. Compared to existing systems, this strategy enhances stimulability, TT transmission efficiency, and resource utilization [11].

Omara F. & Arafa (2017) The NP-complete problem in parallel and distributed computing systems is addressed in this study by presenting two evolutionary algorithms for task graph scheduling and processor mapping that are bound by precedence. Task duplication is used in the second technique to get around communication overhead, while the first algorithm employs fitness functions to reduce execution time and load balance satisfaction. The algorithms have been put into practice and assessed against benchmarks, demonstrating a consistent outperformance over conventional algorithms [12].

Deris et. al (2019) In order to plan timetables, this work provides a hybrid approach that combines constraint-based reasoning and genetic algorithms. The method seeks to overcome the shortcomings of evolutionary algorithms in practical issues by identifying a workable and nearly optimal solution. It is evaluated on actual data for university schedule planning [13].

Ozdamar L. (2020) In order to minimize project length or makespan, the research suggests a genetic algorithm (GA) approach for resource-constrained project scheduling. The GA method makes use of ordered scheduling rules and chosen activity modes in an indirect chromosomal encoding. For this difficult task, the hybrid GA (HGA) technique combines classic scheduling tools with knowledge. The HGA method generates near-optimal solutions in a fair amount of computing time, according to the results [14].

Faghihi V. et. al (2014) For project managers in the AEC sector, the construction project schedule is essential for monitoring time, money, and quality. But creating schedules frequently calls for expertise, planning skills, and an understanding of job packages. In the construction business, the Building Information Model (BIM) has become more popular. A software program has been created that use the Genetic Algorithm (GA) to automatically determine a construction sequence that is structurally stable. The methodology's potential as an expert system tool in building projects was demonstrated through 21 tests that validated it [15].

R. Barman et al. (2015). The rising demand for rail transport increases the need for efficient use of infrastructure capacity. Minor schedule deviations can disrupt service quality. With limited excess capacity, better planning tools are essential. This paper presents a heuristic model combining Fixed Path and Genetic Algorithm approaches to optimize train schedules. The Fixed Path model uses set routes, while the Genetic Algorithm selects the quickest paths. This combined method minimizes travel time and maximizes network capacity, enhancing timetable stability and rail traffic management within safety constraints [16].

Chan, W., et al., (2016). This paper presents a new approach to resource scheduling using genetic algorithms (GAs). The methodology does not rely on any set of heuristic rules. Instead, this strength is in selecting and recombining the GA in order to understand the domain of a particular project network. It can therefore evolve better schedules compared to its objective function. Moreover, unlike existing methods that are class-dependent, it is an all-inclusive model for both resources leveling as well as limited resource allocation problems. This study focuses on the design and mechanisms of the model. Meanwhile, standard test problems are used to serve as case studies for illustrating how the performance of GA-scheduler compares with other heuristic methods when resource availability varies. Exponential growth is not seen in time complexity of larger problem sizes with results from this model [17].

Lee, C. Y., et al., (2017). Analytical processes have demonstrated that dynamic job shop scheduling is an unsolvable issue. By cleverly narrowing the search space that is taken into consideration recent developments in computer technology—particularly in artificial intelligence—have mitigated this issue and made it possible to get better results. Scholars have employed diverse methodologies broadly classified as artificial intelligence to address scheduling issues in job shops. Expert systems genetic algorithms and machine learning have been the most widely used of these. Out of these we find that genetic algorithms and machine learning show promise for scheduling applications in a job shop. In this paper we propose to develop a job shop scheduling system by combining the complementary strengths of induced decision trees a machine learning technique and genetic algorithms. Empirical findings: a genetic algorithm is used to dispatch jobs at each location while machine learning is used to release jobs onto the shop floor [18].

K. Deb and P. Chakraborty, (2018). If the level of service to passengers is to be maximized within the constraints of the resources available bus transit system scheduling needs to be approached as an optimization problem. In this work we formulate a scheduling problem for a transit system whose goal is to minimize the total waiting time for both transferring and no transferring passengers while meeting various resource- and service-related constraints. It has been found that even with a basic transit system (one bus station with three routes) there are too many variables and constraints to handle with traditional mixed-integer optimization methods. The paper demonstrates that genetic algorithms (GAs) are perfect for these kinds of problems primarily because they can easily handle complex algorithmic approaches (involving if then-else conditions) because they (i) allow procedure-based declarations and (ii) naturally handle binary variables which takes care of transfer decision variables which make up the majority of the decision variables in the transit scheduling problem. The study also demonstrates how easily a number of other more practical extensions to the basic transit scheduling problem can be handled by the same GA procedure with only minor changes including buses with limited capacity buses that arrive late and a multiple-station transit system with shared routes between bus stations. The applicability of GAs in more complicated scheduling problems is suggested by the simulation results which demonstrate the effectiveness of GAs in all of these situations [19].

Sachi G. et. al (2010) The present investigation describes a genetic algorithm for optimizing multiprocessor scheduling based on natural evolutionary concepts. Natural selection, crossover, and mutation operators are used in the algorithm. It is NP-complete and was compared to the HEFT heuristic method. The simulation results encompass solution quality, genetic algorithm resilience, and the impact of mutation probability on performance. The algorithm's performance is evaluated in comparison to HEFT. [20].

Saptarini, N. G. et al., (2018). Scheduling courses by hand can be very difficult time-consuming and occasionally even go against strict and flexible rules. Typically, soft constraints are schedule-related preferences of the teacher and students. In order to minimize soft constraint violations and prevent hard constraint violations genetic algorithms (GA) were used in this study. In this study the GA divided the population into several groups. The distributed GA creates population groups and following each iteration migration between the groups is carried out according to the specified migration probability. The distributed GA was used to prevent the original GAs potential for premature convergence. In this investigation there was a probability of migration of 0 0. 1 0. 2 0. 3 0. 4 and 0. 5. According to the study distributed genetic algorithms are successful in preventing hard constraint violations minimizing violations of soft constraints and avoiding premature convergence [21].

S, R., & ER, N. (2013). The use of a pool of resources via the internet for distant users is known as cloud computing. that can be scalable readily available and resource-efficient. to achieve optimal utilization of. resources the assignments must be planned. Choosing the right resources to allocate is the scheduling problem. toward the completed tasks. The uncertainty surrounding task arrival at run time and allocation is known as dynamic scheduling. When multiple tasks come in at once resources become tedious. Avoid this scheduling issue by doing the following. The application of genetic algorithms is made. Natural processes are the subject of the genetic algorithm a heuristic technique. choosing one answer out of all the options. A genetic algorithm is used to schedule the tasks. based on the memory and computation. There is a dynamic schedule for the tasks. The way things were carried out. Parallel processing shortens time. There is cloud storage for the scheduled data. By means of GA to acquire. worldwide efficiency [22].

Delavar, A.G. et al., (2010). This paper introduces a new scheduling algorithm for independent tasks based on Genetic Algorithm which studies all the parameters in a grid environment. Compared to comparable earlier algorithms this one has the potential to be more dependable and efficient. The simulated outcomes and justifications for aiming for increased efficiency and make span in the grid environment. Retry migration and replication are less efficient than the check point method which is employed in grids with high fault rates and high fault rates. This approach keeps its efficacious efficiency in these circumstances. This leads to improvements in servicing quality in a variety of grid environments and a significant reduction in the average task recovery time. The primary goal of this paper is to enhance speed by decreasing the number of generations repeated in the Genetic Algorithm while also considering the communications costs (found in the fitness function) while preserving fitness efficiency. Gridsim is used for the simulations in order to demonstrate how the suggested algorithm has improved over earlier algorithms [23].

V. Di Martino (2003). Approaches to increase the systems overall throughput could be beneficial for the computing GRID infrastructure. Due to the generality of the GRID infrastructure, it is possible that resource requests for jobs will contain different ontologies. An opportunity to optimize a number of factors including network load task costs in relation to deadlines and service quality in general may present itself with such flexible resource requests. The outcome of the GRID jobs allocation simulation is what we present. Unlike prior work on up to 24 jobs the search strategy for this input case does not converge to the optimal case within the limited number of trials performed. The advantages of using genetic algorithms to raise the caliber of scheduling are explored. A distributed group of parallel machines scheduling can be studied through the use of the GGAS software environment which was used to create the simulation. We are able to add additional first level schedule policies for the FCFS under consideration to GGAS thanks to its modular structure. The papers findings recommend using a local search strategy to increase convergence when there are many jobs to consider as in real-world operations [24].

R. C. Correa, et al., (2019). The goal of the multiprocessor scheduling problem is to schedule a given program in a multiprocessor system so that the execution time of the program is as short as possible. Due to the difficulty of precisely solving this problem there are numerous heuristic techniques available for determining a suboptimal schedule. Our proposed approach is a novel combination of genetic algorithms wherein a list heuristic is used in the crossover and mutation genetic operations to improve the genetic algorithms performance by providing some knowledge about the scheduling problem. A pure list heuristic and a pure genetic algorithm from the literature are empirically compared with this knowledge-augmented genetic approach. Experiments conducted using synthetic instances of the scheduling problem demonstrate that despite its slower execution time our knowledge-augmented algorithm yields significantly better-quality solutions [25].

K. P. Dahal, et al., (2021). It is necessary to ascertain the unit commitment (UC) and economic dispatch (ED) for every generator in a power system at each time interval during the scheduling period in order to solve generation scheduling (GS) problems. Simultaneous consideration of these two decisions is necessary for the solution procedure. New approaches to solving GS have drawn a lot of attention from researchers. The aim of this paper is to find a faster way to obtain the optimum or near optimum solution for the GS problem by applying various genetic algorithm (GA) based approaches and exploring ways to improve these techniques. The outcomes demonstrate that the GA-based hybrid approach provides a viable substitute for resolving practical GS issues in a reasonable amount of time [26].

E. Gil, J. Bustos and H. Rudnick (2013). A novel model is suggested to address the hydrothermal systems short-term generation scheduling issue. The model addresses the subproblems of unit commitment economic load dispatch and short-term hydrothermal coordination simultaneously using genetic algorithms (GAs). Hourly generation schedules are obtained for both hydro and thermal units considering a scheduling horizon period of one week. To maximize the quantity of hydro energy used each week future cost curves for hydro generation derived from long- and mid-term models have been consulted. A set of expert operators has been added to the genetic algorithm (GA) implementation in order to enhance the algorithms behavior along with a new method for representing potential solutions. Presentation and discussion of real-world system results are provided [27].

U. Fissgus (2020). The scheduling of mixed task- and data-parallel modules that include communication and computation operations is taken into consideration. The maximum level of task- and data-parallelism that the method to be implemented can have is specified before the program is generated. The degree of parallelism is adjusted in multiple derivation steps to fit a particular distributed memory machine. We introduce a scheduling derivation step that utilizes the genetic algorithm paradigm. In addition to choosing the proper data distributions and task implementation versions scheduling also considers the execution order (independent modules can be executed concurrently by independent groups of processors or consecutively by all available processors). An example from numerical analysis is used to illustrate the algorithms efficiency [28].

Yao-Te Wang, et al., (2008). In order to create course schedules that best meet the needs of teachers and students an automatic course scheduling system is used in this study along with a data mining technique and a genetic algorithm. In order to meet a variety of requirements course scheduling in colleges and universities is an extremely difficult task. Traditional course scheduling primarily considers the needs of the school with little consideration given to the interests and career development requirements of the students. Courses and schedules are designed in accordance with the features of specific departments and institutes. Based on students' needs this study creates a workable automatic course scheduling system that splits the scheduling process into two phases. Students needs in choosing courses are considered in the first stage and an association between the courses they have chosen is mined using the association mining technique in the second stage the course schedule is arranged using a genetic algorithm. In particular this study considers teachers preferred schedules calculates the cost function value of each class period bases its course selection on student's willingness examines the effects of course arrangement in various class periods on students learning performance and finally uses the genetic algorithm for class period exchange to create the best possible course schedule. The experiment results demonstrate that the automatic course scheduling system suggested in this study can effectively replace the laborious process of traditional manual course scheduling. In addition, it can generate course schedules that genuinely satisfy users’ needs and raise teacher and student satisfaction offering a win-win solution for the school teachers and students [29].

L. Ozdamar (2019). In order to create course schedules that best meet the needs of teachers and students an automatic course scheduling system is used in this study along with a data mining technique and a genetic algorithm. In order to meet a variety of requirements course scheduling in colleges and universities is an extremely difficult task. Traditionally course scheduling is approached primarily from the perspective of the school resulting in planned courses and schedules. A genetic algorithm (GA) approach is suggested for the general resource-constrained project scheduling model which allows for the execution of activities in multiple operating modes and accounts for both renewable and nonrenewable resource constraints. The duration and resource requirements vary depending on the activity type both renewable and nonrenewable. Reducing the projects duration or make span is the goal. It is well known that the scheduling issue at hand is among the most challenging and that it is challenging to identify even a workable solution much less the ideal one. The GA approach presented in this paper uses an indirect chromosome encoding consisting of a ordered set of scheduling rules and selected activity operating modes to incorporate problem-specific scheduling knowledge. The chromosomes scheduling rules are employed in an iterative scheduling algorithm to create the schedule that the chromosome produces. The hybrid GA (HGA) approach refers to the suggested GA that combines traditional scheduling tools and knowledge especially created for the resource-constrained project scheduling problem. The outcomes show that in a reasonable amount of computation time the HGA approach yields near-optimal solutions [30].

