CHAPTER ONE
INTRODUCTION

1.1 BACKGROUND OF THE STUDY

Bus Ticket Reservation System enables the bus company's customer to buy bus ticket online; this is an easy method which saves a lot of time, it will enable customers in their comfort zone, to search for available tickets and get their data easily without any prior experience with queuing at the counter.

E-ticket is the easier and quickest way to take bus. The online system is a new system because it hasn't exists in Bus Company and even in AKTC. Currently, staffs at the bus ticket counter are using an internal system to sell ticket at the counter. Customers are unable to buy bus ticket online at this moment and have to go to the counter to buy bus ticket. Sometimes, customer needs to queue up a long queue to buy bus ticket and ask for information. This brings a lot of inconvenience to the customers.

With the Internet now routinely being used as a resource for Bus Ticket Reservation, customer can buy bus ticket, make payment, and ask for information online easily. Furthermore, staff can sell bus ticket using Bus Ticket Reservation System after checking bus ticket availability for the customer and print the bus ticket to the customer that queue up in the counter.

Examination timetable scheduling is a very important process in education institutions. The main function is to schedule examinations to timeslots and rooms over a specific period while satisfying a set of constraints. The University has two semesters per each academic year. Each semester is made up of up to fifteen weeks of teaching, followed by two weeks of examination. There are two examination sessions per day except on Sunday where student are allowed to rest. Examinations are mostly three hours long with a few exceptions which deviates for half an hour or two hours. There are an increasing number of courses which cut across faculties and Universitys-wide courses which are offered to more than a thousand students at the same time. The problem is also complicated by the freedom of choices by students on optimal courses, where students have wide range of choices which cut across department and faculties.

The Examination Timetable Problem (ETP) is usually modeled as an NP-Hard (non-deterministic polynomial time hard problem) combinatorial optimization problem. The problem demands that a given number of exams are scheduled in a limited number of periods and venues in such a way that no student had more than one exam at a time and other constraint are satisfied. Although consideration was based on exams timetabling, the ideas presented here can be extended to many other applications, which include not only other scheduling problems but also multi-criteria problems. In general, the reason to present an application on exams time tabling is justified by the affiliation of the auditors and their awareness of the increased difficulty that some recent strategies have introduced in this academic task. Just as an example, monitoring the tendency towards the flexibility of curricular and the increase of the number of students enrolled in each course.

1.2 TIMETABLING

Timetable problems are assignment activities with time-based constraints. Timetabling is used for scheduling different activities, including courses, crew assignments in airline industries, and matches in sports. In general they are referred to as scheduling problems.

For this project, we can divide timetabling in educational institutions into two major areas:

- **School Timetable**: In this situation, all classes for a week (or more) are scheduled, subject to instructors' resource constraints and/or students' course requirements.
- **Examination Timetable:** This schedules the examinations to meet room availability constraints and to avoid overlapping examination times for each student. Furthermore, there are three major groups who directly or indirectly provide the information for the process.
- **Administration Office:** defines the specific standards, with regards to maximum and minimum number of courses, students are permitted to select for each semester and timetabling.
- **Departments:** department may have demands for specific rooms or timeslots for their examinations. For instance, they may need to assign large examinations early in timetables to facilitate efficient marking, or they may need a room with computer facilities.
- **Students:** usually prefer to have days off between consecutive examinations, may be concerned about the examinations' scheduling order

1.2 STATEMENT OF THE PROBLEM

After analyzing the existing system, observations were made that a lot of problems are being encountered by the examination coordinators due to the manual method being employed in constructing examination time table and invigilation scheduling. The setback encountered includes;

- Time consuming.
- Problem of bad handwriting.
- The system is boring and cumbersome because of its repetitive nature.
- Inaccuracy of information; due to the manual method of the processing.

1.3 AIM AND OBJECTIVES OF THE PROJECT

The aim of this project is to develop an examination timetable scheduling system using Genetic Algorithm that will be useful for our educational institutions.

The following are the set objectives:

- To study the existing method of examination time table generation
- To design a system that will schedule courses, venues, invigilators and supervisors
- To implement the designed system

1.4 METHODOLOGY

The tools engaged in developing the new system include JAVA and XAMP-MYSQL.

The proposed system has some features such as:

- Accuracy in handling of data
- The volume of paper work will be greatly reduced.
- Fast rate of operation, i.e. timetables are generated quickly
- Flexibility (i.e. it can be accessed at any time)
- Easy way to back up or duplicating data in CD's in case of data loss
- Better storage and faster retrieval system
- Errors in the reports will be greatly minimized

1.5 SIGNIFICANCE OF THE PROJECT

The significance of the research work is to curb the examination timetable problems that may arise in the future, based on the conclusion and recommendation made by the researcher. The findings of this project will enable us to understand the importance of good examination timetable scheduling system for the tertiary institution easily and effectively. Also with the use of appropriate examination timetable data, simulation of examination timetable of various resources combination so as to encourage better timetable planning and information gathering is desirable. The following are the significance of study.

- To minimize the length of examination period and enhance student academic performance and also to allocate invigilators to time and venues.
- To give technical knowledge and competence to each student. Since the current system does not have a seemed back-up for files, information may be lose and there will be difficult to fully or partially replace or reproduce data.
- The current system is time consuming in terms of processing and file retrieval. Therefore, the computerized system is expected to enhance speed of creating timetable and organizing examinations.
- Manual processing of information is difficult, therefore designing of automated application as a remedy is better.

1.6 LIMITATIONS OF THE PROJECT

The project is developed to cover the fixing of examination timetables for all students but limited to the school of science and technology at Moshood Abiola University, Abeokuta. The system can be implemented in any other tertiary institution, which can be achieved by merely adjusting the input design of the program.

1.7 SCOPE OF THE STUDY

In this project, attention is focused on formulating mathematical models for the examinations timetable for the school of science and technology at Moshood Abiola University, Abeokuta. This will act as a benchmark for testing heuristic algorithms (describe an algorithm that modifies itself in response to the user) and help future reformations of the problem models. The Examination Timetabling Problems (ETP) differs considerably from the University course scheduling problem.

1.8 DEFINITION OF TERMS

- **Examination:** A formal test involving answering or writing in a particular place at set out time.
- **Optimization:** the design and operation of a system or process to make it as good as possible
- **Slot:** A space where some information will display such as Time, Courses, Invigilators, Venues, etc
- **Simulated:** Made to imitate manual outlook
- **Automation:** The act of converting the controlling of a system or approach into an electronic controls by using computer
- **Timetable:** This is a table of events arranged according to the time when they take place.
- **Genetic Algorithm**: Genetic Algorithm (GA) is a model of machine learning which derived its behavior from metaphor of the process(s) of EVOLUTION in natural sciences.
- **Front-end Application:** This is an application that users interact with directly.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

A review of airline seat reservation was examined with the purpose of accessing the structure of the current ways of seat reservation and designing a web-based computer system solution and could limit to the barest minimum some of the problems identified in the current method as well improvement would come in the form of better data management, data availability and information dissimilation. More so, computerized reservation system is aimed at ease of use and efficiency. Relevant information was gotten through a number of interviews with airline offers and travel agency officials as well as various internet services. Top down design methodology was used for the design. The computerization of reservation system was developed using web resources such as php, html, jquery, jscript. The system was design and tested running. It will curtail the difficulty in booking for airline ticket manually. It is recommended that adequate research should be adopted on this topic. Also, relevance agencies should make relevant policies to aid poorly managed airline. (Roberts, 2002).

The lecture timetabling problem is a typical scheduling problem that appears to be a tedious job in every academic institute once or twice a year. The problem involves the scheduling of classes, students, teachers and rooms at a fixed number of time-slots, subject to a certain number of constraints. An effective timetable is crucial for the satisfaction of educational requirements and the efficient utilization of human and space resources, which make it an optimization problem. Traditionally, the problem is solved manually by trial and hit method, where a valid solution is not guaranteed. Even if a valid solution is found, it is likely to miss far better solutions. These uncertainties have motivated for the scientific study of the problem, and to develop an automated solution technique for it. The problem is being studied for last more than four decades, but a general solution technique for it is yet to be formulated (Datta D. et.al, 2006).

Timetabling problem is one of the hardest problem areas already proven to NP complete and it is worthy of note that as educational institutions are challenged to grow in number and complexity, their resources and events are becoming harder to schedule (Ossam Chohan, 2009).

2.2 REVIEW OF RELEVANT THEORIES AND TECHNOLOGIES

Solutions to timetabling problems have been proposed since the 1980s. Research in this area is still active as there are several recent related papers in operational research and artificial intelligence journals. This indicates that there are many problems in timetabling that need to be solved in view of the availability of more powerful computing facilities and advancement of information technology (S.B. Deris et.al, 1997). The problem was first studied by Gotlieb (1962), who formulated a class-teacher timetabling problem by considering that each lecture contained one group of students, one teacher, and any number of times which could be chosen freely. Since then the problem is being continuously studied using different methods under different conditions. Initially it was mostly applied to schools (de Gans, 1981; Tripathy, 1984). Since the problem in schools is relatively simple because of their simple class structures, classical methods, such as linear or integer programming approaches (Lawrie, 1969; Tripathy, 1984), could be used easily. However, the gradual consideration of the cases of higher secondary schools and universities, which contain different types of complicated class-structures, is increasing the complexity of the problem. As a result, classical methods have been found inadequate to handle the problem, particularly the huge number of integer and/or real variables, discrete search space and multiple objective functions.

There are two main problems in timetabling. The first one is related to the combinatorial nature of the problems, where it is difficult to find an optimal solution because it is impossible to enumerate all nodes in such a large search space. The second one is related to the dynamic nature of the problems where variables and constraints are changing in accordance with the development of an organization (S.B. Deris et al., 1997). Therefore, a timetabling system must be flexible, adaptable and portable, otherwise the users will not use the system optimally or even as decision aids such as for storing, retrieving, and printing timetables, when the timetable planning decisions are made manually. In addition, most of the universities adopting a semester system give freedom to students to choose subjects provided that all pre-requisites are satisfied. This situation further complicates the construction of a timetable.

Various techniques have been proposed to solve timetabling problems. These techniques are neural networks (Gianoglio P, 1990), heuristics (Wright M, 1996), graph coloring, integer programming, Genetic Algorithms (Burke E. et al., 1994; Paechter B. et al., 1994), knowledge-

based, and constraint logic programming (Lajos, 1995). The models formulated by some of these techniques cannot be easily reformulated or customized to support changes, hence the selection of the genetic algorithm for the implementation of this project.

2.3 BASICS FOR A GENETIC ALGORITHM

i. A number, or population, of guesses of the solution to the problem.
ii. A way of determining the states of generated solutions i.e. calculating how well or bad the individual solutions within the population are.
iii. A method for mixing fragments of the better solutions to form new, on average even better solutions.
iv. A mutation operator to avoid permanent loss of diversity within the solutions.

Concisely stated, a genetic algorithm is a programming technique that mimics biological evolution as a problem-solving strategy. Given a specific problem to solve, the input to the GA is a set of potential solutions to that problem, encoded in some fashion, and a metric called a fitness function that allows each candidate to be quantitatively evaluated. These candidates may be solutions already known to work, with the aim of the GA being to improve them, but more often they are generated at random. The GA then evaluates each candidate according to the fitness function. In a pool of randomly generated candidates, of course, most will not work at all, and these will be deleted. However, purely by chance, a few may hold promise - they may show activity, even if only weak and imperfect activity, toward solving the problem. These promising candidates are kept and allowed to reproduce. Multiple copies are made of them, but the copies are not perfect; random changes are introduced during the copying process.

These digital offspring then go on to the next generation, forming a new pool of candidate solutions, and are subjected to a second round of fitness evaluation. Those candidate solutions which were worsened, or made no better, by the changes to their code are again deleted; but again, purely by chance, the random variations introduced into the population may have improved some individuals, making them into better, more complete or more efficient solutions to the problem at hand. Again these winning individuals are selected and copied over into the next generation with random changes, and the process repeats. The expectation is that the average fitness of the population will increase each round, and so by repeating this process for hundreds or thousands of rounds, very good solutions to the problem can be discovered. As

astonishing and counterintuitive as it may seem to some, genetic algorithms have proven to be an enormously powerful and successful problem-solving strategy, dramatically demonstrating the power of evolutionary principles.

Genetic algorithms have been used in a wide variety of fields to evolve solutions to problems as difficult as or more difficult than those faced by human designers. Moreover, the solutions they come up with are often more efficient, more elegant, or more complex than anything comparable a human engineer would produce. In some cases, genetic algorithms have come up with solutions that baffle the programmers who wrote the algorithms in the first place (Adam, 2004).

2.4 METHODS OF REPRESENTATION

Before a genetic algorithm can be put to work on any problem, a method is needed to encode potential solutions to that problem in a form that a computer can process. One common approach is to encode solutions as binary strings: sequences of 1's and 0's, where the digit at each position represents the value of some aspect of the solution (Fleming et. al., 2002). Another, similar approach is to encode solutions as arrays of integers or decimal numbers, with each position again representing some particular aspect of the solution. This approach allows for greater precision and complexity than the comparatively restricted method of using binary numbers only and often "is intuitively closer to the problem space" (Fleming et. al., 2002).

A third approach is to represent individuals in a GA as strings of letters, where each letter again stands for a specific aspect of the solution. One example of this technique is Hiroaki Kitano's "grammatical encoding" approach, where a GA was put to the task of evolving a simple set of rules called a context-free grammar that was in turn used to generate neural networks for a variety of problems (Mitchell, 1996). The advantage of the three methods above is that they make it easy to define operators that cause the random changes in the selected candidates: flip a 0 to a 1 or vice versa, add or subtract from the value of a number by a randomly chosen amount, or change one letter to another.

Another strategy, developed principally by John Koza of Stanford University and called genetic programming, represents programs as branching data structures called trees (Koza et. al., 2003).

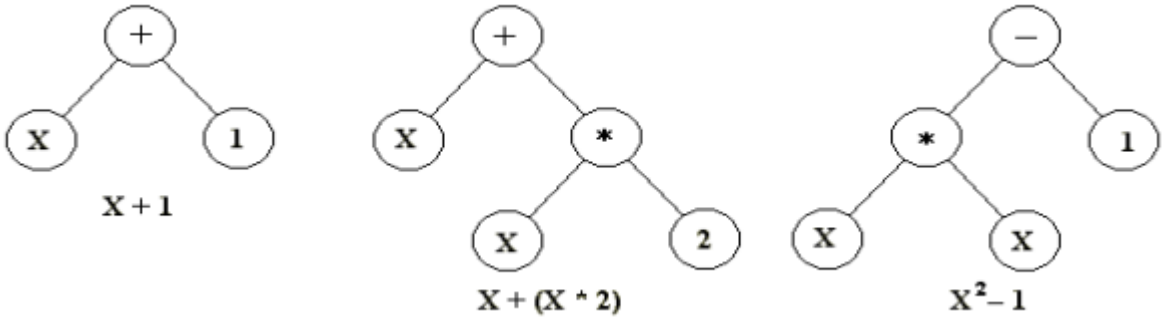


Fig 2.1: Three simple program trees of the kind normally used in genetic

It is important to note that evolutionary algorithms do not necessarily represent candidate solutions as data strings of fixed length. Though some represent them this way, but others do not; e.g. Kitano's grammatical encoding discussed above can be efficiently scaled to create large and complex neural networks, and Koza's genetic programming trees can grow arbitrarily large as necessary to solve whatever problem they are applied to.

2.5 METHODS OF SELECTION

There are many different techniques which a genetic algorithm can use to select the individuals to be copied over into the next generation, but listed below are some of the most common methods. Some of these methods are mutually exclusive, but others can be and often are used in combination.

- **Elitist selection:** The fittest members of each generation are guaranteed to be selected. (Most GAs doesn't use pure elitism, but instead use a modified form where the single best or a few of the best individuals from each generation are copied into the next generation just in case nothing better turns up.)
- **Fitness-proportionate selection:** More fit individuals are more likely, but not certain, to be selected.
- **Roulette-wheel selection:** A form of fitness-proportionate selection in which the chance of an individual's being selected is proportional to the amount by which its fitness is greater or less than its competitors' fitness. (Conceptually, this can be represented as a game of roulette - each individual gets a slice of the wheel, but more fit ones get larger slices than less fit ones. The wheel is then spun, and whichever individual "owns" the section on which it lands each time is chosen).