M. Aggarwal, et al., (2015). The authors introduced a scheduler based on genetic algorithms within highly scalable distributed resource management architectures for grid computing. While achieving competing and sometimes incompatible goals a scheduler must make effective use of the resources at hand. Several jobs with quality-of-service restrictions may make up the grid workload. With respect to arbitrary precedence constraints and arbitrary processing time each job is represented by a directed acyclic graph (DAG). Our grid research group is developing other tools and the scheduler has been designed to work with them. In order to minimize make-span idle time of the available computational resources turnaround time and user-specified deadlines we present the design implementation and test results for such a scheduler. The scheduler can be used at any tier in the hierarchical architecture. Additionally, a research grid made up of sizable clusters connected by a high-bandwidth dedicated network or the intra-grid of a sizable organization can both use it [31].

O. Morandin, et al., (2018). The modeling task and the application of a technique to solve it are involved in the scheduling of the manufacturing systems production problem. Such a problem can be modeled in a few different ways and various search techniques have been applied to the model in an attempt to find an answer. Performance parameters such as makespan or another must be considered in the solution. However, response times can become crucial based on the systems size and complexity usually necessitating a rescheduling. The goal of research is to find a search strategy for the scheduling problem using genetic algorithms. The minimal makespan and response time are the performance criteria used in this work to propose using a genetic algorithm to solve the problem [32].

Wu Ying and Li Bin (2016). A common NP-complete problem job-shop scheduling is a crucial part of manufacturing control and planning for CIMS environments. Studies on the scheduling of job shops concentrate on knowledge-based strategies and heuristic searches which are helpful despite the challenge of acquiring knowledge. Genetic algorithms are optimization techniques that draw inspiration from the principles of natural evolution. Genetic algorithms are effective despite their simplicity. In this paper three novel genetic algorithms models are presented to design a job-shop scheduling algorithm: decimal idle time coding genetic algorithms (DITCGA) binary idle time coding genetic algorithms (BITCGA) and adaptive idle time coding genetic algorithms (AITCGA). By coding this problem during idle processing time, we effectively narrow down the possible solutions. To direct the searching or evolution process we employ an adaptive learning mechanism in our methods. These methods effectiveness is demonstrated by the simulation results [33].

A. Aminu et al. (2019). University scheduling is a very difficult task that takes a lot of time and work to complete. Stakeholders that create a schedule by hand confront a difficult situation because many conflicts can occur. This means that the creation of an effective automated method to eliminate the laborious and demanding tasks associated with examination schedule and invigilation scheduling is required. Examination office received reports of multiple altercations prior to the start of exams and Yusuf Maitama Sule University Kano is not an exception. The issue of the examination schedule and invigilation scheduling at Yusuf Maitama Sule University in Kano Nigeria was addressed in this paper. In order to provide an optimal solution, the paper used Genetic Algorithm (GA) which chooses a time slot venue and invigilators based on a specific teacher-student ratio of 1:40. The solution is implemented in Java using Java Swing-designed Graphical

User Interfaces (GUIs) and SQLite is used for database-related tasks (such as reading and writing data) [34].

M. R. Miryani and M. Naghibzadeh (2009). One of the most crucial issues with multiprocessing is the allocation of tasks optimally. In general prioritizing tasks on a multiprocessor is an NP-hard problem and the complexity of the problem increases with the presence of a precedence task graph. A lot of factors influence how best to divide up tasks. One aspect of multiprocessor systems is the cache reload time. These issues are also present in real-time systems. Real-time systems have a high time sensitivity so scheduling that takes time constraints into consideration is crucial. This paper uses a multiobjective genetic algorithm to propose a suboptimal scheduler for hard realtime heterogeneous multiprocessor systems that takes time constraints and cache reload time into account at the same time. Furthermore, it attempts to suggest a generalized approach based on genetic algorithms for real-time multiobjective scheduling in multiprocessor systems [35].

H. Chen, J. Ihlow and C. Lehmann (2019). The scheduling of job shops—a class of extremely complex combinatorial optimization problems—has seen the application of genetic algorithms. This isnt true for flexible job shops like flexible manufacturing systems where jobs have machine route flexibility. One common misconception regarding these job shop algorithms is that the routes that jobs visit the machines are fixed. In order to solve the flexible job-shop scheduling problem with makespan criterion a new genetic algorithm is presented in this paper. There are two components to the chromosome representation of the problem's solutions. The operations on each machine are outlined in the second section while the routing policy is defined in the first. The algorithm is introduced and reproduced through the use of genetic operators. Our algorithm can identify high-quality schedules as demonstrated by numerical experiments [36].

R. Ibrahim, et al., (2020). Formal approaches are frequently used to precisely express requirements specifications by utilizing mathematical notations. Using an examination scheduling system (ESS) case study as a guide we address the formal specification in this work. Exam schedules are created by the ESS system through the use of genetic algorithms. Exam scheduling is a problem that ESS was created to address in order to save time compared to traditional manual exam venue management. It is possible to generate exam schedules quickly and cut down on the time required for manual exam schedule processing by utilizing genetic algorithms. In order to demonstrate how the implementation of ESS maps to its design through the use of the formal specification the formal specification of ESS using the Z language is also presented. The Z schema is an efficient way to eliminate ambiguity found in natural language specifications. As a result, this will assist in lowering system development defects during the implementation phase [37].

J. Yu and R. Buyya (2006). Grid technologies have advanced toward a service-oriented paradigm in the past few years opening the door to a new approach to service provisioning that is based on utility computing models. Depending on their needs for QoS (quality of service) users utilize these services. The cost of workflow execution in these pay-per-use grids needs to be considered when scheduling users according to their QoS restrictions. In order to achieve results within a given budget we present a budget constraint-based scheduling method in this paper that minimizes execution time. We utilize a simulated grid testbed to evaluate the scheduling algorithm and develop a new kind of genetic algorithm to solve the scheduling optimization problem [38].

M. Agarwal and G. M. S. Srivastava (2016). As an emerging technology in the current context cloud computing allows an organization to use hardware software and applications over the internet at no upfront cost. The issue facing cloud service providers is how to use the underlying computing resources—such as virtual machines networks storage devices bandwidth etc. —as effectively and efficiently as possible. ought to be controlled to ensure that in a dynamic environment no computing device is in a state of under- or over-utilization. To prevent such a scenario dynamic task allocation always requires a strong task scheduling strategy. In this work we will introduce a task scheduling approach based on genetic algorithms that will efficiently distribute the load among virtual machines in order to minimize the overall response time (QoS). The CloudSim simulator is used to compare this Genetic Algorithm based task scheduling technique and the results indicate that it will perform better than the current methods such as the First-Come First-Served (FCFS) techniques based on greed [39].

C. -R. Chen, et al., (2013). The promotion of demand response is a crucial component in reducing the urgency of building power plants easing the pressure on the power company's supply and facilitating customer scheduling of electricity. The utility company's price announcements which assist customers in scheduling electricity use and the utilization of price negotiation techniques which yield benefits for both supply and demand are the primary methods of demand response. With the help of elastic load use time intervals which are divided into elastic and inelastic intervals based on the characteristics of electricity use the paper suggests a way to minimize tariffs for customers. The paper assumes that a customer's one-day electricity use will be used to simulate using a genetic algorithm comparing schedule and tariff variations under various scenarios of electricity use limitation. The results indicate that the minimum tariff objective can be achieved and the viability of the suggested method is confirmed by means of elastic load use time interval changes [40].

H. Murao and S. Kitamura (2018). The online approach to scheduling robot movements in a multi-robot system is presented in this paper. A multi-agent-based search method is presented by the authors to handle slight system modifications that occur while a schedule that is initially generated by a genetic algorithm (GA) is being executed. Applying the method to a welding plant they observe that multiple robots process multiple seams concurrently but there is no direct relationship between the seems and the robots. To minimize the total welding time a GA is used to schedule the welding order of each robots' seams and assign the welded seams to the robots. Managing robot issues and unexpected seam changes that require an online schedule modification is the responsibility of the suggested multi-agent-based search approach. Computer simulations demonstrate that the suggested method performs reasonably well in the event of system perturbations during the task [41].

Q. M. Hussein and A. N. Hasoon (2017). To maximize CPU utilization and achieve high throughput the operating system must schedule system and user processes. This work introduces a dynamic scheduling algorithm that uses genetic algorithm operations to schedule multiple non-preemptive tasks on a single processor system in order to minimize average waiting times and maximize average turnaround times. The algorithm relies on constructing a fitness function that accounts for process dead time in order to prevent starvation while still achieving maximum throughput. The outcomes of Matlabs simulation of the suggested algorithm (8. 3) demonstrate the effectiveness of this approach in achieving the best possible outcome by providing the lowest average turnaround time for carrying out the specified procedures [42].

A. Raghavan, et al., (2020). The authors introduced a day-ahead scheduling approach utilizing Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) for an Energy Storage System (ESS) in a microgrid. In a microgrid where pricing is dynamic the scheduling strategy seeks to reduce the amount that users pay. We give the search space for the optimization problem define an objective function for it and examine its structural characteristics. By using the maximum depths of charge and discharge in an hour (in percentage terms) for Bc and Bd respectively we demonstrate that the search space has a magnification of at least 50 × (Bc - Bd + 1). We obtain ESS schedules that offer average cost reductions of 11. 31% (using GA) and 14. 11% (using PSO) over the ESS schedule obtained using Net Power Based Algorithm in a simulation involving load energy generation and grid price forecasts for three microgrids of varying sizes [43].

X. Zuo, C. Chen, et al., (2015). In order to lower operating costs and raise the caliber of public transportation services vehicle scheduling for urban bus lines is a complex and challenging task. In order to minimize competing goals vehicles must be assigned to cover a set of trips as specified in a schedule. Previous methods group these goals together in a weighted way to create a single goal which is then solved using a single-objective optimization strategy. But they are only able to generate a single solution and it is difficult to determine the appropriate weight for each goal in order to come up with a better solution that can balance several goals. This paper presents a methodology to generate a set of Pareto solutions for this given problem. A collection of potential vehicle blocks is first produced. Subsequently an enhanced multi-objective genetic algorithm in conjunction with a departure-time adjustment process selects multiple block subsets from this candidate set in order to produce multiple Pareto solutions. We suggest a coding scheme that has a low decoding complexity and a comparatively short coding length in order to encode a solution. This method is used on a real-world vehicle scheduling issue involving a bus route in Nanjing China. Tests indicate that this method can generate good Pareto solutions faster than the experience-based solution that is actually applied [44].

S. G. Ahmad, E. et al., (2012). Due to the fact that effective task scheduling can significantly improve system performance overall it is crucial in heterogeneous systems. Through the efficient application of an evolution-based algorithm the task scheduling problem is addressed in this paper. Although the time complexity of genetic algorithms is higher, they show promise for producing results that are almost optimal even in large problem spaces. In addition to offering a schedule that is almost optimal the Performance Effective Genetic Algorithm (PEGA) which is the suggested algorithm also has minimal time complexity. Through rigorous search the PEGA effectively utilizes genetic operators (crossover and mutation) to find the best solution from the search space. This makes PEGA a performance-effective algorithm. Furthermore, efficiency is added to simplicity through chromosome encoding at the b-level. Using standard genetic algorithms (SGA) extensive simulations are used to compare the performance. When it comes to producing nearly optimal schedules with significantly less run time the comparison of results demonstrated that PEGA performs better than SGA [45].

T. Wang, et al., (2014). One of the most important concerns with cloud platforms is task scheduling. There are tons of users and an enormous amount of data. Reuse and asset sharing requests are becoming more and more important. In order to optimize the overall performance of the cloud computing environment an effective task scheduling mechanism should satisfy user needs and boost resource usage. A new scheduling algorithm based on double-fitness adaptive algorithm-job spanning time and load balancing genetic algorithm (JLGA) is established in order to solve this problem taking into consideration the new characteristics of cloud computing and the original adaptive genetic algorithm (AGA). This method not only achieves inter-node load balancing but it also determines a task scheduling sequence with shorter jobs and average job makespan. The population is initialized using a greedy algorithm in this paper which also weights the multi-fitness function and introduces variance to describe nodes that are under a lot of loads. Then using simulations, we contrast JLGA and AGAs performance. The efficacy of the optimization method and the validity of the scheduling algorithm are demonstrated [46].

Zhongni Zheng, et al., (2011). In cloud computing environments like Infrastructure as a Service clouds resource scheduling is an important procedure. We suggest an optimized scheduling algorithm to realize the optimization or sub-optimization for cloud scheduling issues in order to utilize the resources as efficiently as possible. On the basis of allowing the maximum utilization of resources we explore the possibility of arranging the Virtual Machines in a flexible manner to speed up the process of determining the optimal allocation. We view the scheduling issue as mathematically reducing to an imbalance assignment problem. Our scheduling strategy was made possible by the Parallel Genetic Algorithm which operates far more quickly than the conventional Genetic Algorithm. Our approach increased the rate at which resources were allocated and the way in which system resources were used as demonstrated by the experiments [47].

M. T. Jensen (2003). Since real-world applications operate in dynamic environments finding strong or adaptable solutions to scheduling problems is crucial. Rescheduling an existing plan is frequently necessary in such environments because of setbacks (e. g. g. malfunctioning equipment staff illness delayed delivery etc. (). Therefore, a solution that is strong or adaptable might be more useful than one that is ideal but difficult to change. The topic of reliable and adaptable solutions for scheduling issues in job shops is examined in this paper. A robustness measure is defined and its characteristics examined. It is demonstrated through experiments that a genetic algorithm can be used to find robust adaptable schedules with a short makespan. It is shown that these schedules outperform regular schedules in terms of rescheduling following a breakdown. When the performance of another robust scheduling method taken from the literature is compared with the rescheduling performance of the schedules generated by minimizing the robustness measure it is found that the former performs better in many cases [48].

M. Pahlevan and R. Obermaisser (2018). In order to handle the stringent temporal constraints of contemporary mission-critical applications Time-Sensitive Networking (TSN) is presented as a set of Ethernet extensions. Global Time-Triggered (TT) transmission schedules are used by TSN to provide determinism. The majority of scheduling solutions in use today ignore the relationship between scheduling and routing issues and solely use scheduling constraints to determine the design space for system implementations. This approach restricts the ability of previous methods to calculate a global schedule of TT communication for multiple real-time systems. Here we describe a genetic algorithm-based heuristic scheduling method. Our method creates static global schedules in a single step using joint constraints by combining the routing and scheduling constraints. In contrast to methods that solely employ fixed routing the number of scheduling options within the design space that arises from the combination of scheduling constraints and joint routing increases. Our solution thus enhances the schedulability. We also consider the interdependencies of TT flows multicast patterns and the distribution of real-time applications when scheduling using our genetic-based method. The experimental results demonstrate a significant improvement in schedulability TT transmission efficiency and resource utilization over the state-of-the-art solutions because of the optimized task binding and resource allocation [49].

Soon-mi Han, et al., (2008). Todaypsilas high-resolution imaging satellites are capable of complex missions with agility. Thats why since the 1980s there has been a greater interest in and study of the satellite mission scheduling optimization problem. The results of utilizing a genetic algorithm for mission scheduling are presented in this paper. This paper shows that scheduling satellite missions using a genetic algorithm is feasible. When solving mission scheduling problems computational time and solution quality are typically considered. In this work two methods of selection operator are used: tournament and roulette wheel. The case study also looks into how memory constraints and the quantity of iterations affect performance. As a result, it is anticipated that the optimization algorithm-based scheduler that uses a genetic algorithm in this work will lessen the workload associated with mission scheduling and increase the effectiveness of creating a mission timeline free of conflicts [50]

Chapter 3

Methodology

This chapter discusses the research methodology of the study with the following sections: project design, System design, Project development and testing procedures.

Project Design

The research developed an automated scheduling system intended to assist department heads and regular faculty members in selecting their preferred subjects and setting scheduling preferences. The system's flow and design process are elaborated in the system design, encompassing the system context diagram, use case diagram, and entity-relationship diagram.

System Design

The interactions within the GA-based Scheduling System and its external entities. At the center of the diagram is the core system itself, the GA-based Scheduling System, which orchestrates all scheduling activities. Surrounding this central system are three external entities: Faculty Members, Department Heads, and the Central Database. Faculty Members, who are the instructors, input their scheduling preferences into the system and subsequently review the generated schedules. Department Heads, responsible for overseeing academic departments, interact with the system to generate schedules for faculty members, review and adjust these schedules, and determine the number of sections for courses. The Central Database serves as a repository for storing scheduling data, including faculty preferences, generated schedules, and other relevant information. Data flows between these entities: from Faculty Members to the system for preference input and schedule review, from Department Heads to the system for schedule generation and adjustment, and between the system and the Central Database for data storage and retrieval. This diagram offers a clear visualization of how the scheduling system interacts with its users and data storage component, aiding in understanding its operational dynamics.

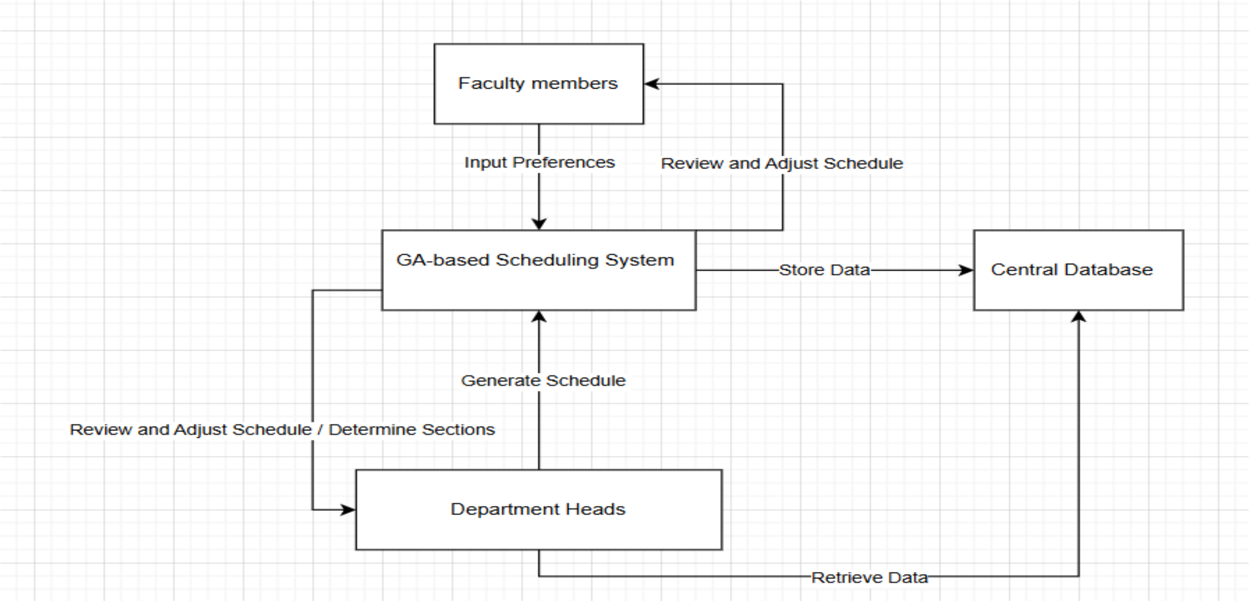


Fig 1. System Context Diagram

1. **Faculty Members:**
    - Input scheduling preferences.
    - Review generated schedules.
2. **Department Heads:**
    - Generate schedules for faculty members.
    - Review and adjust schedules.
    - Determine the number of sections for courses.
3. **Central Database:**
    - Store scheduling data.
        - Faculty preferences.
        - Generated schedules.

        - Other relevant information.

Data Flows

- **From Faculty Members to System:**
    - Scheduling preference input.
    - Schedule review.
- **From Department Heads to System:**
    - Schedule generation.
    - Schedule adjustment.
- **Between System and Central Database:**
    - Data storage.
    - Data retrieval.

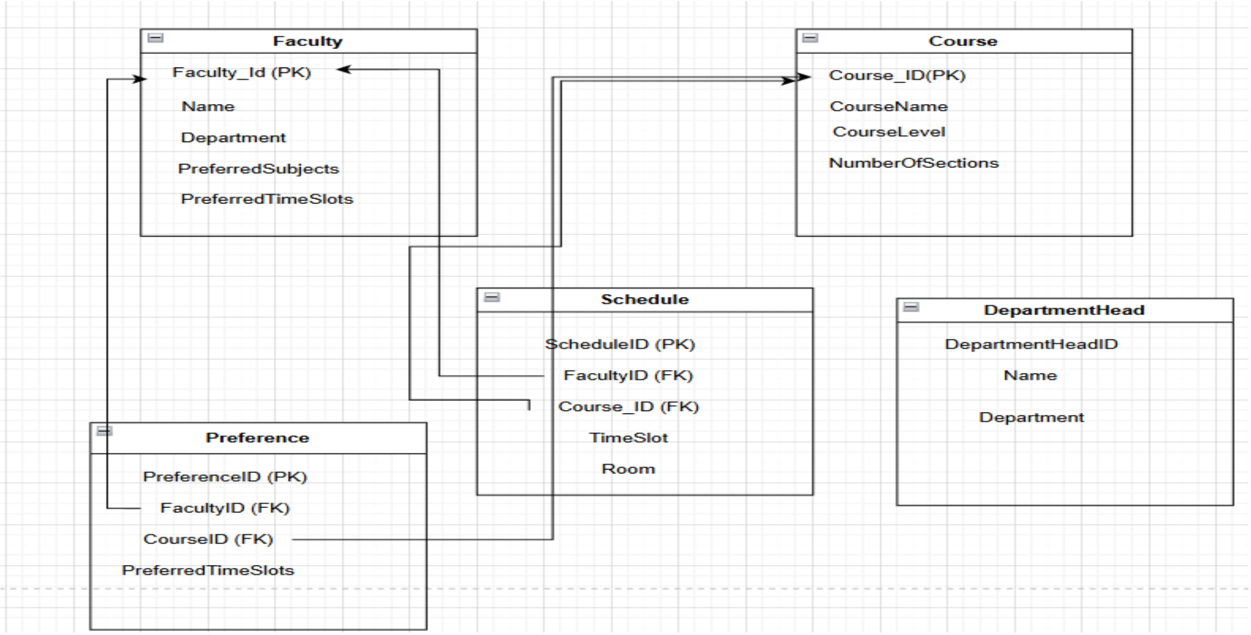


Fig.2 Entity Relationship Diagram of the Scheduling System

The Entity-Relationship Diagram (ERD) for the automated scheduling system includes several key entities: Faculty, Course, Schedule, Department Head, and Preference. Each entity is represented as a table in the database, with a primary key (PK) that uniquely identifies each record. The faculty table contains information about faculty members, including their unique identifier (FacultyID), name, department, preferred subjects, and preferred time slots. The Course table includes details about each course, such as its unique identifier (CourseID), name, level, and number of sections. The Schedule table records scheduling details, with foreign keys (FacultyID and CourseID) linking it to the Faculty and Course tables, ensuring each schedule entry corresponds to a valid faculty member and course. The Department Head table holds information about department heads, while the Preference table captures faculty members' scheduling preferences, also using foreign keys to link back to the Faculty and Course tables. Foreign keys are crucial in maintaining data integrity and referential integrity, as they ensure that relationships between tables are consistent and valid, preventing the creation of orphaned records and ensuring that all referenced data exists within the database. This structure helps the system efficiently manage and optimize schedules according to faculty preferences and department requirements.

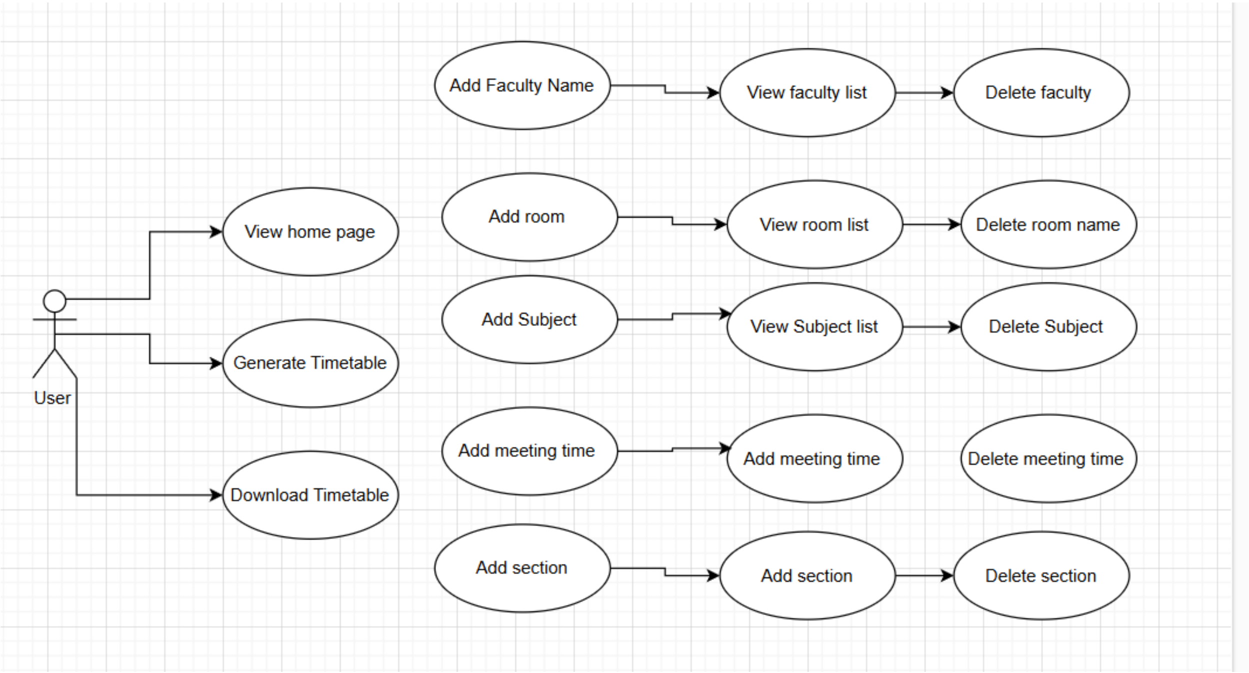


Fig.3 Entity Use Case Diagram

The use case diagram illustrates the key functionalities and interactions within a timetable management system. At its core is the "User" actor, representing anyone interacting with the system, including administrators, faculty, and students. The diagram outlines several primary use cases, starting with the "View Home Page," which grants access to the system's interface. "Generate Timetable" allows users to create schedules for classes, faculty, and rooms, while "Download Timetable as PDF" enables users to obtain a downloadable version of the timetable. Additionally, the diagram groups together use cases related to managing system entities, including adding, viewing, and deleting components such as faculty names, rooms, meeting times, subjects, departments, and sections. Each use case is directly associated with the user, indicating the actions users can perform within the system. These associations are depicted by straight lines with arrows. The diagram also includes notes clarifying the purpose of each use case, emphasizing the CRUD (Create, Read, Update, Delete) operations for managing system entities.

Actors

User: Any individual who interacts with the web application (e.g., administrators, faculty, students).

Use Cases:

1.) View Home Page: Access the home page.

2.) Generate Timetable: Generate and view the timetable.

3.) Download Timetable as PDF: Download the timetable in PDF format.

4.) Add Faculty Name: Add a new faculty name.

5.) View Faculty List: View the list of all faculty names.

6.) Delete Faculty Name: Delete a specific faculty name.

7.) Add Room: Add a new room.

8.) View Room List: View the list of all rooms.

9.) Delete Room: Delete a specific room.

10.) Add Meeting Time: Add a new meeting time.

11.) View Meeting Times List: View the list of all meeting times.