- **Scaling selection:** As the average fitness of the population increases, the strength of the selective pressure also increases and the fitness function becomes more discriminating. This method can be helpful in making the best selection later on when all individuals have relatively high fitness and only small differences in fitness distinguish one from another.
- **Tournament selection:** Subgroups of individuals are chosen from the larger population, and members of each subgroup compete against each other. Only one individual from each subgroup is chosen to reproduce.
- **Rank selection:** Each individual in the population is assigned a numerical rank based on fitness, and selection is based on this ranking rather than absolute difference in fitness. The advantage of this method is that it can prevent very fit individuals from gaining dominance early at the expense of less fit ones, which would reduce the population's genetic diversity and might hinder attempts to find an acceptable solution. Generational selection: The offspring of the individuals selected from each generation become the entire next generation. No individuals are retained between generations.
- **Steady-state selection:** The offspring of the individuals selected from each generation go back into the pre-existing gene pool, replacing some of the less fit members of the previous generation. Some individuals are retained between generations.
- **Hierarchical selection:** Individuals go through multiple rounds of selection each generation. Lower-level evaluations are faster and less discriminating, while those that survive to higher levels are evaluated more rigorously. The advantage of this method is that it reduces overall computation time by using faster, less selective evaluation to weed out the majority of individuals that show little or no promise, and only subjecting those who survive this initial test to more rigorous and more computationally expensive fitness evaluation.

2.6 METHODS OF CHANGE

Once selection has chosen fit individuals, they must be randomly altered in hopes of improving their fitness for the next generation. There are two basic strategies to accomplish this. The first and simplest is called mutation. Just as mutation in living things changes one gene to another, so mutation in a genetic algorithm causes small alterations at single points in an individual's code.

Fig. 2.2: Diagram showing the effect of mutation on an individual in a population of 8-bit

The second method is called crossover, and entails choosing two individuals to swap segments of their code, producing artificial "offspring" that are combinations of their parents. This process is intended to simulate the analogous process of recombination that occurs to chromosomes during sexual reproduction (Adam, 2004). Common forms of crossover include single-point crossover, in which a point of exchange is set at a random location in the two individuals' genomes, and one individual contributes all its code from before that point and the other contributes all its code from after that point to produce an offspring, and uniform crossover, in which the value at any given location in the offspring's genome is either the value of one parent's genome at that location or the value of the other parent's genome at that location, chosen with 50/50 probability.

2.7 APPLICATION OF GENETIC ALGORITHMS IN THIS RESEARCH

Having considered the basis for a genetic algorithm, the outline below highlights the applications of the proposed system in generating examination timetables scheduling (with school of science and technology, Moshood Abiola Polytechic as case study). The timetabling problem is a combinatorial optimization problem (COP) and in order to find a very comprehensive mathematical framework to describe it (and also tackle its NP-hardness), hence the introduction of a highly abstract concept of heuristics (genetic algorithms). The basic property of the timetable problem is the attempt of the genetic algorithm to optimize a function over a discrete structure with many independent variables. The relation between the choices made in the discrete domain and the effects on the objective function value are usually complex and frequently not easy to trace. The unifying framework for COP’s is the Constraint Satisfaction Problem (CSP) in conjunction with the optimization of an objective function (Kostuch, 2003).

It is worthy of note that even though the timetabling problem is treated as an optimization problem, there is actually no fixed objective function, the function that exists is used as an arbitrary measure to check for optimized solutions and degree of constraints satisfaction

(Abramson, 1992). Once the objectives and constraints are specified, genetic algorithms offer the ultimate scenarios of good timetable solutions through evolution processes even though the complexity of assignment is totally dependent on the number of instances and number of constraints. Hence the algorithm considered for use in the proposed system is a scaled down version of the Hybrid Genetic algorithm for the construction of examination timetables developed for the University of Nottingham. The concept though developed for examination timetabling, can be adapted to fit the construction of course timetables. The genetic algorithm employed combines two heuristic algorithms, the first finding a non-conflicting set of exams and the second assigning the selected exam to rooms. The process is repeated until all exams have been scheduled without conflicts (Rupert, 1995).

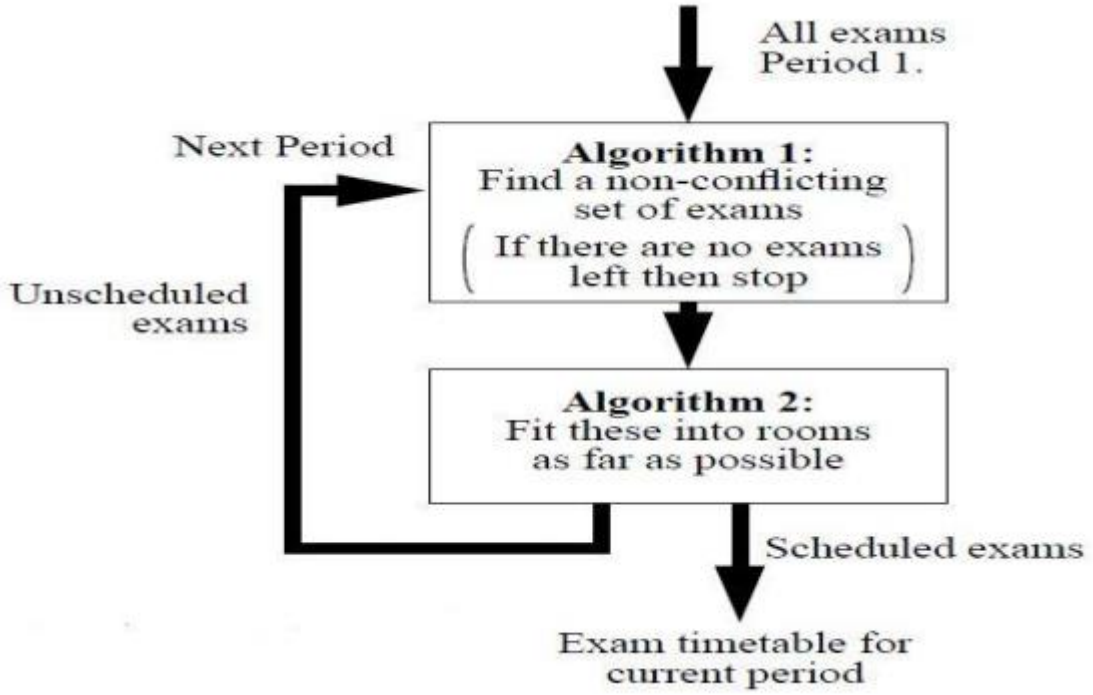


Fig. 2.3: Diagram depicting the Hybrid Genetic Algorithm used in generating time table

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 SYSTEM ANALYSIS

This chapter examines the existing system with the aim of identifying its short comings and designing a new system which will enhance its performance and efficiency. Each of the steps involved in the investigations and analysis stipulate the necessary requirements for actualizing this purpose.

3.1.1 Analysis of the existing system

The existing system of teaching and learning had shown certain weakness which could be overcome through the use of ICT and e-learning procedure. For all the purpose of accomplishing this, there is the need for determining the feasibility of the proposed e-learning products in terms of the technical, operational, time and economic feasibilities.

The technical feasibility would be considered based on two perspectives the developer(s) and the user(s) technical feasibilities. This is the fact that both the developer(s) and the user(s) require certain basic technology or electronic devices. The developer(s) technical feasibility required for the accomplishment of this project is as follows: A set of high processing speed computer with enough memory space and DVD/CD ROM.

It's well known that faculty of science and technology in the school used as case study has numerous students; one of the main problems concerning the scheduling of examination can be traced to manual handling of the system. This allows for manipulation, inaccuracy and redundancy. When this process is done manually, it becomes cumbersome and boring. The main solution to this problem is to design new system for more effective, accurate, efficient and timely production of output. The areas covered during the study were examination scheduling, seating arrangement, available invigilators, and number of students in each hall.

3.1.2 The proposed system

The proposed system was developed to solve the timetabling problem being faced by school of science and technology, Moshood Abiola University, Abeokata every academic year and reduce high cost and slow turnaround involved in the generation of optimal timetables and scheduling.

The system has capabilities for input of the various courses, halls, departments, invigilators, time slot, lecturers and the specification of a few constraints from which the timetable is constructed. The proposed timetabling system for this project seeks to generate optimal timetables using the principles of genetic algorithm (selection and crossover).

3.1.3 Functional requirements

The timetable generation process by the education center staff is:

- Unlike the manual timetabling system, the system offers flexibility.
- It utilizes minimal processing/computing power.
- It greatly reduces the time needed to generate near-optimal timetables.
- It provides an easy means for data entry and revision through an intuitive interface.

It increases productivity.

DATABASE DESIGN

The database is designed to store information for the system. The main tables in the system are Customers, Site owners, video, and customer upload video. Tables 3.3, 3.4, 3.5 and 3.6 give a description of the Customer Information, Site owner Information, customer upload video Information and site owner upload video information respectively.

Table 3.3: Customer Information

Field Name	Data Type	Size
Username	Varchar	20 characters
Fname	Varchar	20 characters
Lname	Varchar	20 characters
Sex	Int	1 character
DOB	Int	10 characters
Email	Varchar	40 characters
Password	Varchar	20 characters
Phone No	Int	20 characters
Address	Varchar	200 characters
City	Varchar	20 characters
State	Varchar	20 characters
Country	Varchar	20 characters

Cart	Text	40 characters
Account	Float	-

Table 3.4: customers upload video Information

Field Name	**Data Type**	**Size**
Username	Varchar	20 characters
Email	Varchar	40 characters
Password	Varchar	20 characters
Video file	Text	20 characters
File type	Varchar	15 characters

3.2 SYSTEM ARCHITECTURE AND FLOWCHART

System flowchart describes the data flow for a data processing system hence providing a logical diagram of how the system operates. It also shows the relationship between inputs, processing and outputs. The flowing are features of a system flowcharts.

1. The sources from which data is generated and device used for this purpose.
2. Various processing step involved in the system.
3. Intermediate and final output prepared and the devices used for their storage

.**SYSTEM ARCHITECTURE**

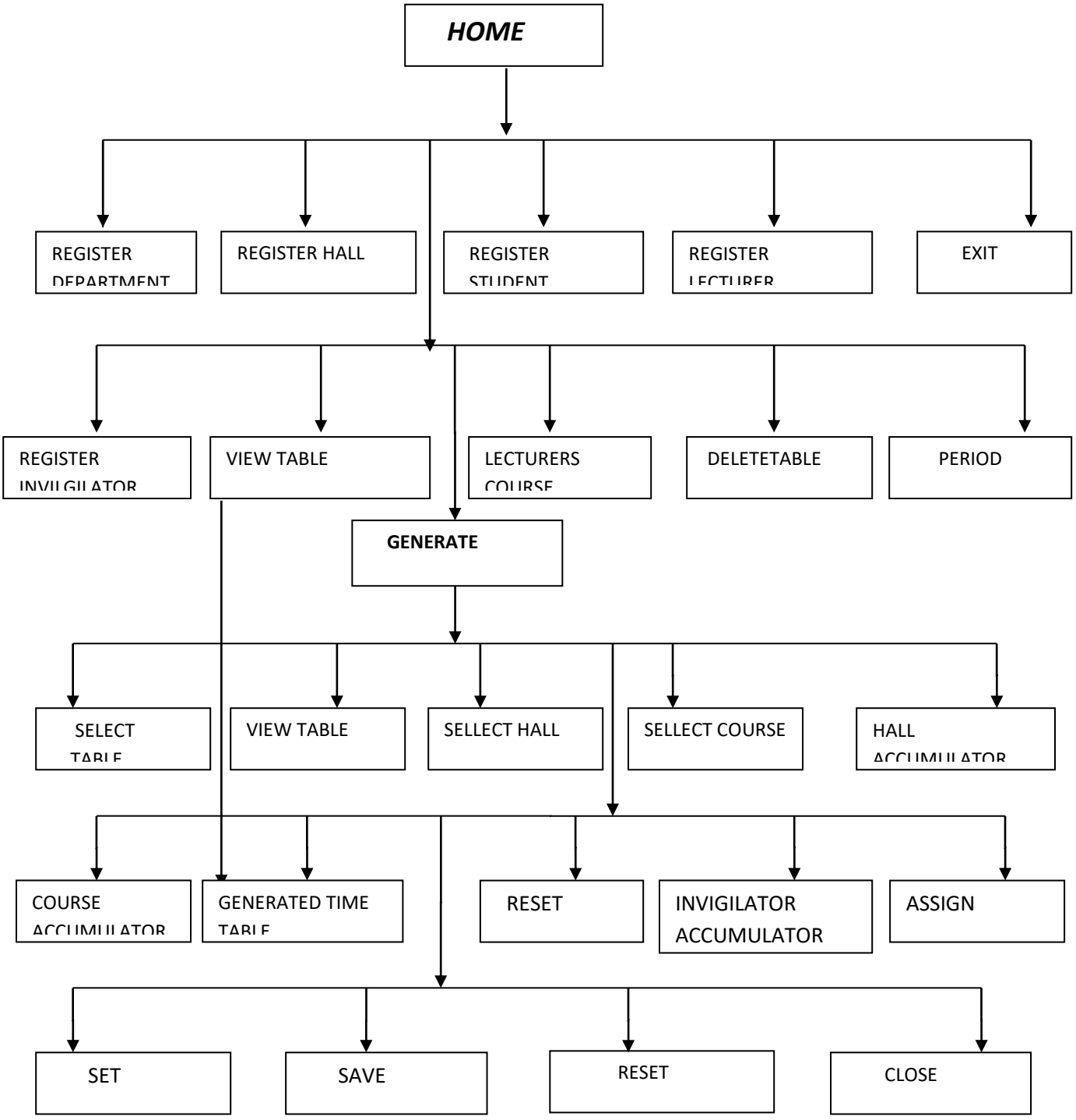
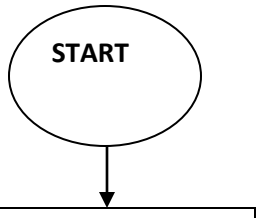


Fig. 3.1: system architecture



SYSTEM FLOWCHART

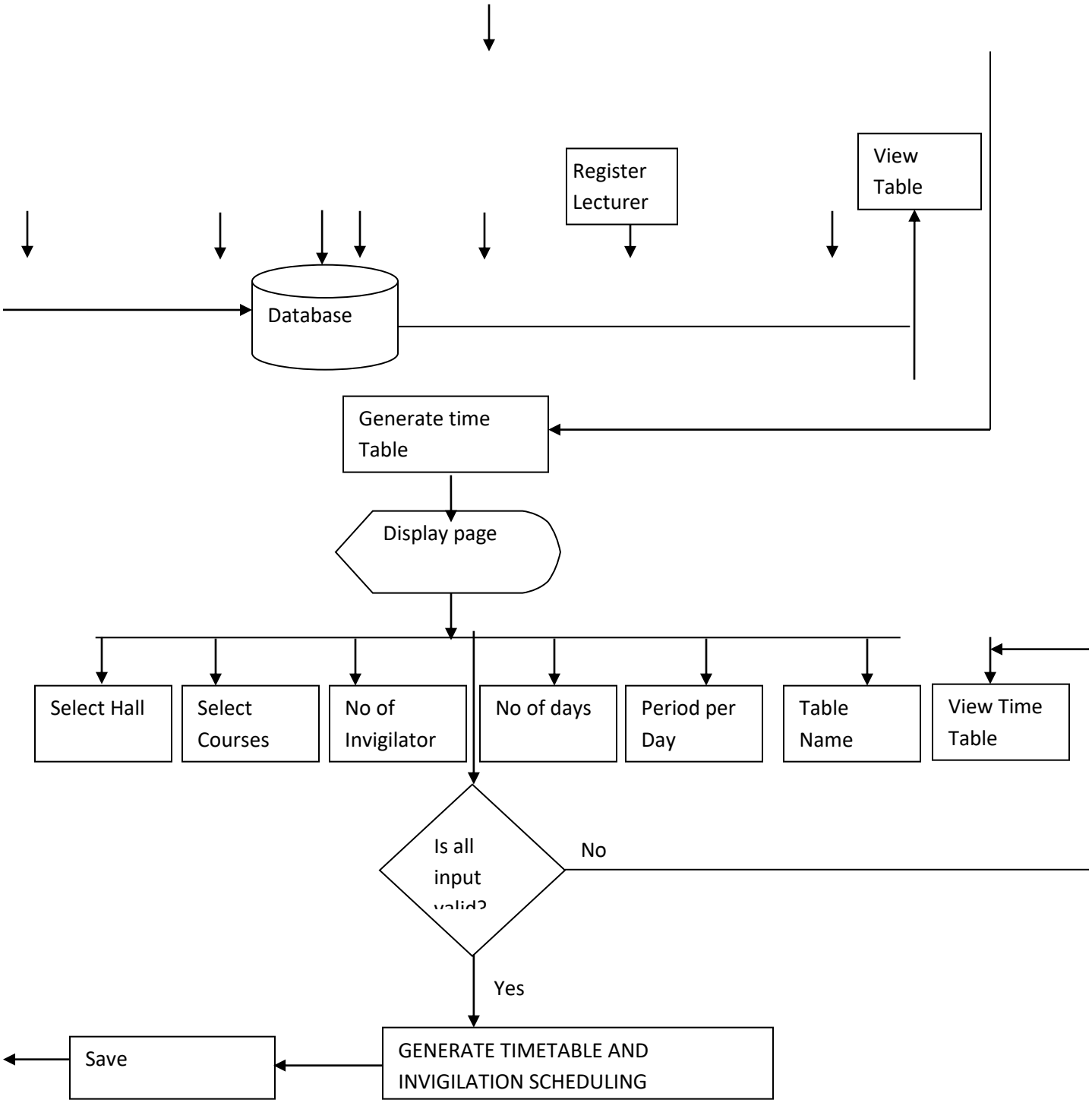


Fig. 3.2: system flowchart

3.3 SYSTEM DESIGN

3.3.1 Physical design

- **INPUT DESIGN**: identifies the type of data that is to be entered into the system in order to have the desired result.
- **OUTPUT DESIGN**: this gave a skeletal look of what the result will be. It also serves as a yardstick to which the effectiveness of the designed system can be measured.
- **DATABASE DESIGN**: this involves building an efficient file that will not support data redundancy and inconsistency, but that will have large memory to contain much information.
- **PROCESS DESIGN**: this shows the actual actions to be carried out from the input process to the output.
- **SYSTEM CONTROL DESIGN**: it involves putting a check on the kind of data that is to be accepted and processed so as to prevent computer related fraud and to ensure the accuracy of the processing and the information generated from the system.
- **TEST DESIGN**: here the test specifications are developed to ascertain the workability of the whole perfect.