12.) Delete Meeting Time: Delete a specific meeting time.

13.) Add Subject: Add a new subject.

14.) View Subject List: View the list of all subjects.

15.) Delete Subject: Delete a specific subject.

16.) Add Department: Add a new department.

17.) View Department List: View the list of all departments.

18.) Delete Department: Delete a specific department.

19.) Add Section: Add a new section.

20.) View Section List: View the list of all sections.

21.) Delete Section: Delete a specific section.

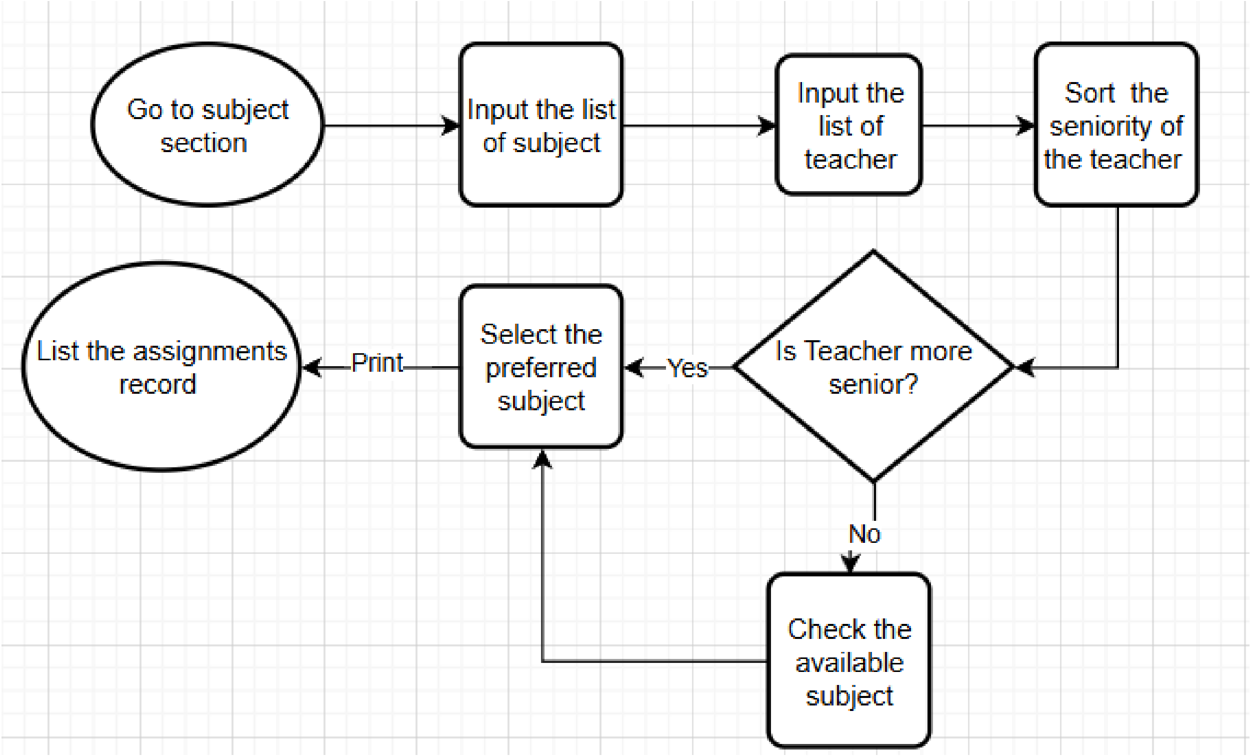

Go to subject section
Input the list of subject
Input the list of teacher
Sort the seniority of the teacher
Is Teacher more senior?
Yes
Select the preferred subject
Print
List the assignments record
No
Check the available subject

Fig.4 Assigning subject through seniority

The process of assigning subjects to teachers based on seniority begins with the initial step of collecting a comprehensive list of teachers. Once this list is compiled, the seniority of each teacher is determined, usually based on their years of experience or tenure within the institution. Following this, the teachers are sorted in descending order of their seniority. Concurrently, a list of all subjects that need to be assigned is collected.

With both lists in hand, the assignment process is initialized. The process enters a decision point, where it checks if there are more teachers left to assign subjects to. If there are teachers remaining, the next most senior teacher is selected. The system then checks the available subjects that has not yet been assigned. Another decision point occurs here: if there are subjects available, the most preferred subject is assigned to the selected teacher. The assigned subject is then removed from the list of available subjects, and the assignment record is updated to reflect this new allocation. The process then loops back to check if there are more teachers to assign subjects to.

If at any point there are no more teachers or no available subjects left, the process concludes, ensuring all subjects are assigned appropriately based on the teachers' seniority. This systematic approach ensures that the most experienced teachers are given priority in subject assignments, promoting a fair and structured allocation process.

Operation and testing procedure

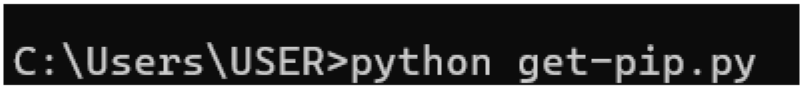
The following procedure are the steps to operate the Scheduling system using Genetic algorithm:

1.) Open the Visual Studio code application

2.) Open the source folder name

3.) Download Python through web browser the version must be at least 3.8 or higher

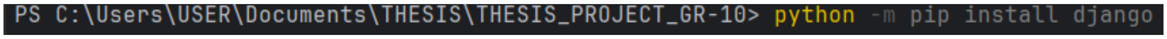
4.) Install “python get-pip.py” using CMD



```
C:\Users\USER>python get-pip.py
```

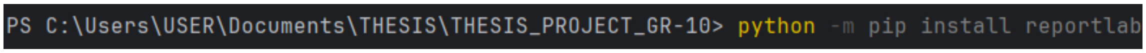
5.) Import the libraries

- python -m pip install django



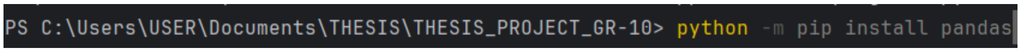
```
PS C:\Users\USER\Documents\THESIS\THESIS_PROJECT_GR-10> python -m pip install django
```

- python -m pip install reportlab



```
PS C:\Users\USER\Documents\THESIS\THESIS_PROJECT_GR-10> python -m pip install reportlab
```

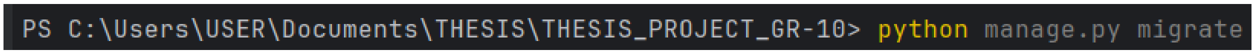
- python -m pip install pandas



```
PS C:\Users\USER\Documents\THESIS\THESIS_PROJECT_GR-10> python -m pip install pandas
```

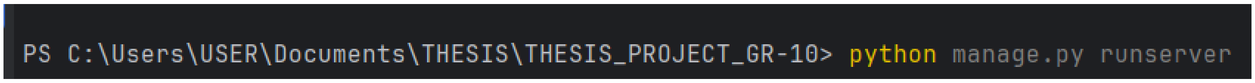
6.) Open terminal

7.) Type “python manage.py migrate”



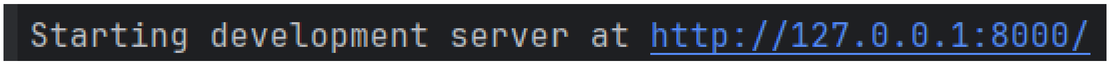
```
PS C:\Users\USER\Documents\THESIS\THESIS_PROJECT_GR-10> python manage.py migrate
```

8.) Type “python manage.py runserver”



```
PS C:\Users\USER\Documents\THESIS\THESIS_PROJECT_GR-10> python manage.py runserver
```

9.) click local server “http://127.0.0.1:8000/” once it appeared



```
Starting development server at http://127.0.0.1:8000/
```

Program Coding

The scheduling system were created using django, reportlab, random, and pandas library. Genetic Algorithm were used for optimizing schedules. The following steps were done to create the scheduled time table:

1.) Getting Course curriculum and teacher’s preferred subject with correspond to their units and shift

2.) Manually input the gathered data in the system UI

3.) Use the CRUD (Create, Read, Update, Delete) to each button in navigation bar

4.) Click the Generate time table

5.) Wait for the Genetic Algorithm to check the fitness and value

6.) Validate if it has duplication

7.) Export time table pdf

Evaluation Procedure

The evaluation tool that was used to assess the system’s acceptability was adapted from ISO 25010, “System and software engineering – System and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.”

The following procedures were conducted to evaluate the acceptability of the developed system implemented in local server:

1. Invited 15 purposively selected respondents who are professional system developers and 15 non-professionals.
2. A Google Form link with an embedded video demonstrating how the Scheduling system works, examples of its explanation, the local server, and how it is used in different programming languages, was sent to the evaluators
3. The evaluators were requested to evaluate the system based on the given evaluation sheets using a 4-point Likert Scale shown in Table 2.
4. The accomplished evaluation sheets were processed, and the data gathered were tabulated to determine the weighted mean ratings.
5. The verbal interpretation for the weighted mean ratings were interpreted using the range of scale value shown in Table 2.

Table 2.

Range of weighted mean values and its Qualitative Interpretation

Scale	**Range**	**Qualitative Interpretation**
4	3.26 – 4.0	Highly Acceptable
3	2.51 – 3.25	Very Acceptable
2	1.76 – 2.5	Acceptable
1	1.00 – 1.7	Not Acceptable

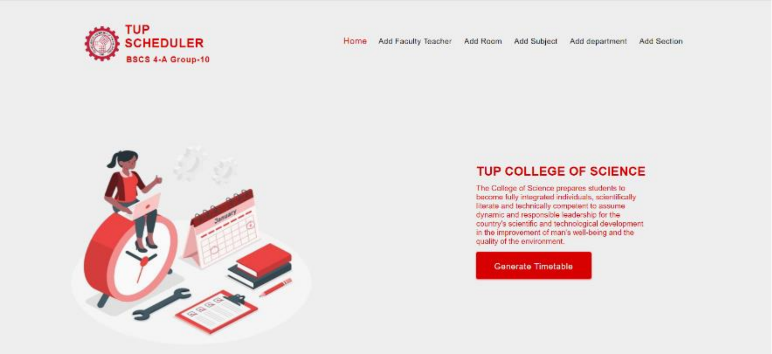
Chapter 4

RESULTS AND DISCUSSIONS

This chapter contains the project description, project structure, project capabilities, and limitations, and results of evaluation for the project.

Project Description

The study developed a Scheduling system using Genetic Algorithm which can help the COS department and faculty members to generate and optimize scheduling fast and efficient which enable them to lessen the hassle from manually creating the schedule from scratch. The system created in different programming language such as Python, CSS, and HTML. The system was only implemented using local host server



TUP COLLEGE OF SCIENCE

The College of Science prepares students to become fully integrated individuals, scientifically literate and technically competent to assume dynamic and responsible leadership for the country's scientific and technological development in the improvement of man's well-being and the quality of the environment.

Figure 5. Scheduling system landing page and documentation

The official documentation for the Scheduling system, Figure 4, includes code samples in various programming languages, guides for installing django, reportlab, pandas, and random. Instructions for utilizing django tools such as framework, random to shuffle and randomize the gathered data, pandas to randomize the preferred time of the faculty, and reportlab to export pdf. Scheduling system Documentation is accessible through this local server: http://127.0.0.1:8000/.

Project Structure

The system was deployed, and then various modules were published to different package management tools and was implemented in local server.

Table 3.

Endpoints table

Endpoint	**Method**	**Description**
/home/	POST	Renders the homepage.
/timetable/	POST	Renders the timetable page.
/timetable/pdf/	POST	Generates and serves a PDF version of the timetable.
/addfacultyname/	POST	Handles the addition of new faculty names.
/facultylist/	POST	Renders a page listing all faculty names.
/addroom/	POST	Handles the addition of new rooms.
/roomlist/	POST	Renders a page listing all rooms.
/subjectlist/	POST	Renders a page listing all subjects.

/addsubject/	POST	Handles the addition of new subjects.
/adddepartment/	POST	Handles the addition of new departments.
/departmentlist/	POST	Renders a page listing all departments.
/addsection/	POST	Handles the addition of new sections.
/sectionlist/	POST	Renders a page listing all sections.

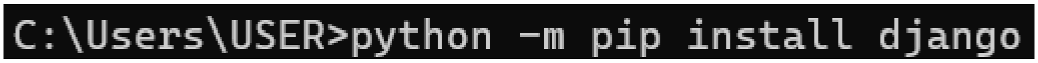
a. Django

Table 3 showed the endpoint for the Django application how it was defined through various view functions. Each view function corresponds to a specific URL pattern and handles the logic for generating responses to requests.

b. Modules and Packages

To facilitate seamless integration of the Scheduling system across numerous programming languages, including Python, CSS, and HTML, a variety of code abstractions have been developed and made accessible through various package management tools, including django, random, reportlab, and pandas.

i. Python module



```
C:\Users\USER>python -m pip install django
```

Figure 6. *Install django through CMD.*

Using PIP, the django, random, reportlab, and pandas packages are simple to install. Figure 6 shows the django module in the Python package which can be accessible through CMD. Users who install these packages can have access to the Scheduling system's UI and all functionalities, which can be used in Python-based projects. The unique methods are provided by the package, each of which corresponds to a different endpoint of the system. The example of django package methods that developer can use:

- **render:** This function is used to render HTML templates with data from views. It takes a request object, template name, and optional context dictionary as arguments.

- **redirect:** This function is used to redirect the user to a different URL. It takes a URL as an argument.

- **HttpResponse:** This class is used to create HTTP responses. It takes content and content type as arguments.

- **request.POST:** This is a dictionary-like object that allows access to submitted data by key name. It contains data sent via POST method.

- **request.method:** This attribute of the request object contains the HTTP method used in the request (e.g., GET, POST).

- **cache:** This is Django's built-in caching framework. It's used here to cache generated schedules to improve performance.

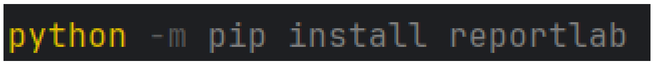
- **cache.set:** This method is used to set a value in the cache with a specified key and optional timeout.

- **cache.get:** This method is used to retrieve a value from the cache using a specified key.

- **Model.objects.all():** This method is used to retrieve all objects of a particular model from the database.

- **Model.save():** This method is used to save changes made to an object back to the database.

- **Model.delete():** This method is used to delete an object from the database.



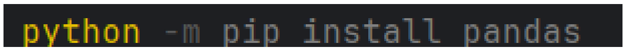
```
python -m pip install reportlab
```

Figure 7. *Install django through CMD*

ReportLab is a powerful Python library for generating PDF documents. Figure 7 shows the reportlab module in the Python package which can be accessible through VS Code. The examples of reportlab methods that can be used.

Adding Text: ReportLab allows you to add text to the PDF document with options for font selection, size, color, alignment, and rotation.

- **Rich Text Formatting:** You can include rich text elements like bold, italic, underline, and subscript/superscript in the text.
- **Adding Images:** ReportLab enables you to insert images (e.g., PNG, JPEG) into the PDF document and control their position, size, and scaling.
- **Image Processing:** You can manipulate images by rotating, scaling, cropping, or applying filters before adding them to the PDF.
- **Creating Tables:** ReportLab provides functionality to create tables with customizable rows, columns, cell styles, and content.
- **Formatting Tables:** You can format table cells with different colors, borders, padding, and alignment.



```
python -m pip install pandas
```

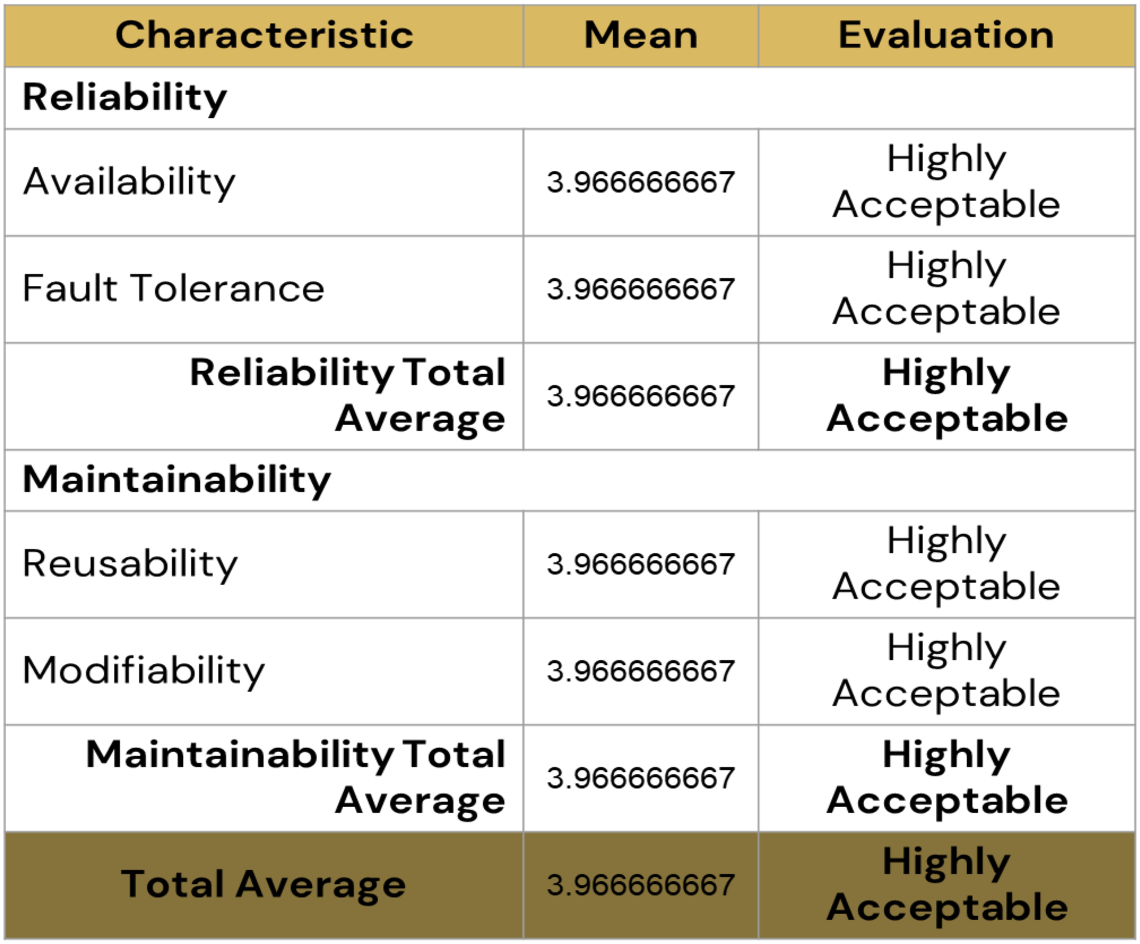
Figure 8. *Install pandas through VS Code.*

Pandas is a powerful data manipulation and analysis library in Python. Figure 8 shows the pandas module in the Python package which can be accessible through VS Code. The examples of pandas methods that can be used:

- **DataFrame:** Create a DataFrame from various data structures like lists, dictionaries, or NumPy arrays.
- **read_csv, read_excel, read_sql:** Read data from CSV files, Excel files, or SQL databases into a DataFrame.
- **loc:** Select rows and columns by labels.
- **groupby:** Group DataFrame using a mapper or by a Series of columns.
- **agg, aggregate:** Apply aggregation functions (e.g., sum, mean, count) to grouped data.

Evaluation result

Table 4



Characteristic	Mean	Evaluation
Reliability		
Availability	3.966666667	Highly Acceptable
Fault Tolerance	3.966666667	Highly Acceptable
Reliability Total Average	3.966666667	**Highly Acceptable**
Maintainability		
Reusability	3.966666667	Highly Acceptable
Modifiability	3.966666667	Highly Acceptable
Maintainability Total Average	3.966666667	**Highly Acceptable**
Total Average	3.966666667	**Highly Acceptable**

The project's Developing a College of Science Scheduling System Using Genetic Algorithms, has shown positive results.

Its rating for a criterion was recorded in all evaluation of our respondents, with a total rating of 3.97, which evaluates to “Highly Acceptable”. This showcases the project’s exceptional performance in terms of satisfying the criterion pertaining to “Reliability and maintainability, it has delivered its intended purpose in which users can expect accurate and reliable service of its functions.

Chapter 5

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATION

This chapter present the summary of findings, the conclusions drawn from the findings, and the corresponding recommendation for further enhancement of the project.

Summary of Findings

Based on the testS and evaluation results, the developed Scheduling system entitled "Developing a College of Science Scheduling System Using Genetic Algorithms" can be used to generate and optimize Course schedule using different programming languages. The following is the summary of findings using the system:

- Test results

1. **Functionality**: The test result showed that the Scheduling system performed well, passing all the test, and demonstrating its robustness and accuracy in generating and effectively optimizing faculty schedule. The testing procedure consisted of sending a variety of data via POST request to various endpoints to evaluate them.
2. **Compatibility:** The test result demonstrated the developed system's versatility and dependability in a variety of programming environments. The modules and packages developed successfully validated the system's easy integration and effectiveness across multiple programming languages

3. **Artificial Intelligent testing and evaluation:** The testing and evaluation of the developed scheduling system using genetic algorithm showed that it performed exceptionally well in optimizing and presenting the schedule in a chromosome and generating faculty schedule from the process of natural selection.

- Evaluation result

1. **Functional Suitability.** The evaluators rated the Scheduling system as highly acceptable for its ability to generate optimal class schedules, considering constraints like room availability, faculty availability, and avoiding conflicts. It also counts the units per course and converts the time table into PDF.
2. **Performance Efficiency.** The evaluators rated the Scheduling system as highly acceptable in terms of response rate, appropriateness of response, and creating conflict-free schedules efficiently.
3. **Compatibility.** The evaluators rated the Scheduling system as highly acceptable in terms of its versatility, dependability, and effectiveness across multiple programming environments and how it exchanges information with different programming languages.
4. **Usability.** The evaluators rated the Scheduling system as highly acceptable in terms of its appropriateness based on developers' needs for implementing it in their projects and satisfying interactions, accessibility, ease of learning, operating, and consistent layout for its documentation.
5. **Reliability.** The evaluators rated the Scheduling system as highly acceptable in terms of its consistency, performance, and reliability under varying conditions, accessibility,

and responsiveness to developers' request without interruptions, and efficiency in handling errors or unexpected events without compromising its functionality.

6. **Maintainability.** The evaluators rated the Scheduling system as highly acceptable regarding how easy it is to diagnose the system when failure occurs and how it offered convenient and efficient mechanisms for identifying and troubleshooting issues when they arise.
7. **Portability.** The evaluators rated the Scheduling system as highly acceptable regarding how the system can be easily used by installing the modules and packages created for Python, CSS, and HTML.

Conclusions

The following conclusions were derived from the above findings:

1. The Scheduling system was successfully designed to have the following features:
    a. Ability to encapsulating methods to fetch data from database that allows for easy access to rooms, faculty names, subjects, and departments.
    b. Ability to initialize schedules by assigning random faculty members to sections and subjects, choosing preferred time slots, and ensuring that the time slots are consistent with the defined periods in the database.
    c. The fitness of a schedule is inversely proportional to the number of conflicts. Conflicts can arise from overlapping schedules, room capacity issues, or faculty availability conflicts.

d. Can be implemented in different programming languages Python, CSS, and HTML, utilizing the modules and packages specifically created for each language.

e. Can handle form and CRUD operations that clear the cached schedule to ensure that any changes are reflected in newly generated time tables

f. Can convert the generated time tables into PDF version of the timetable using reportlab, providing a structured and printable format of the schedule.

2. The Scheduling system was successfully developed and built using VS Code and Python programming language.
3. The Scheduling system was tested and improved, which showed that the system is compatible and usable in different programming languages.
4. The Scheduling system was evaluated to be highly acceptable in terms of Functional Suitability, Performance Efficiency, Compatibility, Usability, and Reliability, which proves that the system can be a very helpful tool.

Recommendations

The following recommendations are put forward for further enhancement of the developed scheduling system.

1. **Responsive Design:** Ensure the system is fully responsive and works seamlessly across different devices (e.g., desktops, tablets, smartphones). This includes optimizing layouts for various screen sizes and ensuring touch-friendly interactions on mobile devices.
2. **Customizable Reports:** Allow users to customize reports based on their specific needs and export them in various formats (e.g., CSV, Excel, PDF). You can provide robust customizable report generation capabilities in your scheduling system, enhancing its functionality and user satisfaction.
3. **User Authentication and Authorization:** Enhance the authentication and authorization mechanisms to ensure only authorized personnel can access and modify schedules. This will improve the security and integrity of the system.
4. **Import data:** Ensure your scheduling system can query the database to fetch the imported data and use it for schedule generation and optimization. (e.g., Excel) This will enhance the system's functionality and improve its efficiency and usability.

Appendices

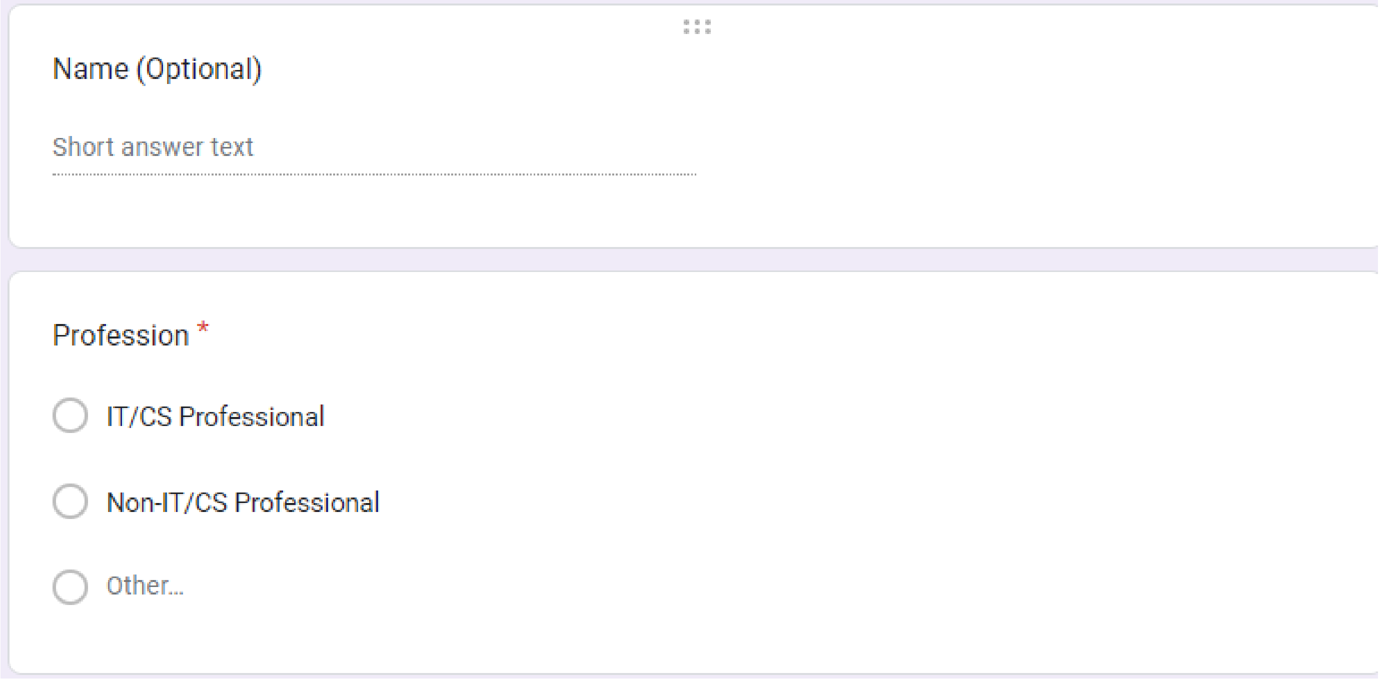
Developing a College of Science Scheduling System Using Genetic Algorithm

Good day!