There is need to provide an efficient data storage procedure for holding and securing critical information. This data storage module is known as the data base. A database is an index computer data storage and retrieval. The database management system (DBMS) creates and packages the database. There are numerous DBMS technology available today such as; ORACLE, MS ACCESS, SQL SEVER, and so no. for this project, the DBMS used is the MS ACCESS. MS ACCESS is the database editor of the popular Msoffice tools/Application, MS Access stores data or information in tables which are structured into fields. The ms access database file for the project is user INEC.mbd. It holds information on the administrator user's password.

3.3.2 Logical design

System block diagram

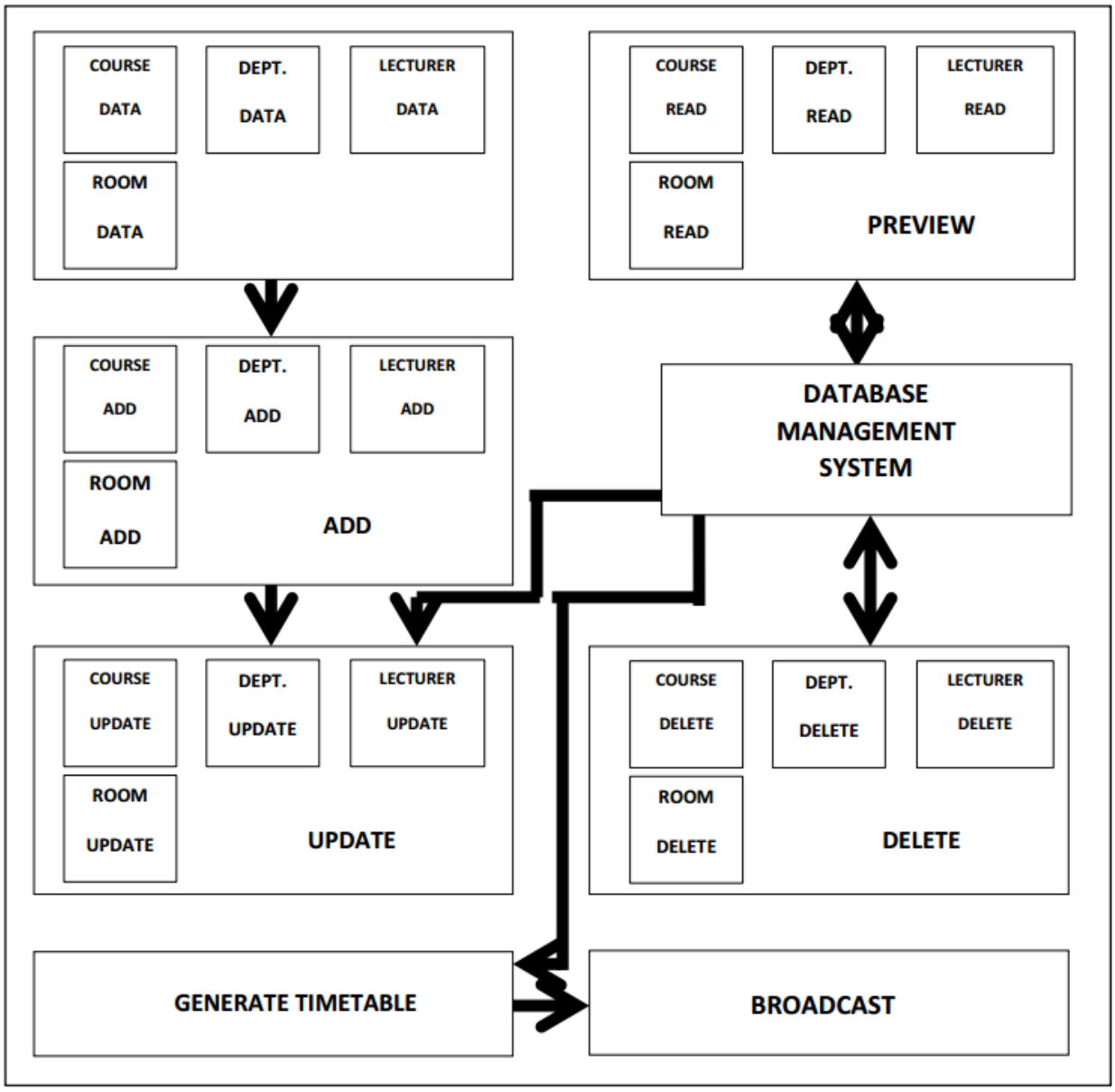


Fig. 3.3: System Block Diagram

Modeling the system

Modeling a system is the process of abstracting and organizing significant features of how the system would look like. Modeling is the designing of the software application before coding. Unified Modeling Language (UML) tools were used in modeling the system.

UML (Unified Modeling Language)

This is the object-oriented system notation that provides a set of modeling conventions that is used to specify or describe a software system in terms of objects. The UML has become an object modeling standard and adds a variety of techniques to the field of system analysis and development hence its choice for this project. UML offers ten different diagrams to model a system. These diagrams are listed below:

- Use case diagram
- Class diagram
- Object diagram
- Sequence diagram
- Collaboration diagram
- State diagram
- Activity diagram
- Component diagram
- Deployment diagram
- Package Diagram

In this project, the Use case diagram, Class diagram and Sequence diagram was be used for system modeling.

Use case diagram

Use case diagrams describe what a system does from an external observer's standpoint. The emphasis of use case diagrams is on what a system does rather than how. They are used to show the interactions between users of the system and the system. A use case represents the several users called actors and the different ways in which they interact with the system.

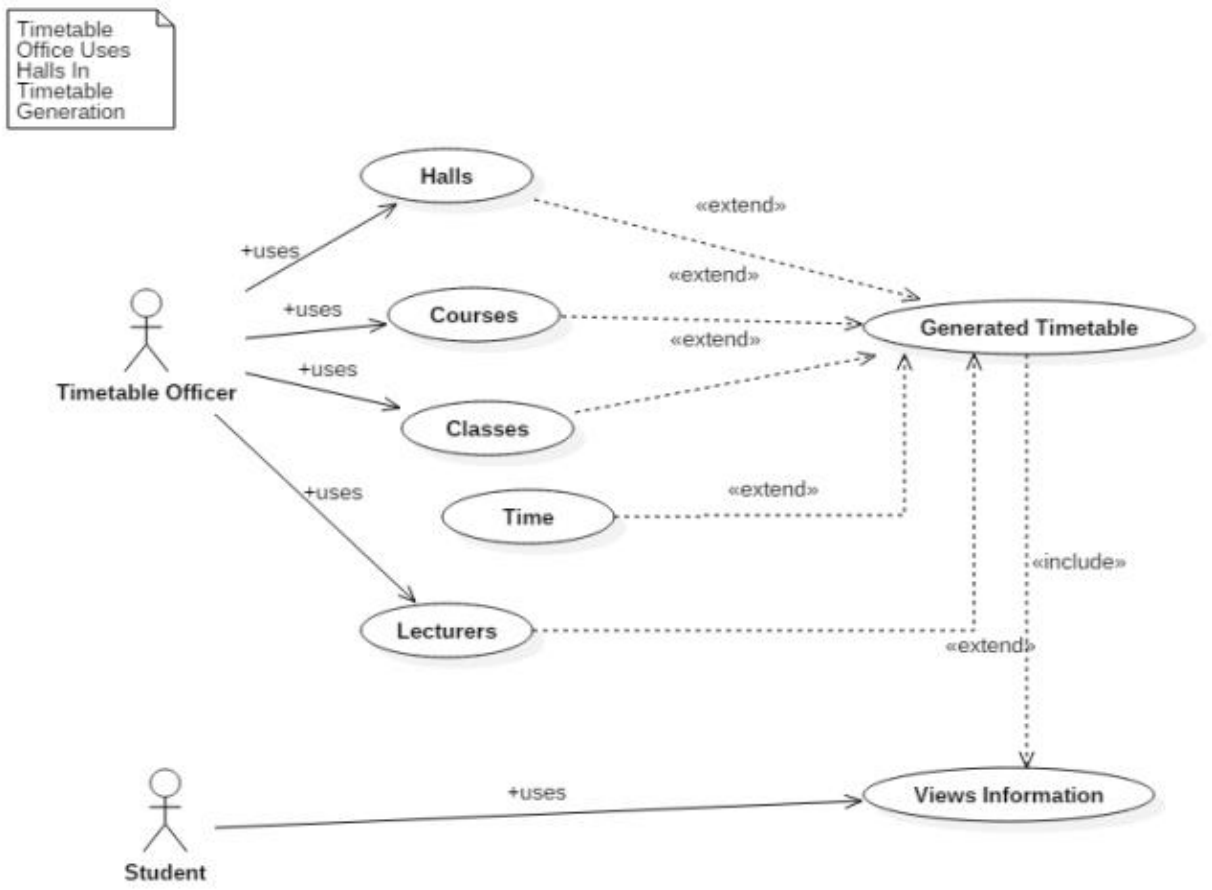


Fig. 3.4: Use Case Diagram

Class diagram

This is an organization of related objects. It gives an overview of a system by showing its classes and their relationships. Class diagrams only displays what interacts and not what happens during the interaction.

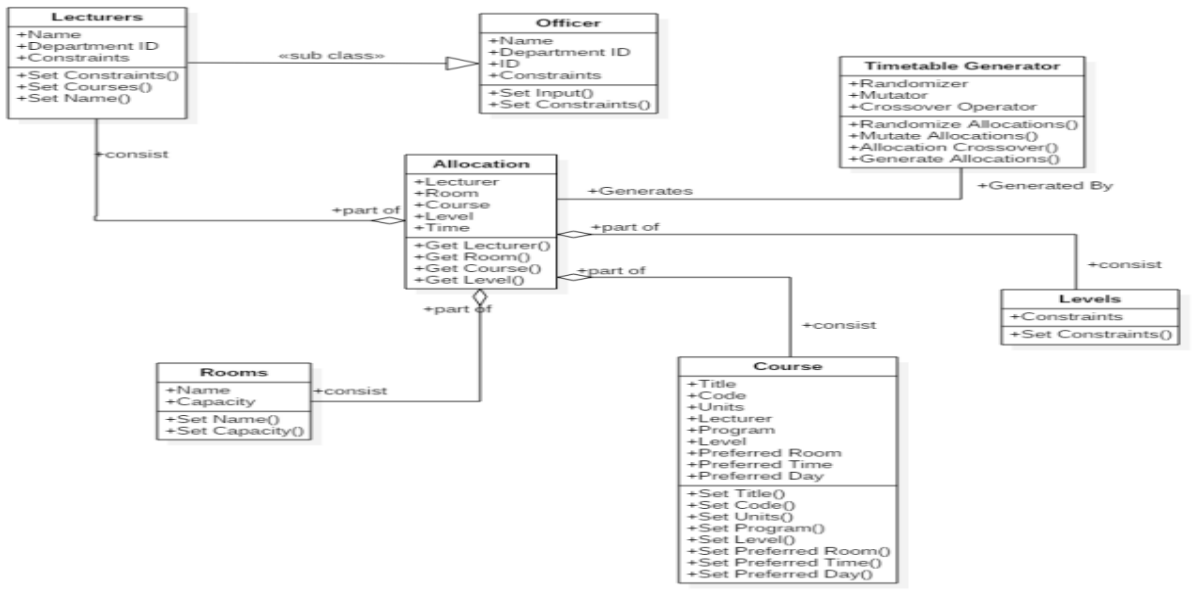


Fig. 3.5: Class Diagram

Sequence diagram

This describes how objects interact with each other through messages during the execution of a use case or any operation. They illustrate how messages are sent and received between objects and the sequence of message transfer. It also describes how operations are carried out according to the time of operation.

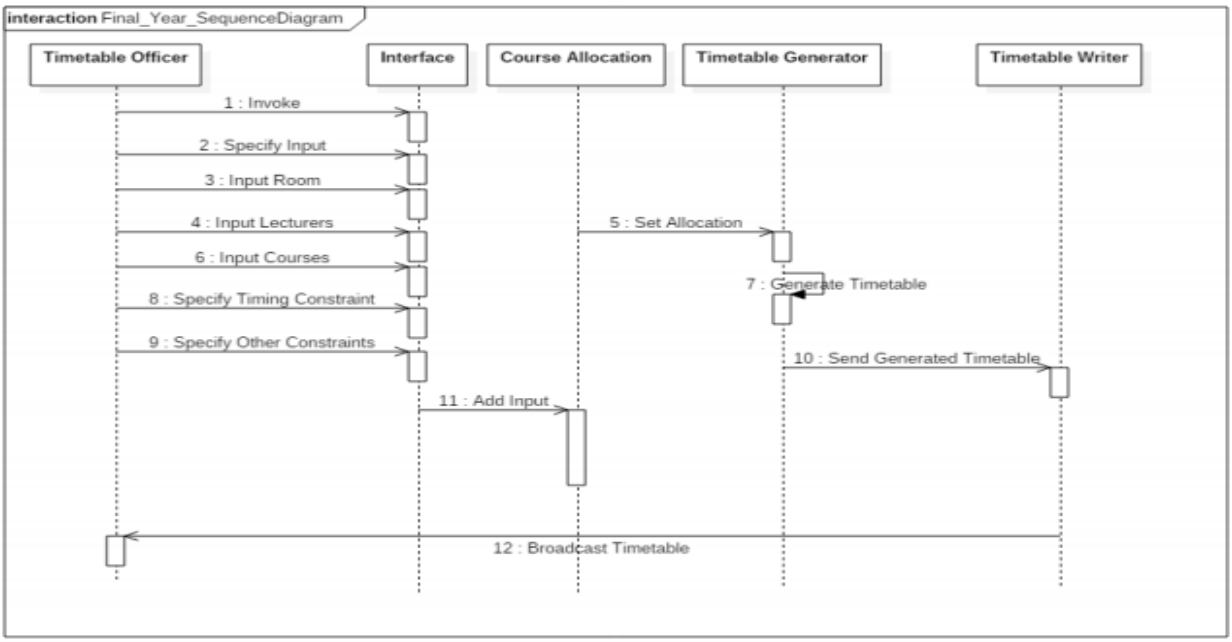


Fig. 3.6: Sequence Diagram

Updating

The program is designed in such a way that, whenever there is need for change. It can be easily done by viewing the source code of that particular item that is to be change. The program that ran the modules includes Java programming language, jasper report, and Xamp/Wanp-SQL as database engine.