We are 4th Year BS Computer Science students from Technological University of the Philippines - Manila. We would like to ask for a brief portion of your time to partake in evaluation of our thesis entitled " Developing a College of Science Scheduling System Using Genetic Algorithm".

The project's objectives are to manage scheduling complexity, optimize course scheduling, and adhere to university standards by developing a genetic algorithm-based scheduling system for a college of science. Further, it seeks to reduce scheduling conflicts, adapt to limitations such as teacher availability, evaluate resource efficiency, and offer suggestions for enhancement. The goal of the study is to show how genetic algorithms may improve student learning, academic administration, and course scheduling.

Data gathered through this form will only be used for results gathering and is covered by Republic Act 10173 also known as Data Privacy Act of 2012.



Name (Optional)

Short answer text

Profession *

○ IT/CS Professional

○ Non-IT/CS Professional

○ Other...

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 check if the software functionalities cover all specified tasks and user objectives

Functional Correctness check if the software provides results that are accurate based on the industry standard applications and mathematics

Functional Appropriateness check if the software functionalities can be used for accomplishment of specified tasks and objectives.

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 (such as CPU, memory, storage, network devices, energy, materials...) under specified conditions.

Time Behavior check if time taken for the software to process data and give responses to its users.

Resource Utilization check if Software utilization and management of resources when performing its functions.

Capacity check if maximum limitations of the software are enough to performs its functions.

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

Learnability check if the Degree to which the user can learn to operate the software with efficiency and effectiveness.

Operability Software design that affects the easiness of operating the system.

Reliability Degree to which a system, product or component performs specified functions under specified conditions for a specific period of time.

Availability check if the software design that affects the easiness of operating the system.

Fault Tolerance check if The system operates as intended despite the presence of hardware or software faults.

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.**

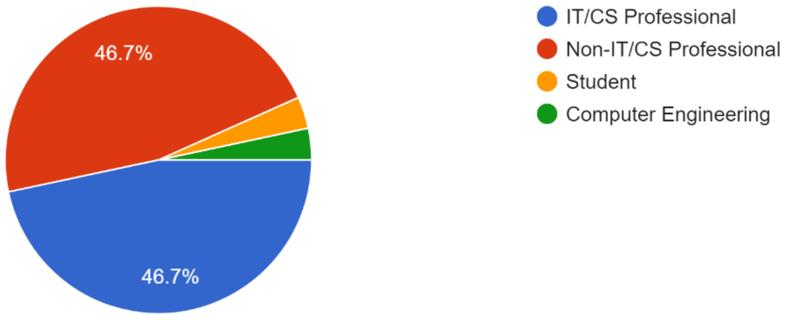
Reusability check if the software can be used to generate results that can be reused for other projects.

Modifiability check if the software can be modified efficiently without introducing defects or degrading existing product quality.

Testability Software's effectiveness and efficiency can easily be tested through the different test criteria.

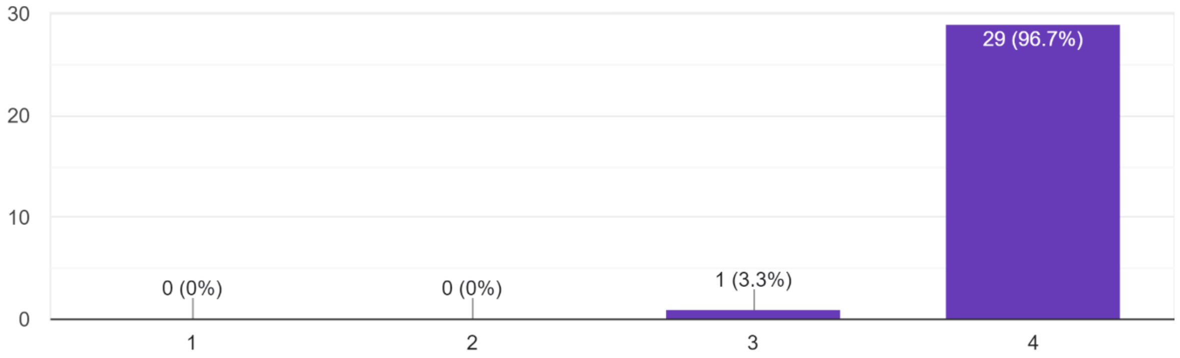
Profession

30 responses



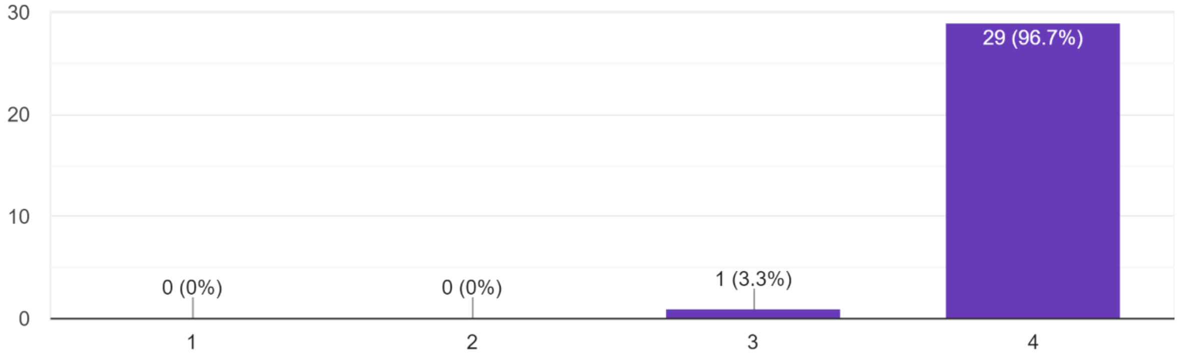
Functional Completeness

30 responses



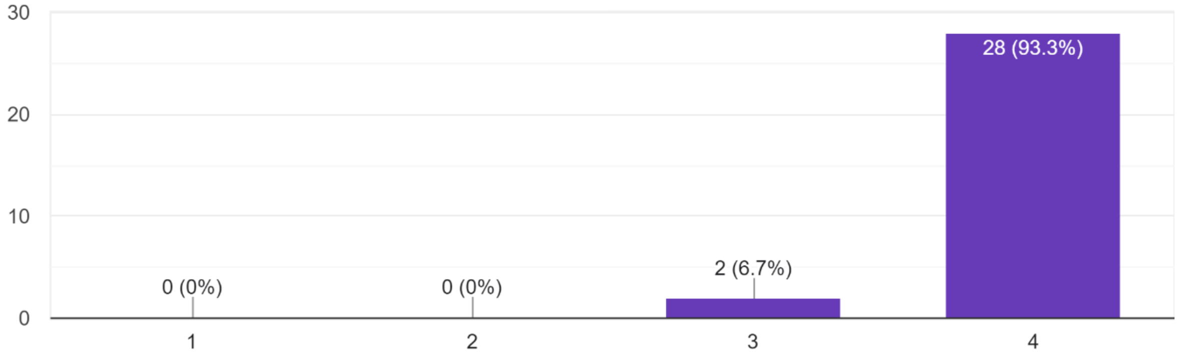
Functional Correctness

30 responses



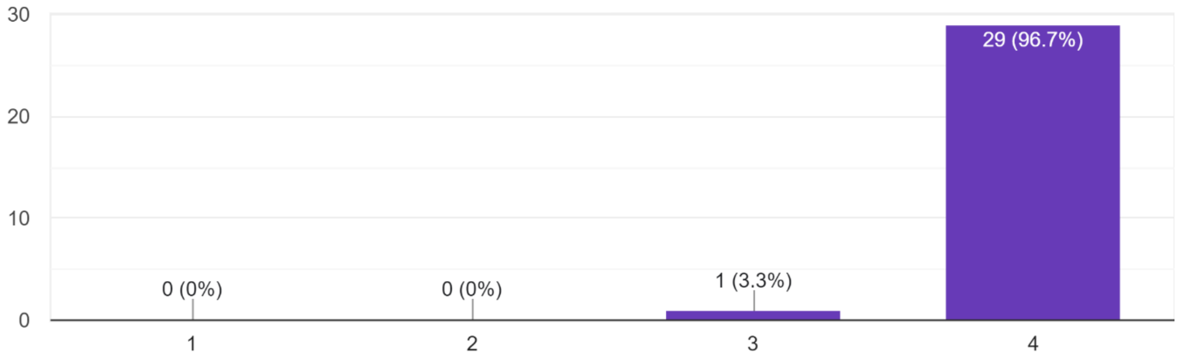
Functional Appropriateness

30 responses



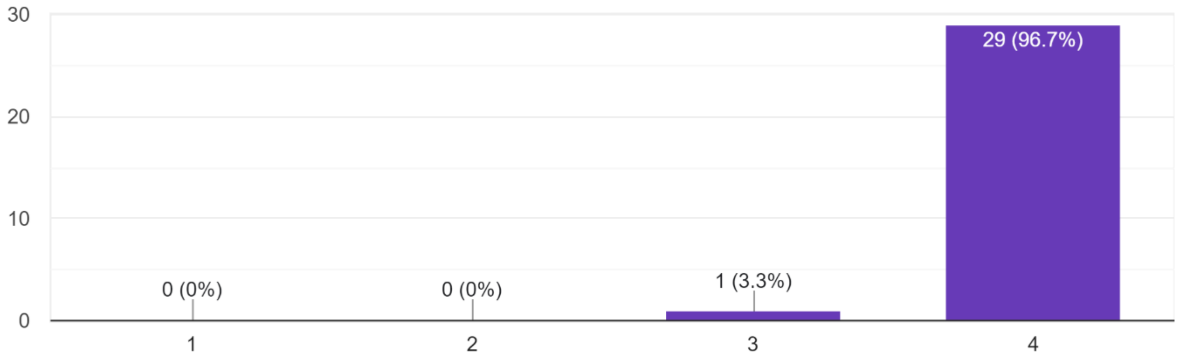
Time Behavior

30 responses



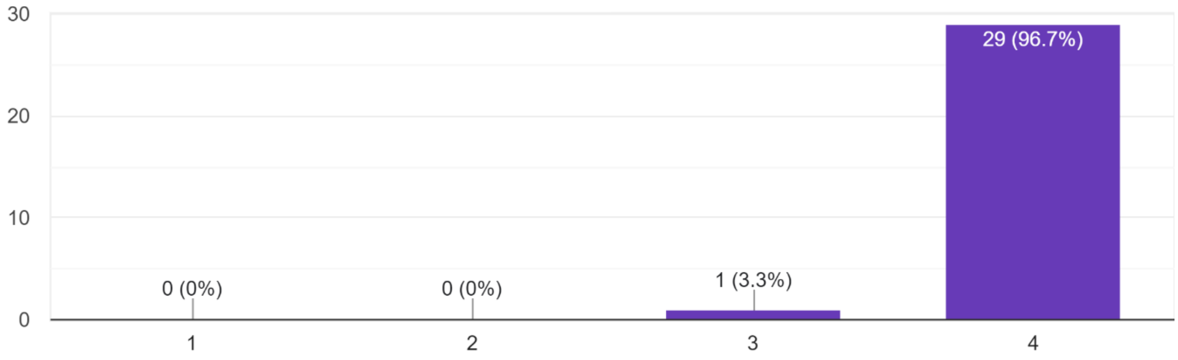
Resource Utilization

30 responses



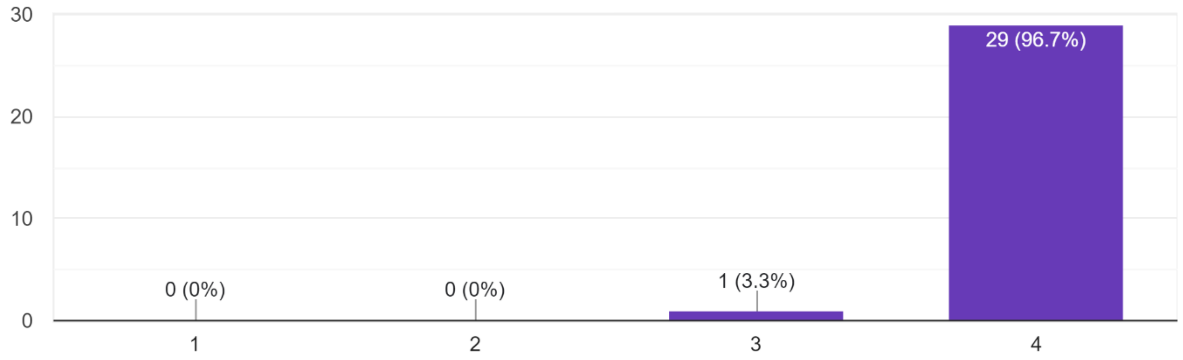
Capacity

30 responses



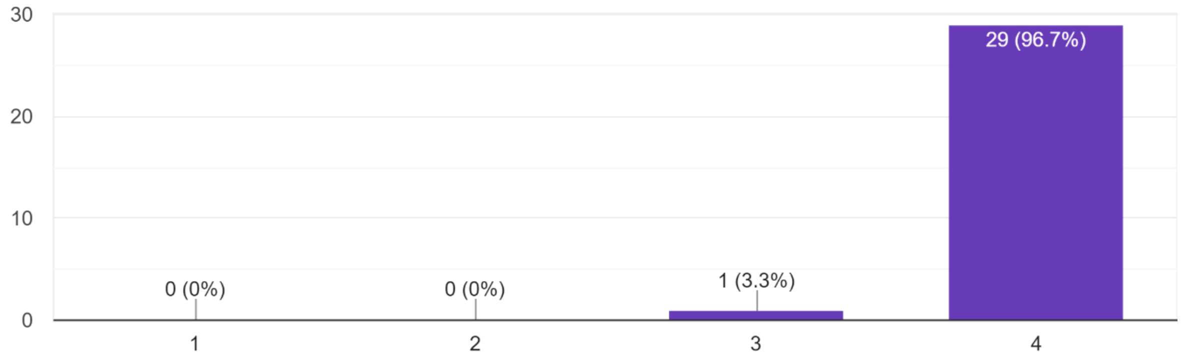
Learnability

30 responses



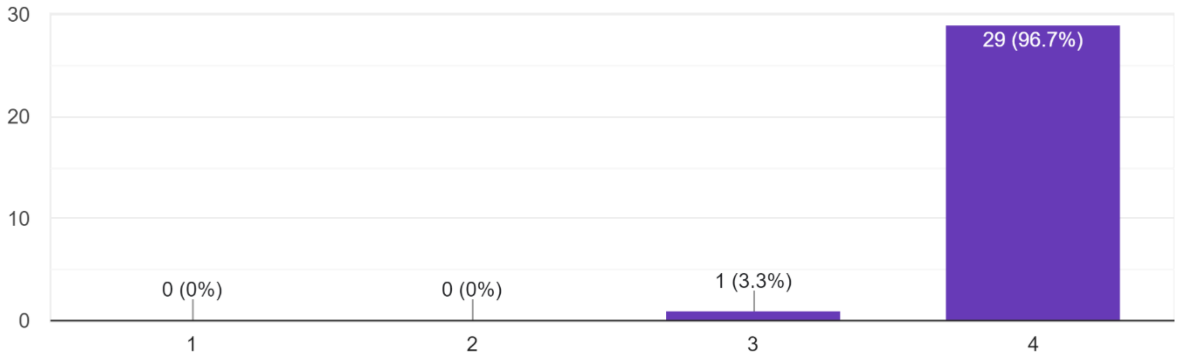
Operability

30 responses



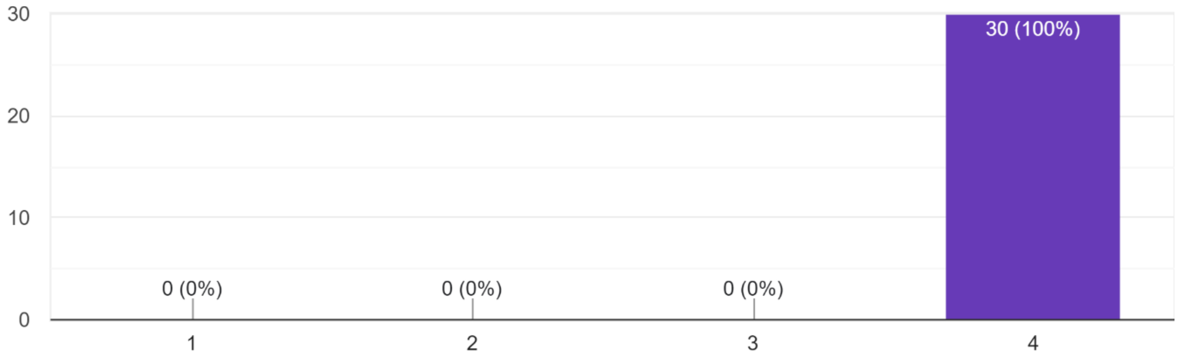
Availability

30 responses



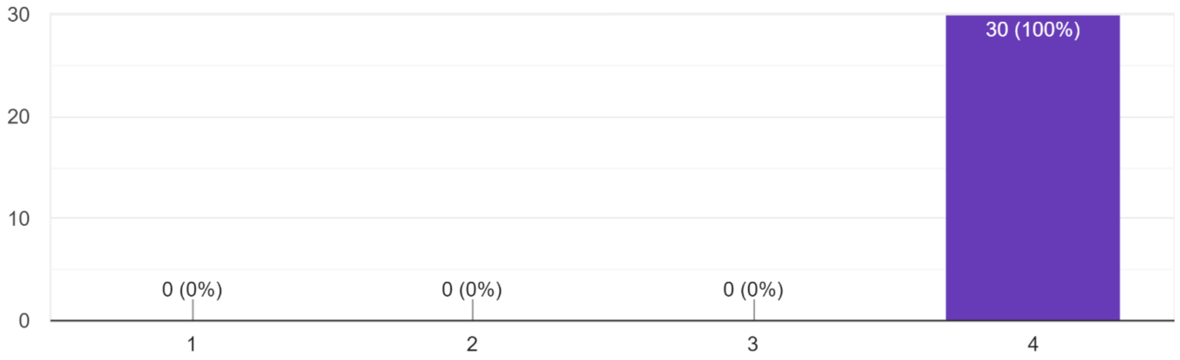
Fault Tolerance

30 responses



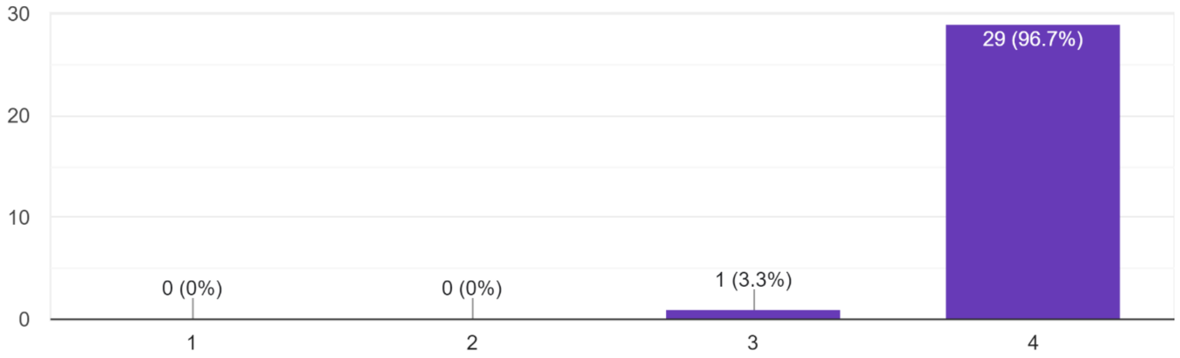
Reusability

30 responses



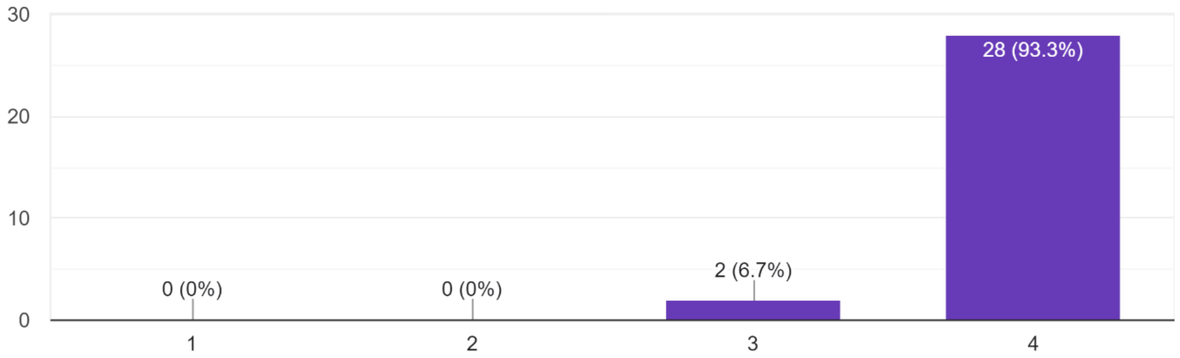
Modifiability

30 responses

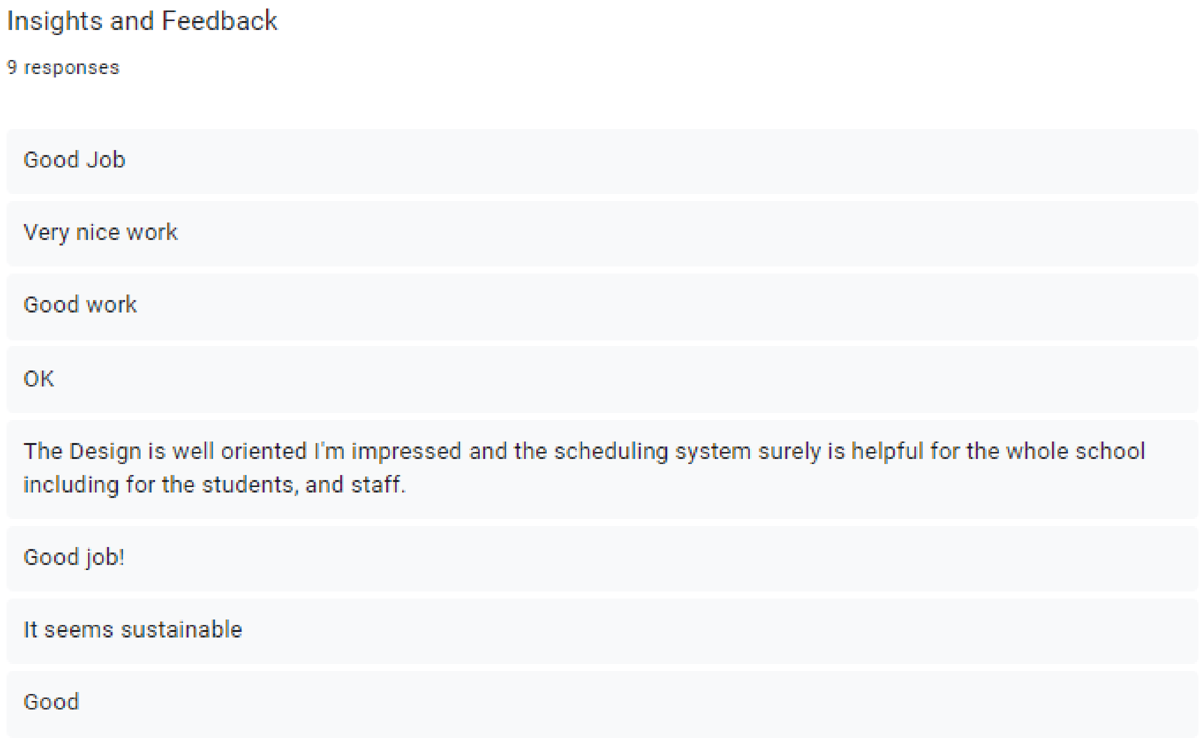


Testability

30 responses



Responses



Insights and Feedback

9 responses

Good Job

Very nice work

Good work

OK

The Design is well oriented I'm impressed and the scheduling system surely is helpful for the whole school including for the students, and staff.

Good job!

It seems sustainable

Good

References

1. Using a genetic algorithm optimizer tool to solve University timetable scheduling problem. (2007, February 1). IEEE Conference Publication | IEEE Xplore.
2. Solving timetable scheduling problem using genetic algorithms. (2003). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/1225396
3. Modified genetic algorithm for high school time-table scheduling with fuzzy time window. (2017, November 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8304115
4. Time table scheduling using Genetic Algorithms employing guided mutation. (2010, December 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/5705788
5. University time table scheduling using genetic artificial immune network. (2009, October 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/5329471
6. Tormos, P., Lova, A., Barber, F., Ingolotti, L., Abril, M., & Salido, M. A. (2008). A genetic algorithm for railway scheduling problems. In Studies in computational intelligence (pp. 255–276). https://doi.org/10.1007/978-3-540-78985-7_10
7. An Evolutionary Algorithm for Solving Academic Courses Timetable Scheduling Problem. Baghdad Sci.J [Internet]. 2022 Apr. 1 [cited 2024 May 14];19(2):0399. Available from: https://bsj.uobaghdad.edu.iq/index.php/BSJ/article/view/5309
8. Burke E.K., Elliman D.G. and Weare R.F. (2016) A Genetic Algorithm Based University Timetabling System", AISB Workshop on Evolutionary Computing,

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=74edcc0236b637903585f4d7a4ddff07d2ce99a7

9. Oh, J., & Wu, C. (2004). Genetic-algorithm-based real-time task scheduling with multiple goals. Journal of Systems and Software/˜ the œJournal of Systems and Software, 71(3), 245–258. https://doi.org/10.1016/s0164-1212(02)00147-4

10. Mortaza Abbaszadeh & Saeed Saeedvand. A Fast Genetic Algorithm for Solving University Scheduling Problem (2016). IAES International Journal of Artificial Intelligence (IJ-AI) https://www.researchgate.net/profile/Saeed-Saeedvand/publication/294428240_A_Fast_Genetic_Algorithm_for_Solving_University_Scheduling_Problem/links/56c0e6cf08aee5caccf678e2/A-Fast-Genetic-Algorithm-for-Solving-University-Scheduling-Problem.pdf

11. Genetic Algorithm for scheduling Time-Triggered traffic in Time-Sensitive networks. (2018, September 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8502515

12. Omara, F. A., & Arafa, M. (2010). Genetic algorithms for task scheduling problem. Journal of Parallel and Distributed Computing, 70(1), 13–22. https://doi.org/10.1016/j.jpdc.2009.09.009

13. Deris, S., Omatu, S., Ohta, H., & Saad, P. (2019). Incorporating constraint propagation in genetic algorithm for university timetable planning. Engineering Applications of Artificial Intelligence, 12(3), 241–253. https://doi.org/10.1016/s0952-1976(99)00007-x

14. A genetic algorithm approach to a general category project scheduling problem. (2020, February 1). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/740669/authors#authors

15. Faghihi, V., Reinschmidt, K. F., & Kang, J. (2014). Construction scheduling using Genetic Algorithm based on Building Information Model. Expert Systems With Applications, 41(16), 7565–7578. https://doi.org/10.1016/j.eswa.2014.05.047

16. R. Barman et al., "Automated train scheduling system using Genetic Algorithm," 2015 International Symposium on Advanced Computing and Communication (ISACC), Silchar, India, 2015, pp. 28-33, doi: 10.1109/ISACC.2015.7377310

17. Chan, W., Chua, D. K. H., & Kannan, G. (2016). Construction Resource Scheduling with Genetic Algorithms. Journal of Construction Engineering and Management, 122(2), 125–132. https://doi.org/10.1061/(asce)0733-9364(1996)122:2(125

18. Lee, C. Y., Piramuthu, S., & Tsai, Y. K. (2017). Job shop scheduling with a genetic algorithm and machine learning. International Journal of Production Research, 35(4), 1171–1191. https://doi.org/10.1080/002075497195605

19. K. Deb and P. Chakroborty, "Time Scheduling of Transit Systems With Transfer Considerations Using Genetic Algorithms," in Evolutionary Computation, vol. 6, no. 1, pp. 1-24, March 2018, doi: 10.1162/evco.2018.6.1.1.