3.4 SYSTEM INVESTIGATION

A system can be loosely defined as a collection of component or sub-system that are interrelated and work together to achieve a common goal. OR it is an organized unit of component that work together to achieve the same goal. Example of this is body system, computer system etc. System investigation is concerning with the indebt and thorough study of the strength and weakness of an existing system with regards to its procedures in working out an improved system. Or system investigation is defined as the fact finding which is perform mainly in the preliminary study of the proposed phase of the system decision of work . This phase may be made by distinguishing between:

1. Fact about the existing system
2. The further fact required to enable system analyst to undertake a new design of a new system. This study is carried out so as to define and establish relevant facts that were helpful in designing a new system.

3.4.1 Fact finding

The essence of this is to identify the strength and the weakness of the system as it relates to the problem stated in the statement of the problem. This involves the gathering of relevant facts as well as the techniques employed which includes; interview method, questionnaire method, observation of staff at work, work published, examination of documents.

3.4.2 Observation method

Observation were made and notation of the day-to-day sequence of operations. Information/facts where gathered by watching people, events etc. as it relate to subject matter, reliable facts are gotten which helping an unbiased analysis. The observation carried out confirmed what has been stated in statement of problem.

3.4.3 Choice of programming language

Programming language used is JAVA and XAMP-MYSQL for creating database. These were chosen because;

- It is interactive
- It has user friendly interface
- It is simple
- It is suitable for the nature of this project work
- It is the programming language the author of this work knows very well.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.1 INTRODUCTION

This chapter described the implementation of the proposed system. The output, the input and the file specifications for the proposed products differs from one another. The output specification for the web based tutorial is mainly two and they are the tutorial content and the business content. In either case, text and flash animations will be included. For the CAL, its output specification will include message box and the help animation. The e-book will make use of still images and hyperlinks. The web base tutorial will accept its input only through the feedback form. The categories for the information are others. The information gotten from this interface is not strictly used for tutorial alone but for all the proposed e-learning products. Figures 12 and 13 in the Appendix display the HTML codes of the feedback confirmation page respectively. The input specification of the CAL application software is limited to the password requirement and the input made into the "Application Assistance" input box. The password input box must correspond with the default or the change password. Similarly, the help text item entered in the Application Assistance input box must correspond with the help index of application content. There is no special input specification for the e-book.

4.2 SYSTEM REQUIREMENTS

The system requirement is divided into two, which are hardware requirements and software requirement.

4.2.1 Hardware requirements

The following are the hardware specifications for the designed application:

1. 787MHz or Higher Intel Premium Processor.
2. 512Mb Memory (RAM) or Higher.
3. VGA 800 x 600, 256 colour.
4. Uninterrupted Power Supply (optional)
5. Hard Disk Storage of 20GB Minimum.

4.2.2 Software requirements

1. Windows XP, Vista, windows 7 or 8.
2. Netbeans 5.0 Above.
3. Reliable and licensed Antivirus software like Avast.

4. Mysql database Server (Xampp or Wamp).

4.3 TESTING

The new system was tested with the required constraints and the output was a feasible timetable solution.

- **Alpha Test:** This means self or in-house test of designed application against bugs, syntax and semantics errors. It is done by the programmers during the design completion.
- **Beta Test:** This application was released to the department as a case study for testing ` the suitability of usage.
- **Audit Test:** This test is likewise to be chosen in preference to beta test depending on the organization's choice concordance with the programmer. It is achieved while the program designer establishes a scheduled training to the department administrator in study

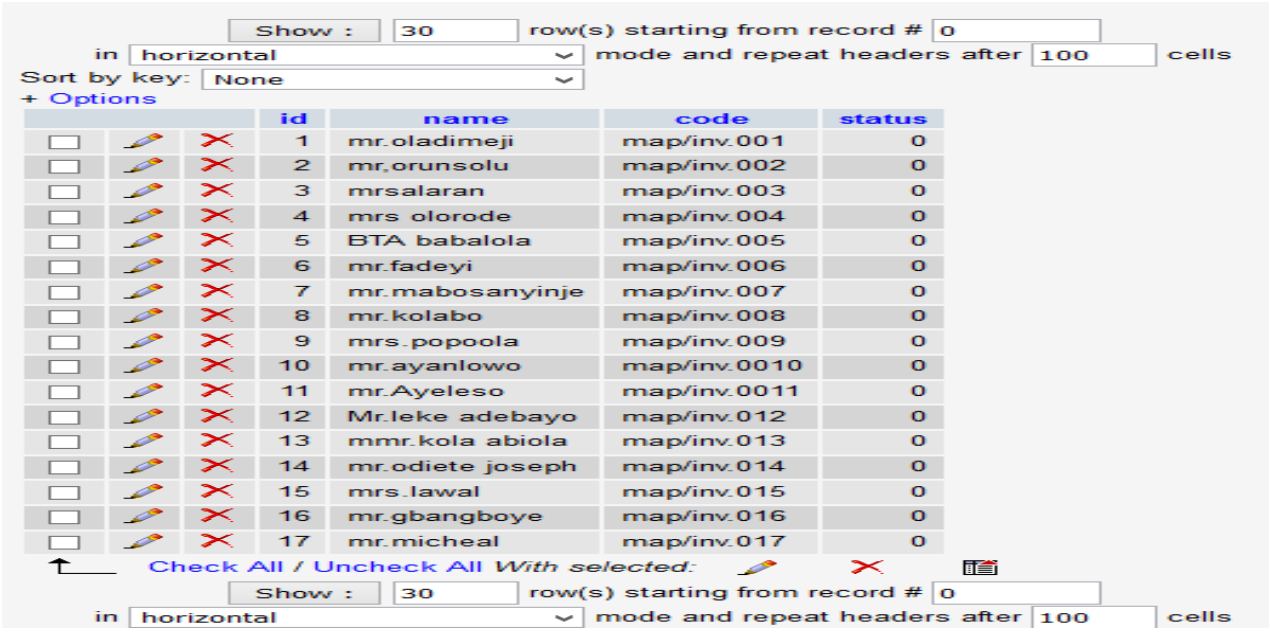
4.4 SYSTEM SNAPSHOT

Courses database: it keeps or save all information about each available courses.

			id	code	size	unit	department	level	tittle	status
☐	✎	✕	1	com 111	100	2	computer science	ND 1	intro to fotran	0
☐	✎	✕	2	com 112	100	3	computer science	ND 1	database	0
☐	✎	✕	3	com 132	100	3	SLT	ND 1	intro web	0
☐	✎	✕	4	com 133	110	3	computer science	ND 1	file management	0
☐	✎	✕	5	sta 111	100	4	computer science	ND 1	intro to stat	0
☐	✎	✕	6	sta 122	100	3	computer science	ND 1	algebra	0
☐	✎	✕	7	com 221	80	3	computer science	ND 2	java	0
☐	✎	✕	8	com 241	100	3	computer science	ND 2	intro to networking	0
☐	✎	✕	9	com 225	200	3	computer science	ND 2	system repair	0
☐	✎	✕	10	mth 212	200	3	computer science	ND 2	calculua	0
☐	✎	✕	11	sta 231	200	3	computer science	ND 2	logic and linear algebral	0
☐	✎	✕	12	com 322	120	3	computer science	HND 1	c#	0
☐	✎	✕	13	STA 324	120	3	computer science	HND 1	operation research1	0
☐	✎	✕	14	com 346	120	3	computer science	HND 1	system programing	0
☐	✎	✕	15	com 352	120	2	computer science	HND 1	pacal language	0
☐	✎	✕	16	com 344	120	3	computer science	HND 1	database 2	0
☐	✎	✕	17	slt 114	150	3	SLT	ND 1	intro to biology	0
☐	✎	✕	18	slt 121	150	3	SLT	ND 1	intro chemistry	0
☐	✎	✕	19	slt 113	150	2	SLT	ND 1	intro to physics	0

Fig. 4.1: course database.

Invigilator database: keeps registered invigilators names.



Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells
Sort by key: None
+ Options

	id	name	code	status
	1	mr.oladimeji	map/inv.001	0
	2	mr.orunsolu	map/inv.002	0
	3	mrsalaran	map/inv.003	0
	4	mrs olorode	map/inv.004	0
	5	BTA babalola	map/inv.005	0
	6	mr.fadeyi	map/inv.006	0
	7	mr.mabosanyinje	map/inv.007	0
	8	mr.kolabo	map/inv.008	0
	9	mrs.popoola	map/inv.009	0
	10	mr.ayanlowo	map/inv.0010	0
	11	mr.Ayeleso	map/inv.0011	0
	12	Mr.leke adebayo	map/inv.012	0
	13	mmr.kola abiola	map/inv.013	0
	14	mr.odiete joseph	map/inv.014	0
	15	mrs.lawal	map/inv.015	0
	16	mr.gbangboye	map/inv.016	0
	17	mr.micheal	map/inv.017	0

Check All / Uncheck All *With selected:*

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Fig. 4.2: Invigilators database.

Time table database: keeps the whole time table for future reference or retriever.

	id	time_table_id	period	day	data	period_data	status
	160	1	4	10	com 344: computer science HND 1 food 121: foodtec...	com 344: computer science HND 1 120 food 121: foo...	0
	159	1	3	10	slt 212: SLT ND 2 pham 222: pharmtech ND 2 mst 2...	slt 212: SLT ND 2 150 pham 222: pharmtech ND 2 90...	0
	157	1	1	10			0
	158	1	2	10	com 111: computer science ND 1 mst 111: math and ...	com 111: computer science ND 1 100 mst 111: math ...	0
	155	1	3	9	pham 311: pharmtech ND 2	pham 311: pharmtech ND 2 70 HALL NAME: 800 seate...	0
	156	1	4	9	com 133: computer science ND 1 mst 134: math and ...	com 133: computer science ND 1 110 mst 134: math ...	0
	152	1	4	8	slt 114: SLT ND 1	slt 114: SLT ND 1 150 HALL NAME: OGD hall 1500 n...	0
	153	1	1	9			0
	154	1	2	9	mst 412: math and stat HND 1	mst 412: math and stat HND 1 95 HALL NAME: keshi...	0

Fig. 4.3: timetable database.

Home page: This is the home page where administrator can view and navigate to all sections and the actions that can be perform on each sections.

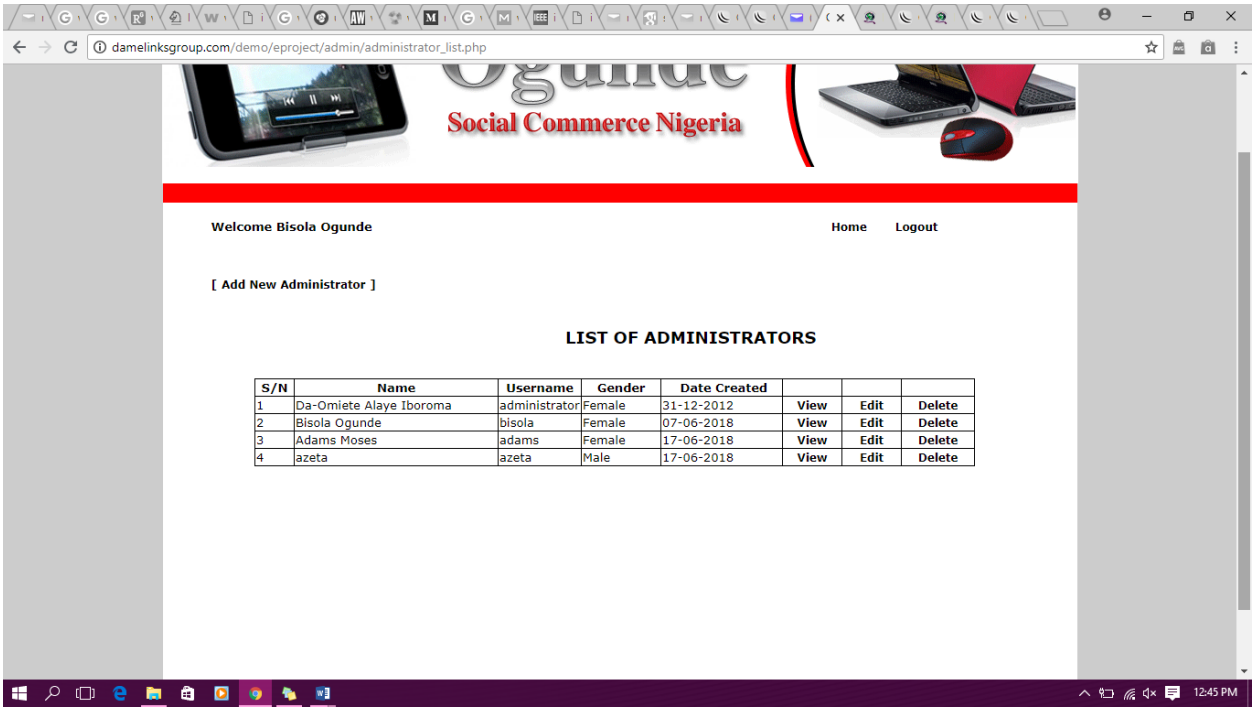


Fig. 4.4: Home page.

Hall registration section: This is a section where administrator can register available halls and as well update registered hall.

Fig. 4.5: hall registration section.

Time table accumulators sections: This section allow administrator to select from database the saved time tables

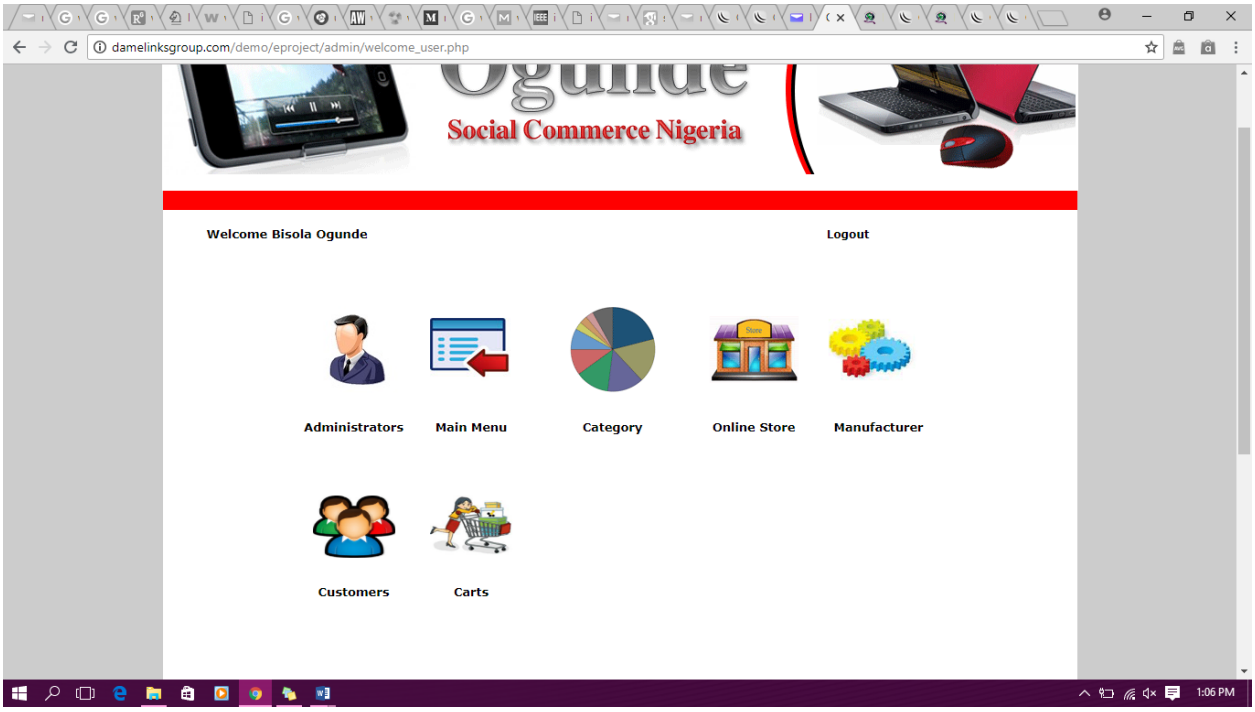


Fig. 4.6: Tables Accumulators section.

Lecturer's section: This is a section where administrator can assign courses to a particular lecturer.

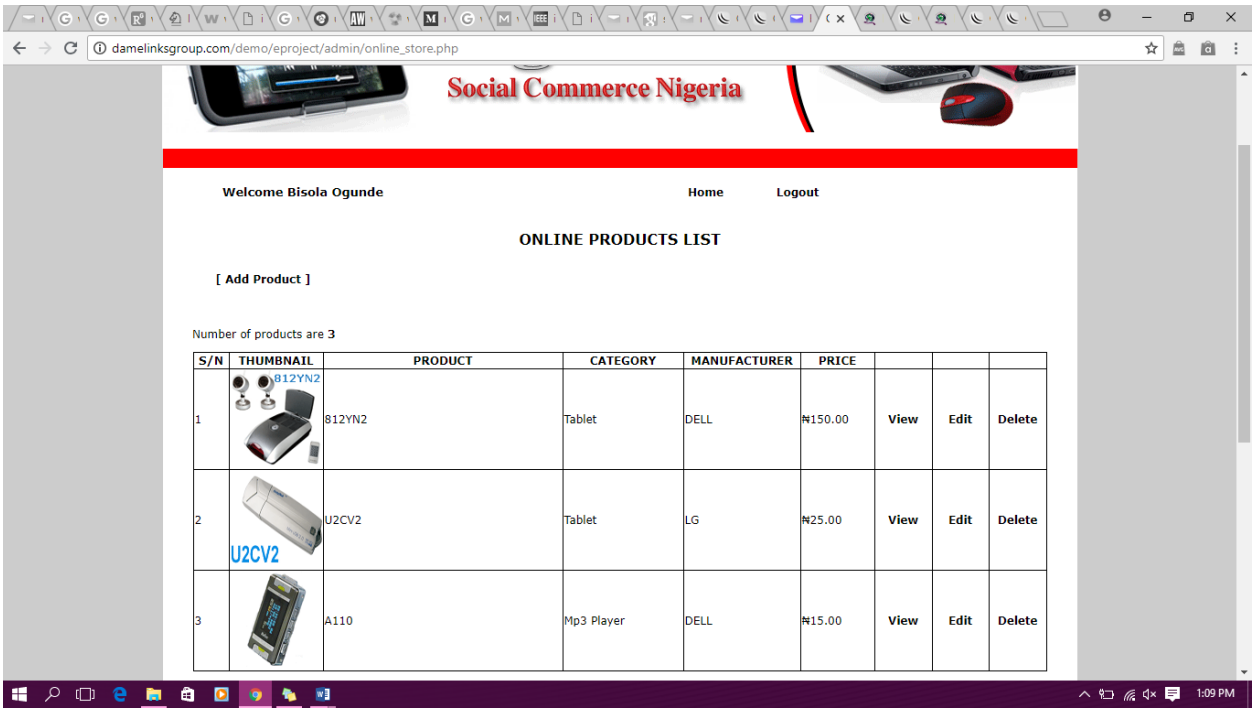


Fig. 4.7: lecturer's section.

Departmental section: This is a section for administrator to register and update available department.

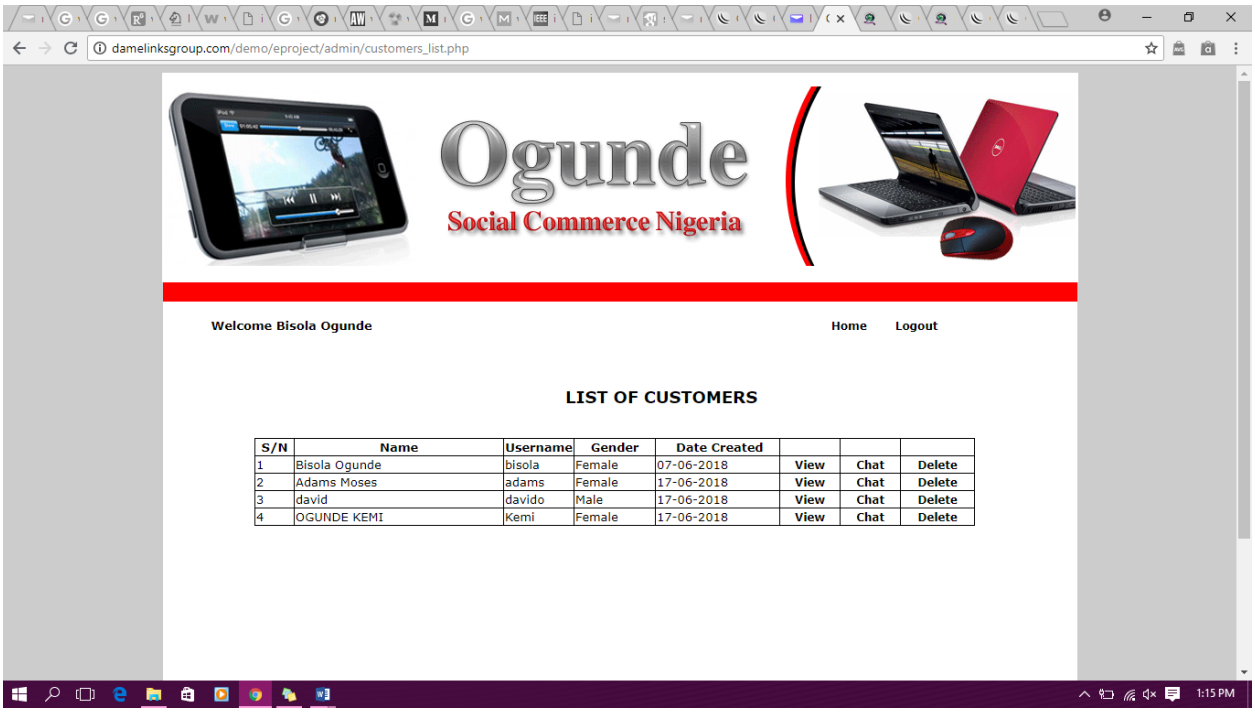


Fig. 4.8: departmental section.

Course registration section: This is a section for administrator to register and update available courses.

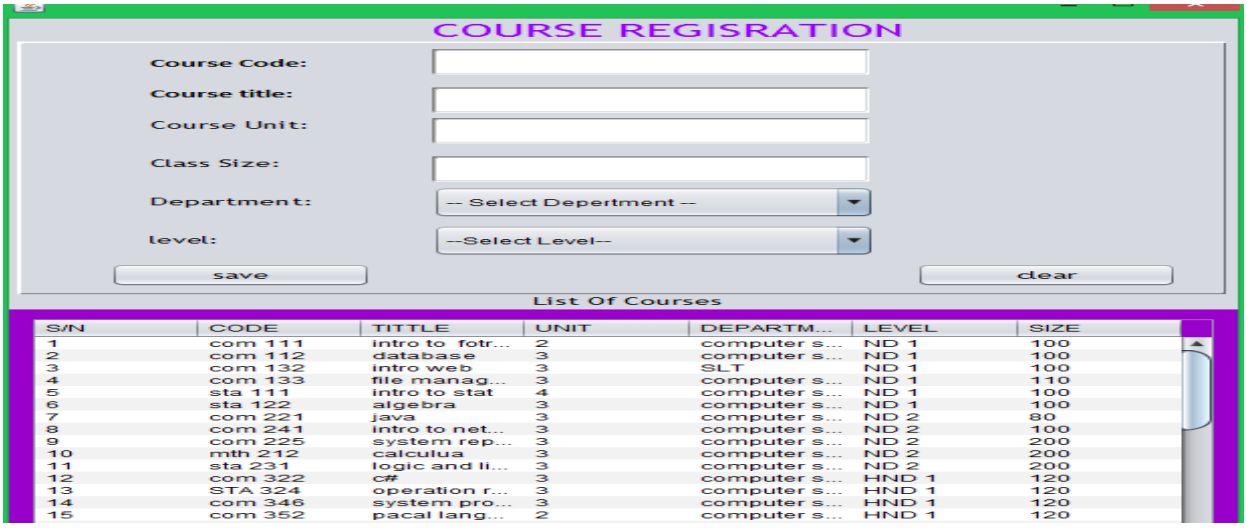


Fig. 4.9: course registration section.

lecturer's registration section: This is a section for administrator to register and update available lecturers.

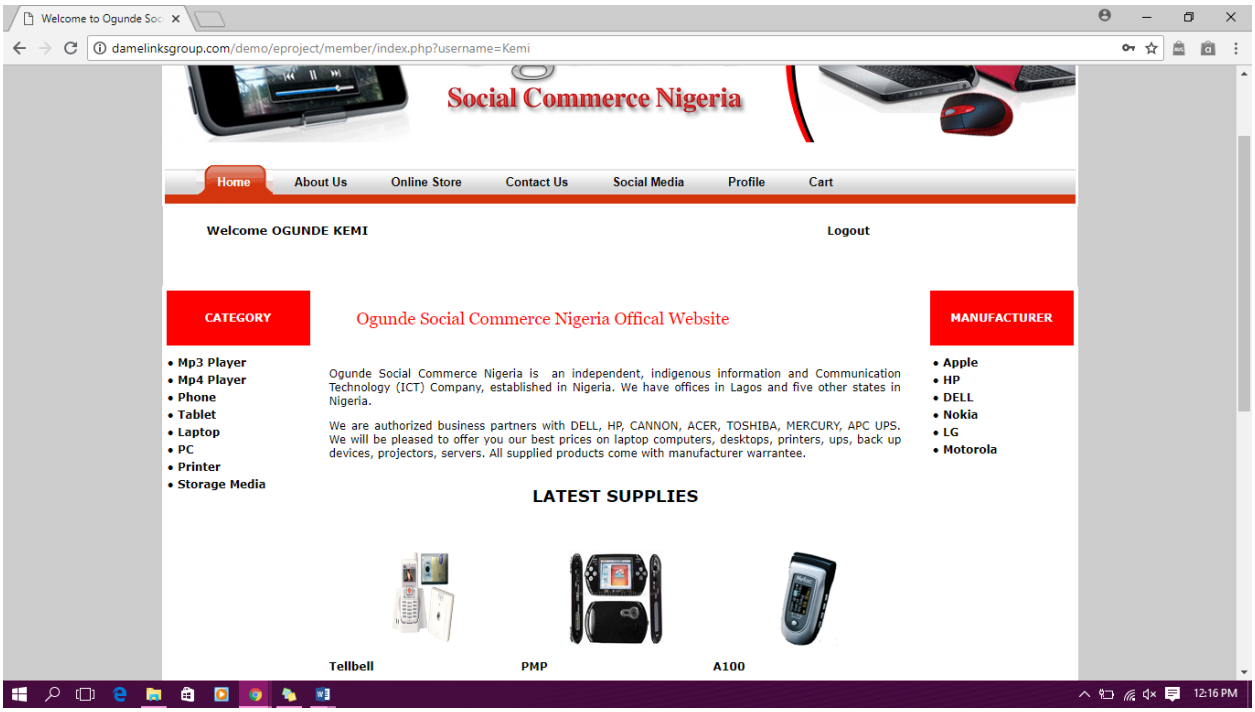


Fig. 4.10: lecturers' registration section.

Invigilators registration section: This is a section for administrator to register and update available invigilators.

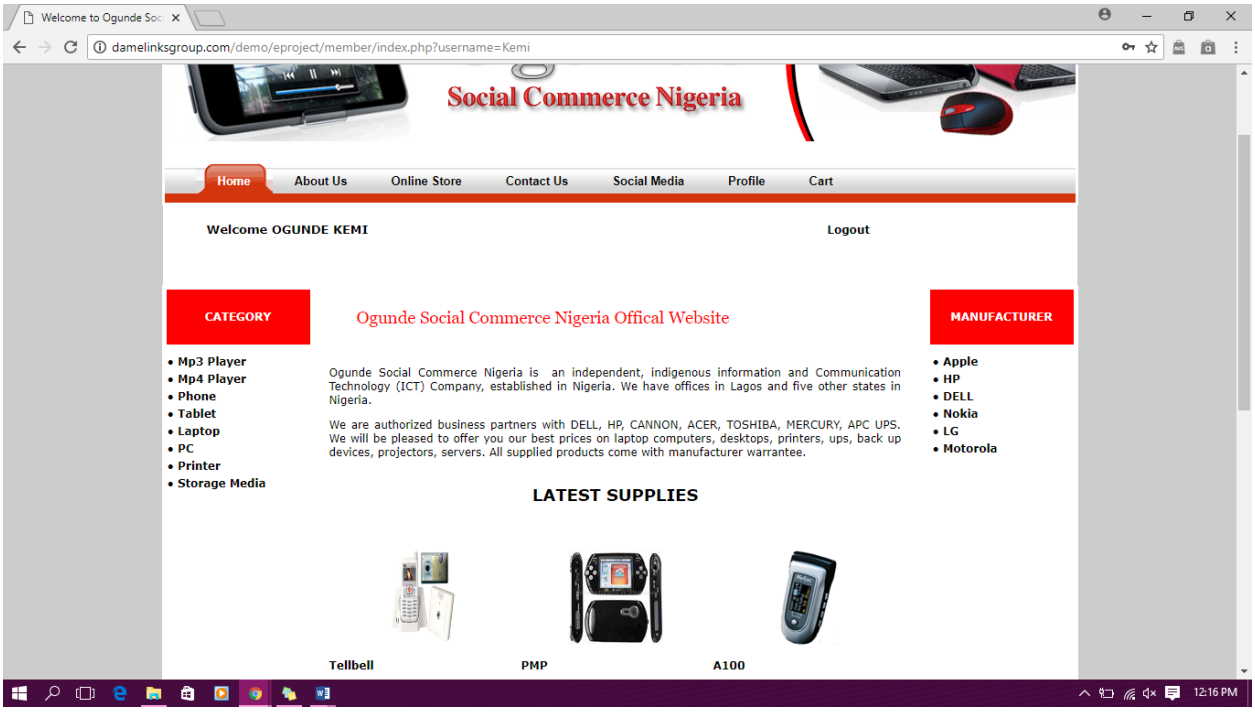


Fig. 4.11: Invigilators registration section.

Time table page: This is a wide and complex section where the actual time table was generated. Here administrator selects and fell all required field and generate the real time table.