20. S. Gupta, V. Kumar and G. Agarwal (2010) Task scheduling in multiprocessor system using genetic algorithm. (2010, February 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/5460728

21. Saptarini, N. G. a. P. H., Suasnawa, I. W., & Ciptayani, P. I. (2018). Senior high school course scheduling using genetic algorithm. Journal of Physics. Conference Series, 953, 012067. https://doi.org/10.1088/1742-6596/953/1/012067

22. S, R., & ER, N. (2013). Dynamic scheduling of data using genetic algorithm in cloud computing. International Journal of Computing Algorithm, 2(1), 11–15. https://doi.org/10.20894/ijcoa.101.002.001.003

23. Arash Ghorbannia Delavar, M. Nejadkheirallah and M. Motalleb, "A new scheduling algorithm for dynamic task and fault tolerant in heterogeneous grid systems using Genetic Algorithm," 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China, 2010, pp. 408-412, doi: 10.1109/ICCSIT.2010.5564753

24. V. Di Martino, "Sub optimal scheduling in a GRID using genetic algorithms," Proceedings International Parallel and Distributed Processing Symposium, Nice, France, 2003, pp. 7 pp.-, doi: 10.1109/IPDPS.2003.1213282.

25. R. C. Correa, A. Ferreira and P. Rebreyend, "Scheduling multiprocessor tasks with genetic algorithms," in IEEE Transactions on Parallel and Distributed Systems, vol. 10, no. 8, pp. 825-837, Aug. 2019, doi: 10.1109/71.790600.

26. K. P. Dahal, S. J. Galloway, G. M. Burt and J. R. McDonald, "Generation scheduling using genetic algorithm-based hybrid techniques," LESCOPE 01. 2001 Large Engineering Systems Conference on Power Engineering. Conference Proceedings. Theme: Powering Beyond 2021 (Cat. No.01ex490), Halifax, NS, Canada, 2001, pp. 74-78, doi: 10.1109/LESCPE.2021.941630.

27. E. Gil, J. Bustos and H. Rudnick, "Short-term hydrothermal generation scheduling model using a genetic algorithm," in IEEE Transactions on Power Systems, vol. 18, no. 4, pp. 1256-1264, Nov. 2013, doi: 10.1109/TPWRS.2003.819877.

28. U. Fissgus, "Scheduling using genetic algorithms," Proceedings 20th IEEE International Conference on Distributed Computing Systems, Taipei, Taiwan, 2020, pp. 662-669, doi: 10.1109/ICDCS.2020.840983.

29. Yao-Te Wang, Yu-Hsin Cheng, Ting-Cheng Chang and S. M. Jen, "On the application of data mining technique and genetic algorithm to an automatic course scheduling system," 2008 IEEE Conference on Cybernetics and Intelligent Systems, Chengdu, 2008, pp. 400-405, doi: 10.1109/ICCIS.2008.4670852.

30. L. Ozdamar, "A genetic algorithm approach to a general category project scheduling problem," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 29, no. 1, pp. 44-59, Feb. 2019, doi: 10.1109/5326.740669.

31. M. Aggarwal, R. D. Kent and A. Ngom, "Genetic algorithm based scheduler for computational grids," 19th International Symposium on High Performance Computing Systems and Applications (HPCS'05), Guelph, ON, Canada, 2015, pp. 209-215, doi: 10.1109/HPCS.2005.27.

32. O. Morandin, E. R. R. Kato, A. C. Deriz and D. S. Sanches, "A search method using Genetic Algorithm for production reactive scheduling of manufacturing systems," 2008 IEEE International Symposium on Industrial Electronics, Cambridge, UK, 2018, pp. 1843-1848, doi: 10.1109/ISIE.2008.4677072.

33. Wu Ying and Li Bin, "Job-shop scheduling using genetic algorithm," 2016 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No.96CH35929), Beijing, China, 2016, pp. 2014-2019 vol.3, doi: 10.1109/ICSMC.1996.565434.

34. A. Aminu et al., "Design and Implementation of An Automatic Examination Timetable Generation and Invigilation Scheduling System Using Genetic Algorithm," 2019 2nd International Conference on Applied Engineering (ICAE), Batam, Indonesia, 2019, pp. 1-5, doi: 10.1109/ICAE47758.2019.9221700.

35. M. R. Miryani and M. Naghibzadeh, "Hard real-time multiobjective scheduling in heterogeneous systems using genetic algorithms," 2009 14th International CSI Computer Conference, Tehran, Iran, 2009, pp. 437-445, doi: 10.1109/CSICC.2009.5349619.

36. H. Chen, J. Ihlow and C. Lehmann, "A genetic algorithm for flexible job-shop scheduling," Proceedings 2019 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Detroit, MI, USA, 2019, pp. 1120-1125 vol.2, doi: 10.1109/ROBOT.1999.772512.

37. R. Ibrahim, A. A. Bani Amin and M. Zainuri Saringat, "The Z Specification for Exam Scheduling System (ESS) thru Genetic Algorithm," 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 2020, pp. 1-5, doi: 10.1109/ICCIT-144147971.2020.9213749.

38. J. Yu and R. Buyya, "A budget constrained scheduling of workflow applications on utility Grids using genetic algorithms," 2006 Workshop on Workflows in Support of Large-Scale Science, Paris, France, 2006, pp. 1-10, doi: 10.1109/WORKS.2006.5282330.

39. M. Agarwal and G. M. S. Srivastava, "A genetic algorithm inspired task scheduling in cloud computing," 2016 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2016, pp. 364-367, doi: 10.1109/CCAA.2016.7813746.

40. C. -R. Chen, M. -J. Lan, C. -C. Huang, Y. -Y. Hong and S. H. Low, "Demand Response Optimization for Smart Home Scheduling Using Genetic Algorithm," 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 2013, pp. 1461-1465, doi: 10.1109/SMC.2013.252.

41. H. Murao and S. Kitamura, "Online scheduling of a multi-robot system by using genetic algorithms," IEEE International Symposium on Industrial Electronics. Proceedings. ISIE'98

(Cat. No.98TH8357), Pretoria, South Africa, 2018, pp. 709-713 vol.2, doi: 10.1109/ISIE.1998.711710.

42. Q. M. Hussein and A. N. Hasoon, "Dynamic process scheduling using genetic algorithm," 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), Baghdad, Iraq, 2017, pp. 111-115, doi: 10.1109/NTICT.2017.7976116.

43. A. Raghavan, P. Maan and A. K. B. Shenoy, "Optimization of Day-Ahead Energy Storage System Scheduling in Microgrid Using Genetic Algorithm and Particle Swarm Optimization," in IEEE Access, vol. 8, pp. 173068-173078, 2020, doi: 10.1109/ACCESS.2020.3025673.

44. X. Zuo, C. Chen, W. Tan and M. Zhou, "Vehicle Scheduling of an Urban Bus Line via an Improved Multiobjective Genetic Algorithm," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 1030-1041, April 2015, doi: 10.1109/TITS.2014.2352599.

45. S. G. Ahmad, E. U. Munir and W. Nisar, "PEGA: A Performance Effective Genetic Algorithm for Task Scheduling in Heterogeneous Systems," 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, Liverpool, UK, 2012, pp. 1082-1087, doi: 10.1109/HPCC.2012.158.

46. T. Wang, Z. Liu, Y. Chen, Y. Xu and X. Dai, "Load Balancing Task Scheduling Based on Genetic Algorithm in Cloud Computing," 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, Dalian, China, 2014, pp. 146-152, doi: 10.1109/DASC.2014.35.

47. Zhongni Zheng, Rui Wang, Hai Zhong and Xuejie Zhang, "An approach for cloud resource scheduling based on Parallel Genetic Algorithm," 2011 3rd International Conference on

Computer Research and Development, Shanghai, 2011, pp. 444-447, doi: 10.1109/ICCRD.2011.5764170.

48. M. T. Jensen, "Generating robust and flexible job shop schedules using genetic algorithms," in IEEE Transactions on Evolutionary Computation, vol. 7, no. 3, pp. 275-288, June 2003, doi: 10.1109/TEVC.2003.810067.

49. M. Pahlevan and R. Obermaisser, "Genetic Algorithm for Scheduling Time-Triggered Traffic in Time-Sensitive Networks," 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 2018, pp. 337-344, doi: 10.1109/ETFA.2018.8502515.

50. Soon-mi Han, Seung-woo Beak, Kyuem-rae Cho, Dae-woo Lee and Hae-dong Kim, "Satellite mission scheduling using genetic algorithm," 2008 SICE Annual Conference, Tokyo, 2008, pp. 1226-1230, doi: 10.1109/SICE.2008.4654845.

Elmer II B. Absalon

Researcher

MOBILE:
09085720211

EMAIL:
lmrbsln@gmail.com

PERSONAL INFORMATION

Age: 23

Date of Birth: June 01, 2001

Address: 298 Atherton St. Brgy. North Fairview, QC

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA

Course: Computer Science

2020 - 2024

METRO MANILA COLLEGE

Strand: STEM

2018-2020

NORTH FAIRVIEW HIGH SCHOOL

2014 - 2018

NORTH FAIRVIEW ELEMENTARY SCHOOL

2008-2014

PROFESSIONAL SKILLS

- Basic Programming Language (C, C++, Python and Visual Basic)
- Web Development (HTML, CSS, JAVA, PHP)
- Arduino

Ivan Joshua C. Gepanayao

Researcher

MOBILE:
09098281399

EMAIL:
ivanjoshuaw12@gmail.com

PERSONAL INFORMATION

Age: 22

Date of Birth: August 08, 2001

Address: Blk. 51 Lot 1 Area 3, Kaunlaran Village, Caloocan City

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA

Course: Computer Science

2020 - 2024

MANILA CENTRAL UNIVERSITY

Strand: STEM

2014-2018

MANILA CENTRAL UNIVERSITY

2008-2014

PROFESSIONAL SKILLS

- Basic Programming Language (C, C++, Python and Visual Basic)
- Web Development (HTML, CSS, JAVA, PHP)
- Basic Mobile Application (Android Studio)

Ronne Eduard P. Reyes

Chief of Research

MOBILE:
09913497886

EMAIL:
ronneeduard01@gmail.com

PERSONAL INFORMATION

Age: 23 **Date of Birth:** January 15, 2001

Address: 1283 Int. 7 Tambunting St. Sta Cruz Manila

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES MANILA

Course: Computer Science

San Marcelino St. Ayala Blvd. Ermita, Manila
2019-2024

PHILIPPINE COLLEGE OF SCIENCE AND HEALTH

Strand: STEM

1813 Claro M. Recto Ave., Manila
2017 - 2019

FELIPE G. CALDERON

hermosa Street, Brgy. 201, Manila, Metro Manila 2013-2017

MARULAS ELEMENTARY SCHOOL

Pio Valenzuela St. Marulas, Valenzuela City
2007-2014

PROFESSIONAL SKILLS

- Basic Programming Language (C, C++, Python and Visual Basic)
- Web Development (HTML, CSS, JAVA, PHP)
- Basic Mobile Application (Android Studio)

Carl Matthew R. Tagapia

Chief Developer

MOBILE:
09277389930

EMAIL:
carlmatthew605@gmail.com

PERSONAL INFORMATION

Age: 22

Date of Birth: July 17, 2001

Address: Bistekville 10 Buenamar Drive, Brgy. Nova Proper, Novaliches, QC

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA
Course: Computer Science
2020 - 2024

METRO MANILA COLLEGE
Strand: STEM
2018-2020

DDONA ROSARIO HIGH SCHOOL
2014 - 2018

DONA ROSARIO ELEMENTARY SCHOOL
2008-2014

PROFESSIONAL SKILLS

- Basic Programming Language (C, C++, Python and Visual Basic)
- Web Development (HTML, CSS, JAVA, PHP)
- Bootstrap and MIPSAssemply

Janina Marie P. Tonelete

Researcher

MOBILE:	**EMAIL:**
09077387052	*janinatonelete84@gmail.com*

PERSONAL INFORMATION

Age: 22

Date of Birth: September 08, 2001

Address: Longos, Malabon City

EDUCATION

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA

Course: Computer Science

2020 - 2024

PROFESSIONAL SKILLS

- Basic Programming Language (C, C++, Python and Visual Basic)
- Web Development (HTML, CSS, JAVA, PHP)
- MS Office, Adobe Premier