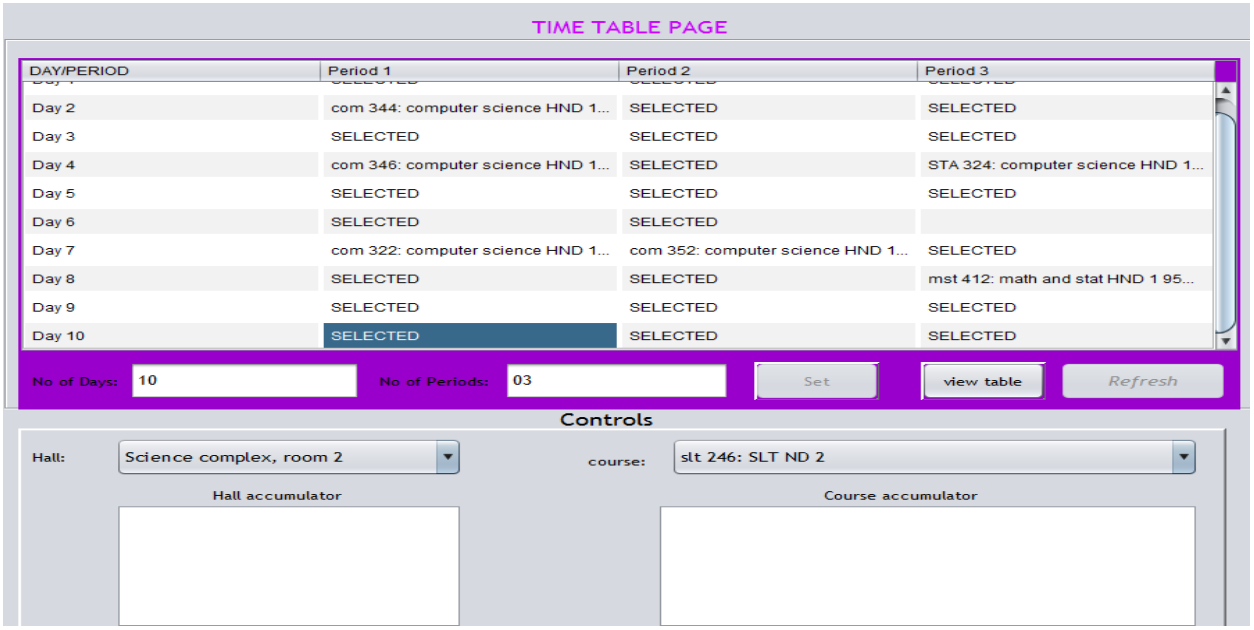
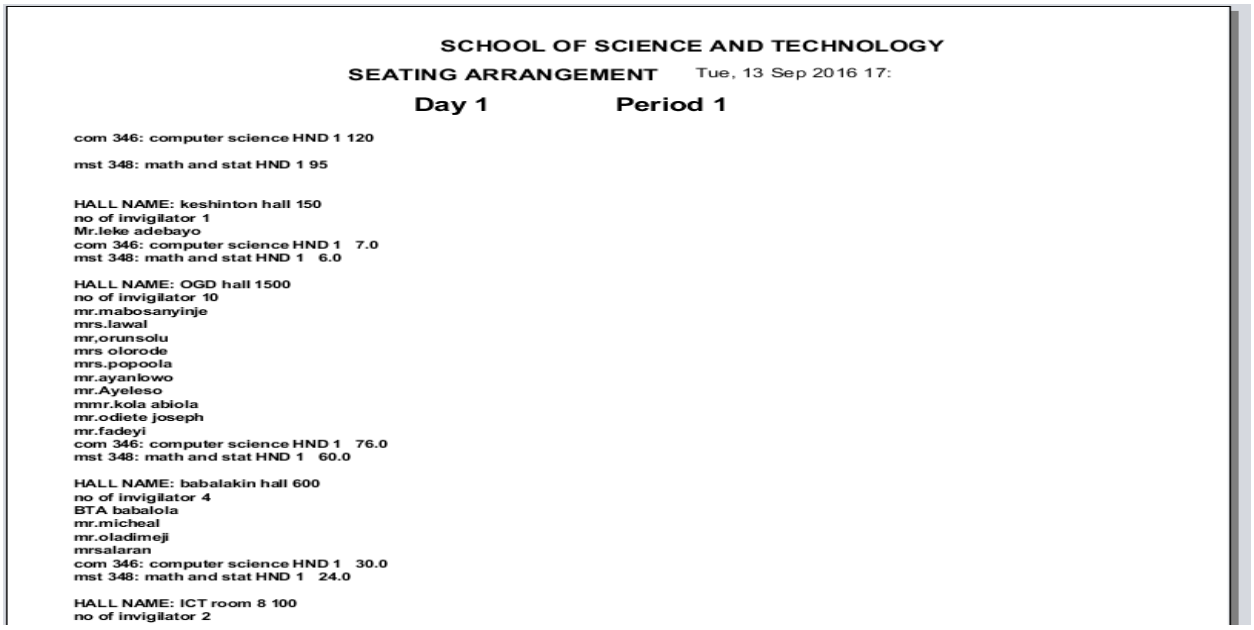


Fig. 4.12: Timetable generation page section.

Time table: This is a complete time table generated from the application.

Scheduling arrangement: This is a complete scheduling arrangement generated from the application.



SCHOOL OF SCIENCE AND TECHNOLOGY

SEATING ARRANGEMENT Tue, 13 Sep 2016 17:

Day 1 Period 1

com 346: computer science HND 1 120

mst 348: math and stat HND 1 95

HALL NAME: keshinton hall 150
no of invigilator 1
Mr.leke adebayo
com 346: computer science HND 1 7.0
mst 348: math and stat HND 1 6.0

HALL NAME: OGD hall 1500
no of invigilator 10
mr.mabosanyinje
mrs.lawal
mr,orunsolu
mrs olorode
mrs.popoola
mr.ayanlowo
mr.Ayeleso
mmr.kola abiola
mr.odiete joseph
mr.fadeyi
com 346: computer science HND 1 76.0
mst 348: math and stat HND 1 60.0

HALL NAME: babalakin hall 600
no of invigilator 4
BTA babalola
mr.micheal
mr.oladimeji
mrsalaran
com 346: computer science HND 1 30.0
mst 348: math and stat HND 1 24.0

HALL NAME: ICT room 8 100
no of invigilator 2

Fig. 4.13: Generated seating arrangement.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 SUMMARY

This chapter reports the evaluation result of FARME-D approach in enhancing the quality of Fuzzy expert system. This evaluation is mainly directed towards the fuzzy expert system knowledge-base being the back bone of the system. The chapter presents the evaluation result of quantitative measure of accuracy and comprehensibility over fuzzy expert system with FARME-D approach (FARMES) as against fuzzy expert system with standard rule-base formulation. It also gives the report of statistical analysis of the test cases result.

5.2 CONCLUSION

For the purpose of evaluation as reported in chapter four, test case approach was used to verify the accuracy of the new approach. The test cases consist of 20 non-smoking men record outside the mining dataset to determine the completeness of the knowledge-base. The quantitative measure of comprehensiveness is used to determine the compactness of fuzzy-mining expert system.

The accuracy measure is used to determine the probability that the system can correctly make a decision. Also, t-test was carried out to determine the significant difference between FES with 79 rules (FARMES) and ATP III result, FES with 108 rules and the ATP III result. ANOVA test was also carried out to determine if there exists a significant differencebetween the three alternative results. All are reported in this section.

This chapter reports the evaluation result of FARME-D approach in enhancing the quality of Fuzzy expert system. This evaluation is mainly directed towards the fuzzy expert system knowledge-base being the back bone of the system. The chapter presents the evaluation result of quantitative measure of accuracy and comprehensibility over fuzzy expert system.Result of quantitative measure of accuracy and comprehensibility over fuzzy expert system.

5.3 RECOMMENDATIONS

In this research objective evaluation is used to assay the performance of the IIMA system. This comparism is directed towards the execution time and extracted rules in order to reveal the performance of both system depending on the number of keywordsets. It is also aimed at examining which of the system (existing integrated mining system or IIMA) generates frequent keywordsets from the most important keywords rather than both the important and unimportant keywords. Consequently, this leads to extract interesting and uninteresting rules. The result of this evaluation helps to determine which system extracts the more interesting rules at short time.

REFERENCES

Abramson D.,&Abela J. (1992). "A parallel genetic algorithm for solving the school timetabling problem."In Proceedings of the 15th Australian Computer Science Conference, Hobart, 1-11.

Adam, M. (2004)."Genetic Algorithms and Evolutionary Computation ".Available online at http://www.talkorigins.org/faqs/genalg/genalg.html.

Al-Attar, A.(1994). White Paper: "A hybrid GA-heuristic search strategy." AI Expert, USA.

Alberto, C., Marco D., Vittorio, M. (1992)."A Genetic Algorithm to Solve the Timetable Problem" Journal of Computational Optimization and Applications, 1, 90-92.

Burke E, Elliman D and Weare R (1994)."A genetic algorithm for university timetabling system."Presented at the East-West Conference on Computer Technologies in Education, Crimea, Ukraine.

Carrasco M.P.,&Pato M.V. (2001). "A Multi-objective genetic algorithm for the class/teacher timetabling problem." In Proceedings of the Practice and Theory of Automated Timetabling (PATAT'00), Lecture Notes in Computer Science, Springer, 2079, 3-17.

Chan, H. W. (1997) "School Timetabling Using Genetic Search."2th International Conference on the Practice and Theory of Automated Timetabling, PATAT'97.

Costa, D (1994) "A tabu search algorithm for computing an operational timetable." European Journal of Operational Research, 76(1), 98-110.

Datta, D., Deb K., & Fonseca, C.M.(2006).Multi-objective evolutionary algorithm for university class timetabling problem, In Evolutionary Scheduling, Springer-Verlag Press.

DeGans O.B.(1981) "A computer timetabling system for secondary schools in the Netherlands". European Journal of Operations Research,7, 175-182.

Eley, M. (2006)."Ant Algorithms for the Exam Timetabling Problem."6th International Conference on the Practice and Theory of Automated Timetabling, PATAT'06.

Fang, H. L. (1994). "Genetic Algorithms in Timetabling Problems."PhD Thesis, University of Edinburgh.

Fernandes, C. (2002). "Infected Genes Evolutionary Algorithm for School Timetabling." WSES International Conference.

Gianoglio, P. (1990)."Application of neural networks to timetable construction."Presented at the 3rd International Workshop on Neural Network and Their Applications, Nimes, France.

Gotlieb, C .C. (1962)."The construction of class-teacher timetables."Proceedings of IFIP Congress, North-Holland Pub. Co., Amsterdam, 73-77.

Gröbner, M., Wilke, P (2002)."A General View on Timetabling Problems." 4th International Conference on the Practice and Theory of Automated Timetabling - PATAT'02.

Lajos, G. (1995). "Complete university modular timetabling using constraint logic programming." Presented at the Ist International Conference on the Practice and Theory of Automated Timetabling, Scotland, UK.

Paechter B, Cumming A, Luchian H and Petriuc M (1994)."Two solutions to the general timetable problem using evolutionary methods."Presented at the 3rd World Conference on Evolutionary Computing. Florida, USA.

PATAT (1995). The International Series of Conferences on the Practice and Theory of Automated Timetabling (PATAT) - http://www.asap.cs.nott.ac.uk/patat/patat-index.shtml

S. B. Deris, S. Omatu, H. Ohta, P. A. B. D. Samat(1997)."University timetabling by constraint-based reasoning: A case study" Journal of the Operational Research Society, 48(12), 2-4

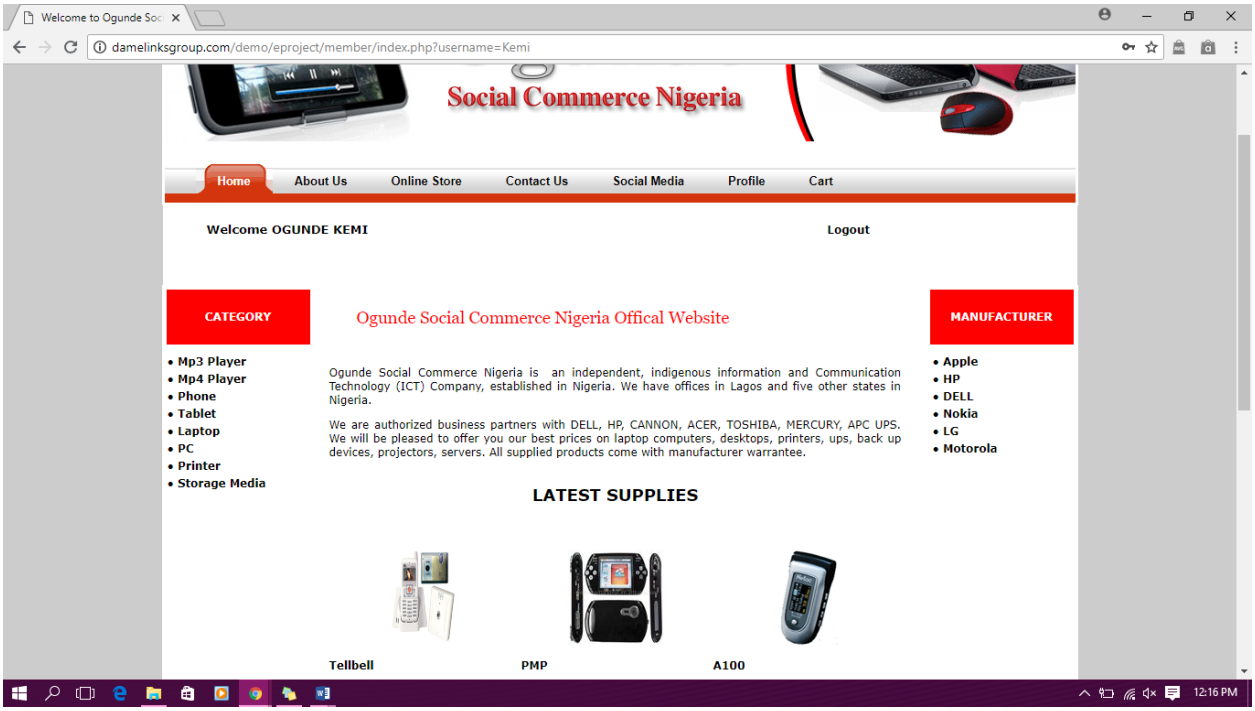
Tripathy A. (1984) "School timetabling - A case in large binary integer linear programming" Management Science, 30(12), 1473-1489

Ossam, C. (2009) "University Scheduling using Genetic Algorithm." Master Thesis - Department of Computer Engineering, Dalarna Univeristy, 3-4.

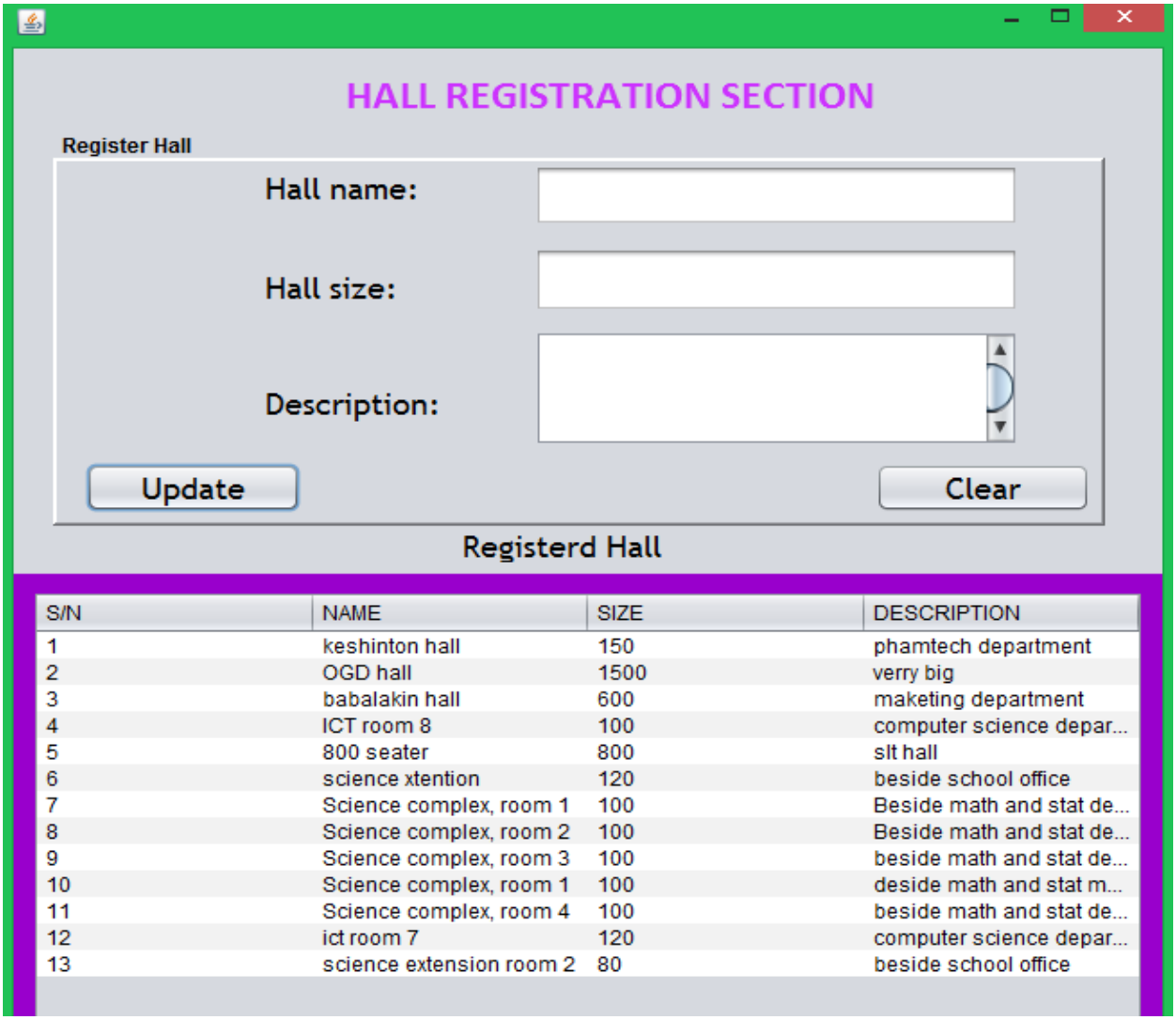
APPENDIX A

USER INTERFACE

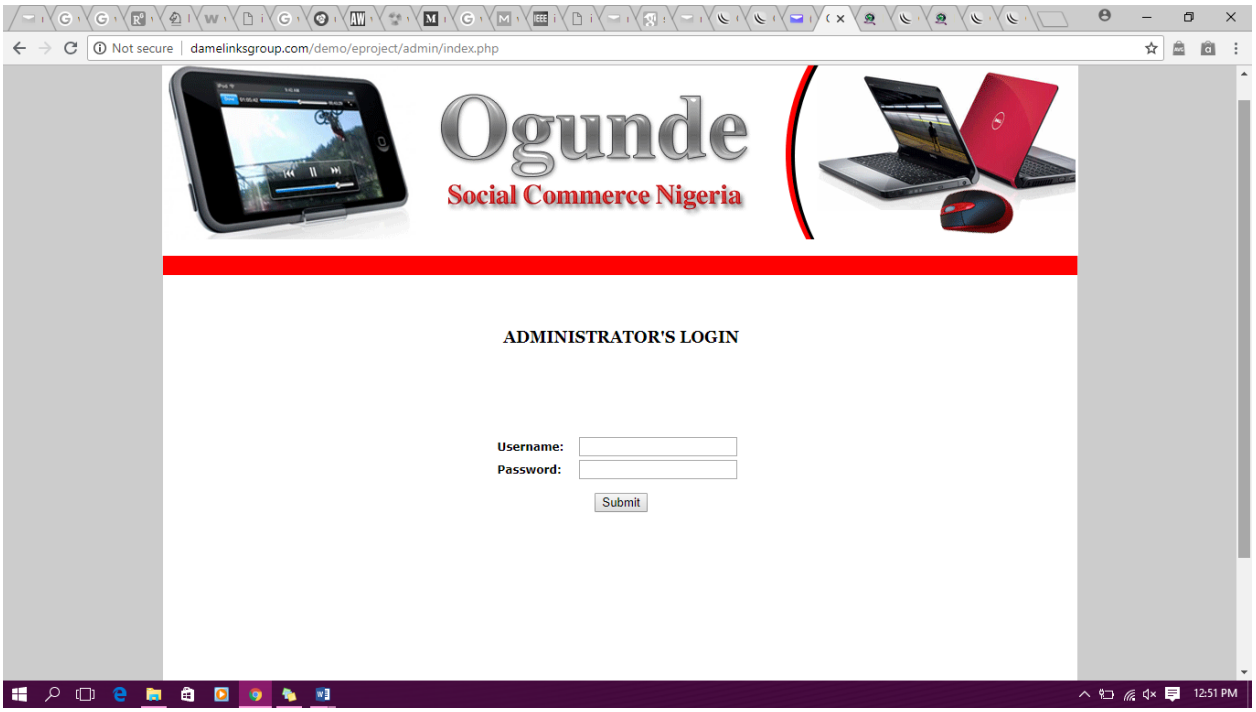
Homepage



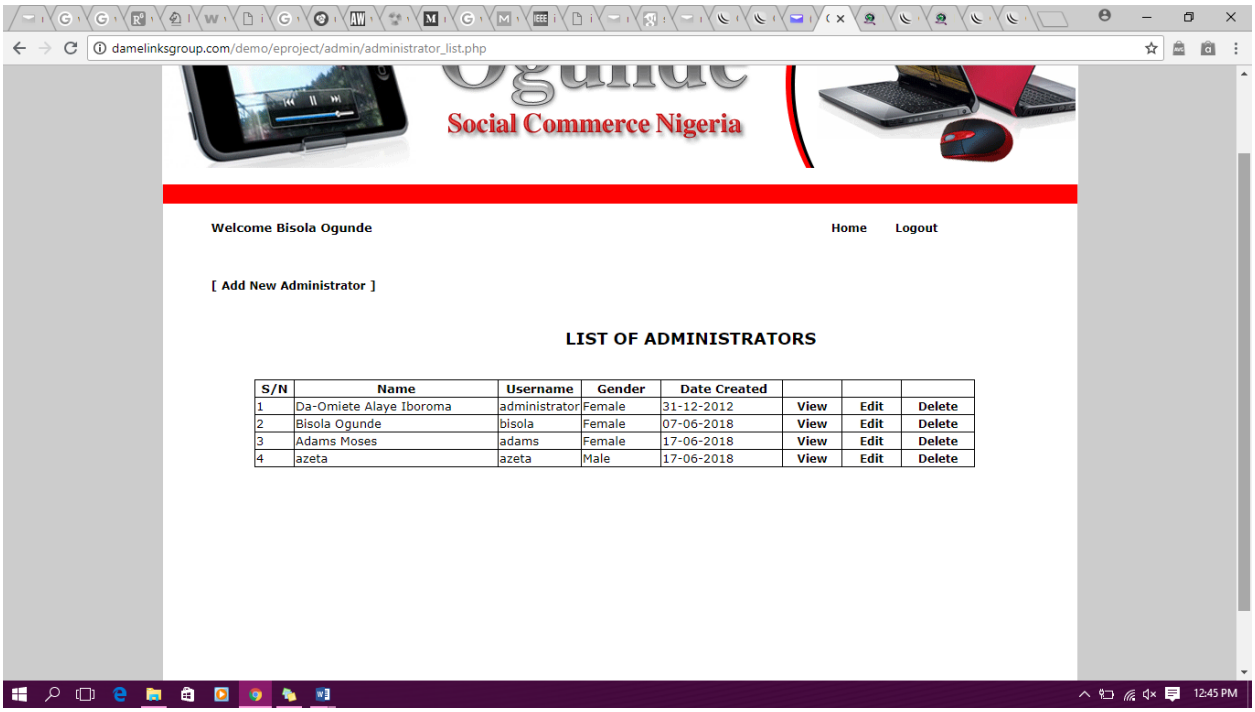
Hall registration



S/N	NAME	SIZE	DESCRIPTION
1	keshinton hall	150	phamtech department
2	OGD hall	1500	verry big
3	babalakin hall	600	maketing department
4	ICT room 8	100	computer science depar...
5	800 seater	800	slt hall
6	science xtention	120	beside school office
7	Science complex, room 1	100	Beside math and stat de...
8	Science complex, room 2	100	Beside math and stat de...
9	Science complex, room 3	100	beside math and stat de...
10	Science complex, room 1	100	deside math and stat m...
11	Science complex, room 4	100	beside math and stat de...
12	ict room 7	120	computer science depar...
13	science extension room 2	80	beside school office

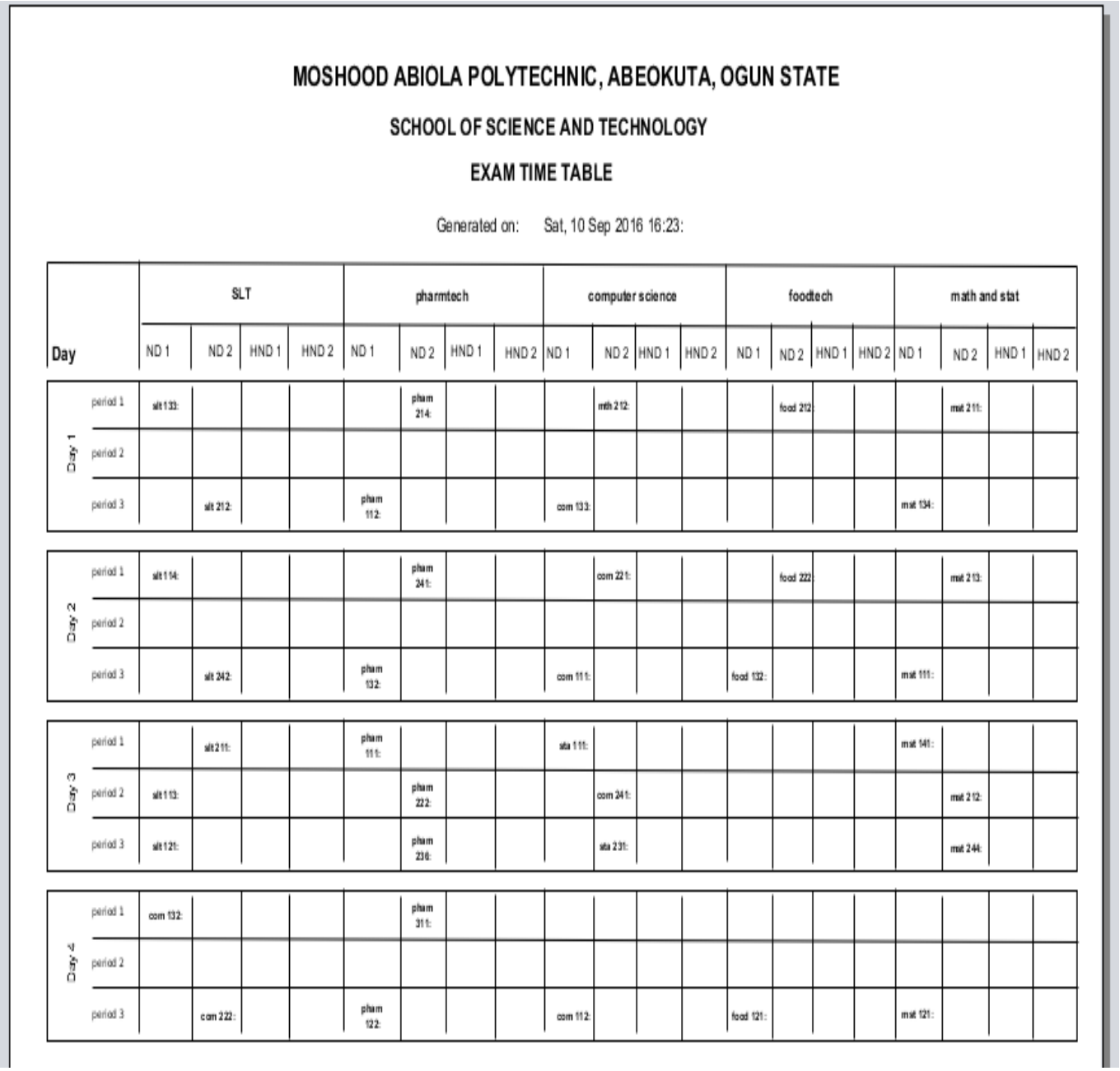


Admin Login page



Admin Registration page

Generated Exam Time Table



MOSHOOD ABIOLA POLYTECHNIC, ABEOKUTA, OGUN STATE

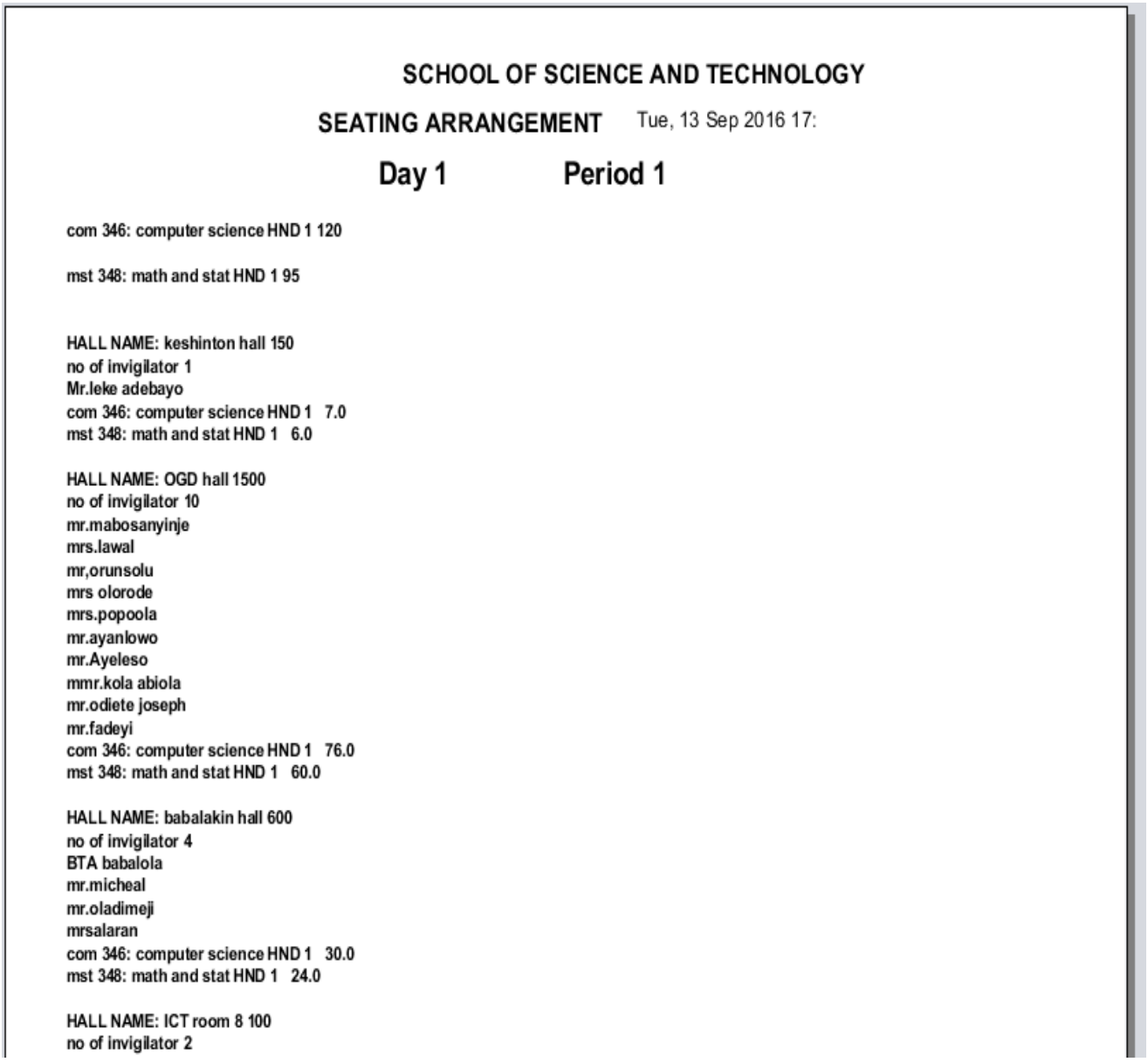
SCHOOL OF SCIENCE AND TECHNOLOGY

EXAM TIME TABLE

Generated on: Sat, 10 Sep 2016 16:23:

Day		SLT				pharmtech				computer science				foodtech				math and stat			
		ND 1	ND 2	HND 1	HND 2	ND 1	ND 2	HND 1	HND 2	ND 1	ND 2	HND 1	HND 2	ND 1	ND 2	HND 1	HND 2	ND 1	ND 2	HND 1	HND 2
Day 1	period 1	slt 133:					pharm 214:				mth 212:				food 212:				mat 211:		
	period 2																				
	period 3		slt 212:			pharm 112:				com 133:								mat 134:			
Day 2	period 1	slt 114:					pharm 241:				com 221:				food 222:				mat 213:		
	period 2																				
	period 3		slt 242:			pharm 132:				com 111:				food 132:				mat 111:			
Day 3	period 1		slt 211:			pharm 111:				sta 111:								mat 141:			
	period 2	slt 113:					pharm 222:				com 241:								mat 212:		
	period 3	slt 121:					pharm 236:				sta 231:								mat 244:		
Day 4	period 1	com 132:					pharm 311:														
	period 2																				
	period 3		com 222:			pharm 122:				com 112:				food 121:				mat 121:			

Scheduling Arrangement



SCHOOL OF SCIENCE AND TECHNOLOGY

SEATING ARRANGEMENT Tue, 13 Sep 2016 17:

Day 1 Period 1

com 346: computer science HND 1 120

mst 348: math and stat HND 1 95

HALL NAME: keshinton hall 150
no of invigilator 1
Mr.leke adebayo
com 346: computer science HND 1 7.0
mst 348: math and stat HND 1 6.0

HALL NAME: OGD hall 1500
no of invigilator 10
mr.mabosanyinje
mrs.lawal
mr,orunsolu
mrs olorode
mrs.popoola
mr.ayanlowo
mr.Ayeleso
mmr.kola abiola
mr.odiete joseph
mr.fadeyi
com 346: computer science HND 1 76.0
mst 348: math and stat HND 1 60.0

HALL NAME: babalakin hall 600
no of invigilator 4
BTA babalola
mr.micheal
mr.oladimeji
mrsalaran
com 346: computer science HND 1 30.0
mst 348: math and stat HND 1 24.0

HALL NAME: ICT room 8 100
no of invigilator 2

APPENDIX B

DATABASE DESIGN INTERFACE

Courses Database

			id	code	size	unit	department	level	tittle	status
☐	✎	✖	1	com 111	100	2	computer science	ND 1	intro to fotran	0
☐	✎	✖	2	com 112	100	3	computer science	ND 1	database	0
☐	✎	✖	3	com 132	100	3	SLT	ND 1	intro web	0
☐	✎	✖	4	com 133	110	3	computer science	ND 1	file management	0
☐	✎	✖	5	sta 111	100	4	computer science	ND 1	intro to stat	0
☐	✎	✖	6	sta 122	100	3	computer science	ND 1	algebra	0
☐	✎	✖	7	com 221	80	3	computer science	ND 2	java	0
☐	✎	✖	8	com 241	100	3	computer science	ND 2	intro to networking	0
☐	✎	✖	9	com 225	200	3	computer science	ND 2	system repair	0
☐	✎	✖	10	mth 212	200	3	computer science	ND 2	calculua	0
☐	✎	✖	11	sta 231	200	3	computer science	ND 2	logic and linear algebral	0
☐	✎	✖	12	com 322	120	3	computer science	HND 1	c#	0
☐	✎	✖	13	STA 324	120	3	computer science	HND 1	operation research1	0
☐	✎	✖	14	com 346	120	3	computer science	HND 1	system programing	0
☐	✎	✖	15	com 352	120	2	computer science	HND 1	pacal language	0
☐	✎	✖	16	com 344	120	3	computer science	HND 1	database 2	0
☐	✎	✖	17	slt 114	150	3	SLT	ND 1	intro to biology	0
☐	✎	✖	18	slt 121	150	3	SLT	ND 1	intro chemistry	0
☐	✎	✖	19	slt 113	150	2	SLT	ND 1	intro to physics	0

Invigilators Database

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

			id	name	code	status
☐	✎	✕	1	mr.oladimeji	map/inv.001	0
☐	✎	✕	2	mr,orunsolu	map/inv.002	0
☐	✎	✕	3	mrsalaran	map/inv.003	0
☐	✎	✕	4	mrs olorode	map/inv.004	0
☐	✎	✕	5	BTA babalola	map/inv.005	0
☐	✎	✕	6	mr.fadeyi	map/inv.006	0
☐	✎	✕	7	mr.mabosanyinje	map/inv.007	0
☐	✎	✕	8	mr.kolabo	map/inv.008	0
☐	✎	✕	9	mrs.popoola	map/inv.009	0
☐	✎	✕	10	mr.ayanlowo	map/inv.0010	0
☐	✎	✕	11	mr.Ayeleso	map/inv.0011	0
☐	✎	✕	12	Mr.leke adebayo	map/inv.012	0
☐	✎	✕	13	mmr.kola abiola	map/inv.013	0
☐	✎	✕	14	mr.odiete joseph	map/inv.014	0
☐	✎	✕	15	mrs.lawal	map/inv.015	0
☐	✎	✕	16	mr.gbangboye	map/inv.016	0
☐	✎	✕	17	mr.micheal	map/inv.017	0

Check All / Uncheck All *With selected:* ✎ ✕

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Time Table Database

			id	time_table_id	period	day	data	period_data	status
☐			160	1	4	10	com 344: computer science HND 1 food 121: foodtec...	com 344: computer science HND 1 120 food 121: foo...	0
☐			159	1	3	10	slt 212: SLT ND 2 pham 222: pharmtech ND 2 mst 2...	slt 212: SLT ND 2 150 pham 222: pharmtech ND 2 90...	0
☐			157	1	1	10			0
☐			158	1	2	10	com 111: computer science ND 1 mst 111: math and ...	com 111: computer science ND 1 100 mst 111: math ...	0
☐			155	1	3	9	pham 311: pharmtech ND 2	pham 311: pharmtech ND 2 70 HALL NAME: 800 seate...	0
☐			156	1	4	9	com 133: computer science ND 1 mst 134: math and ...	com 133: computer science ND 1 110 mst 134: math ...	0
☐			152	1	4	8	slt 114: SLT ND 1	slt 114: SLT ND 1 150 HALL NAME: OGD hall 1500 n...	0
☐			153	1	1	9			0
☐			154	1	2	9	mst 412: math and stat HND 1	mst 412: math and stat HND 1 95 HALL NAME: keshi...	0

APPENDIX C

FLOW CHART AND UML DIAGRAM

System Flowchart

3.10 SYSTEM FLOWCHART

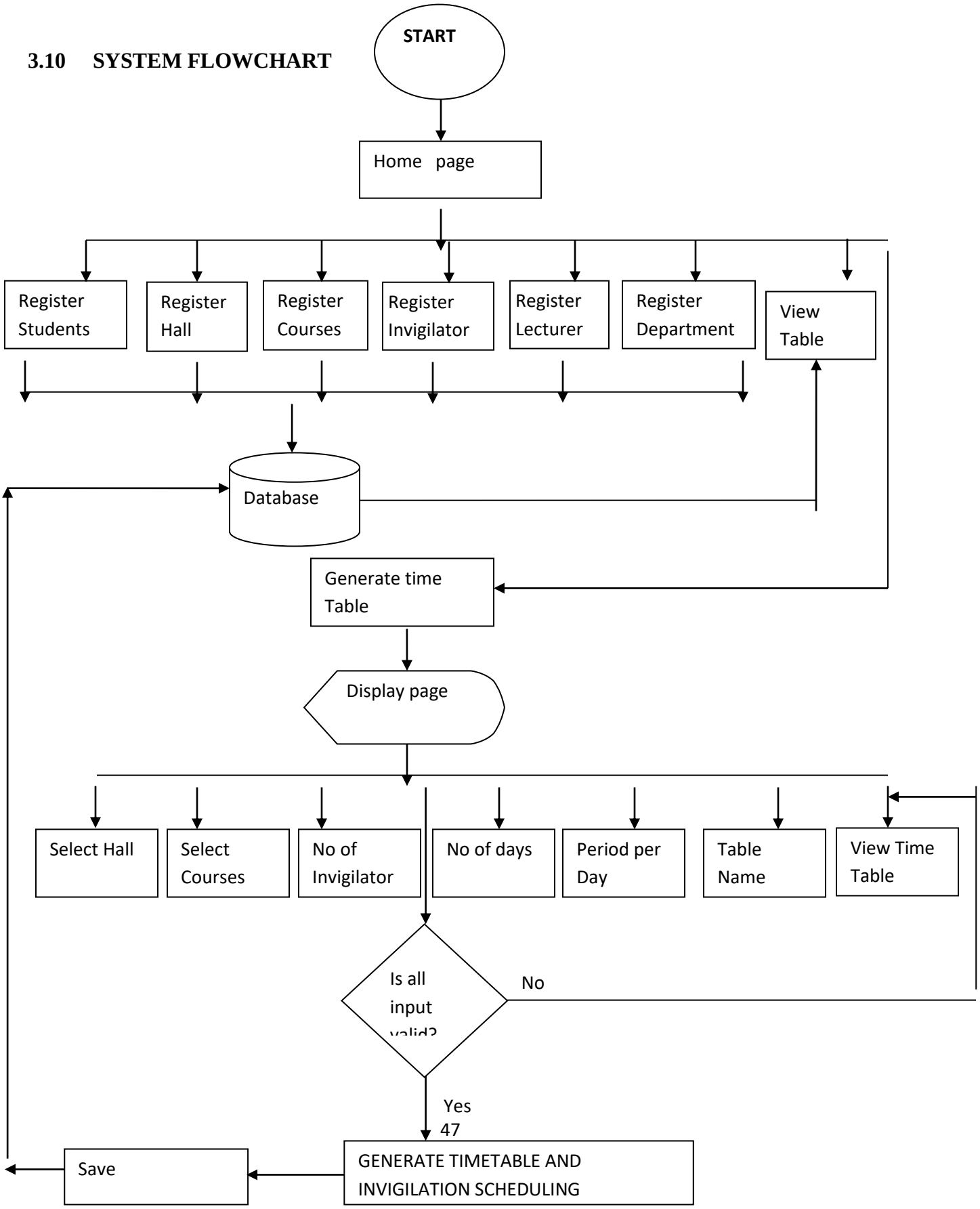
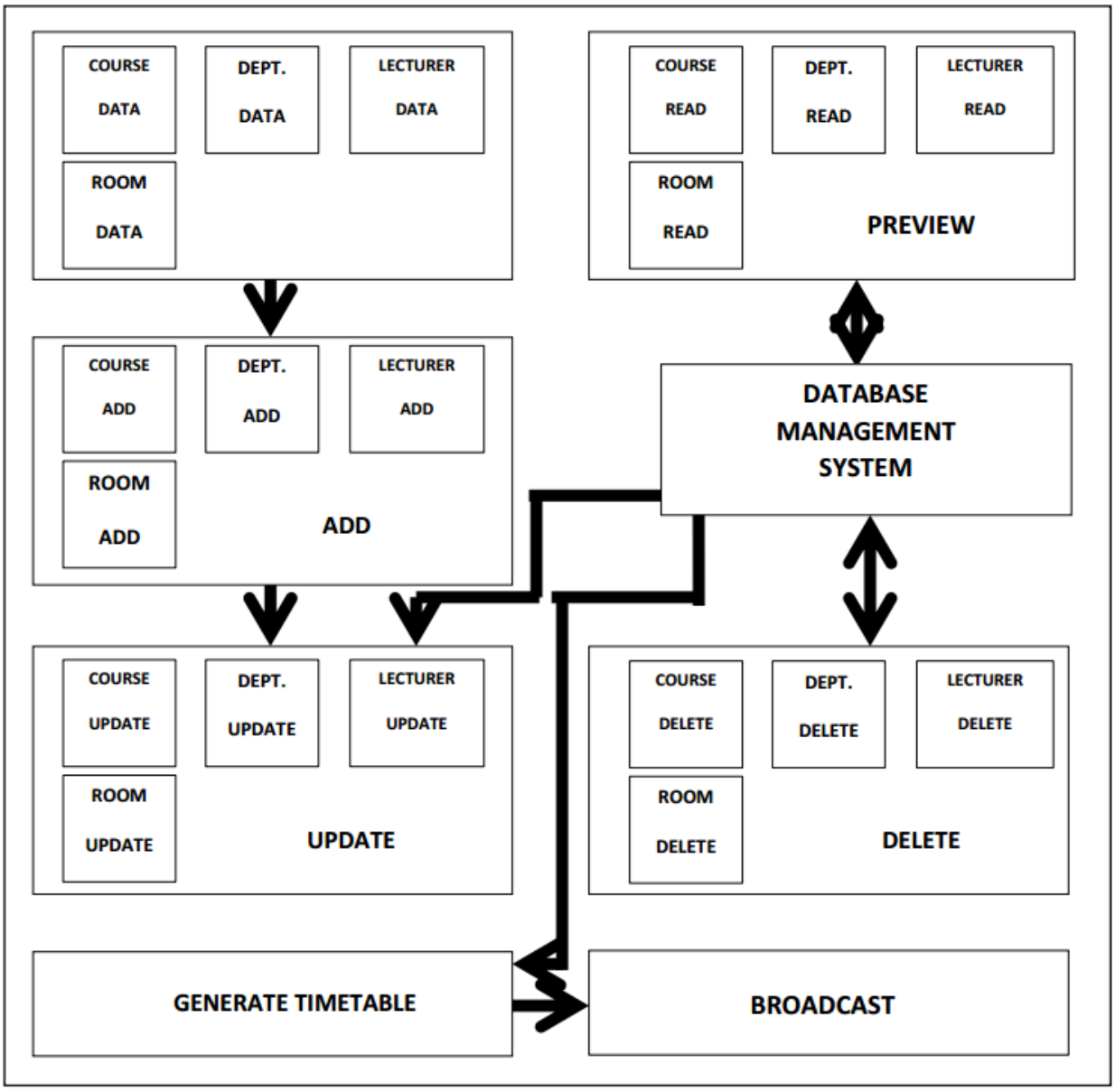
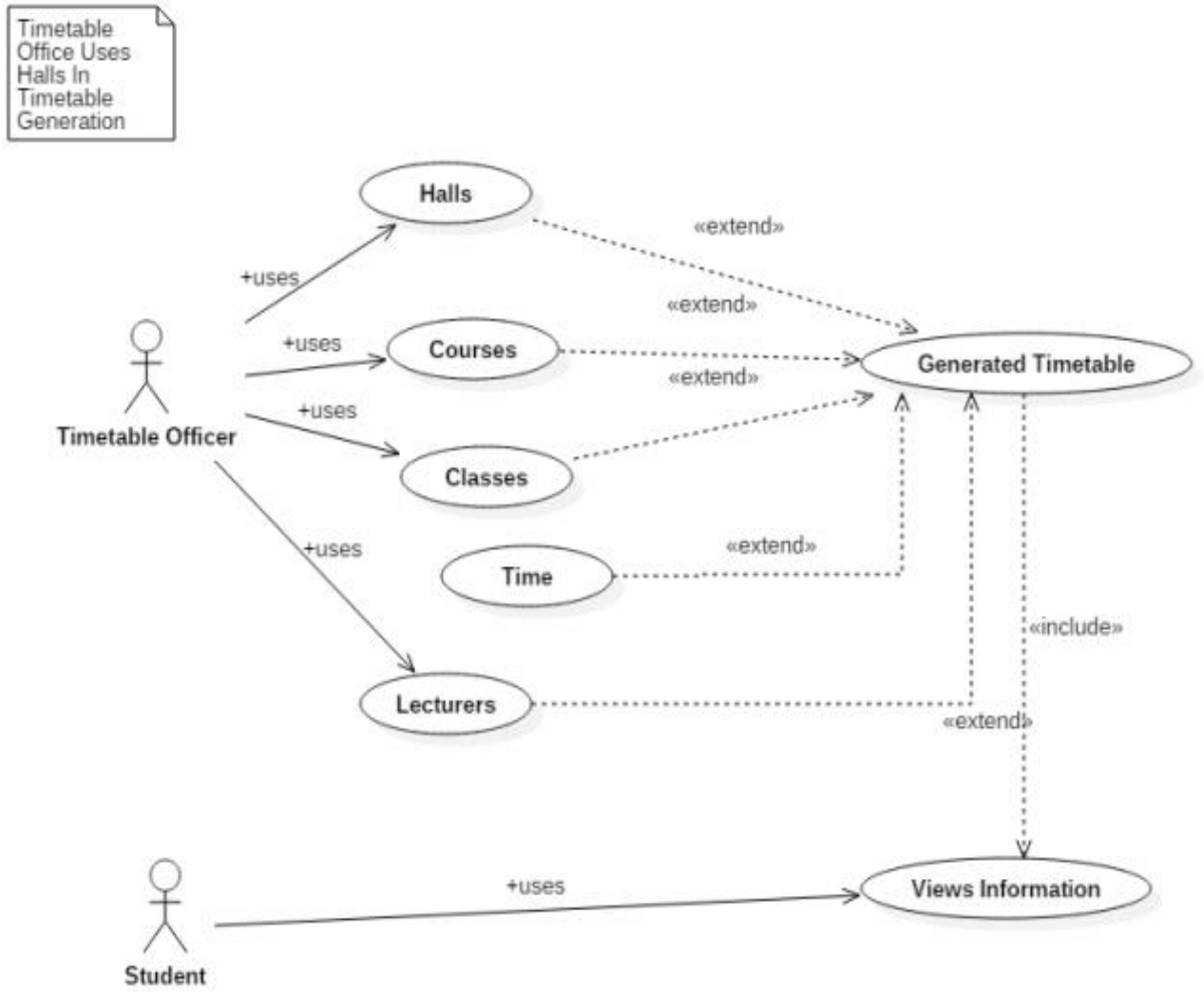


Fig. 10: system flowchart

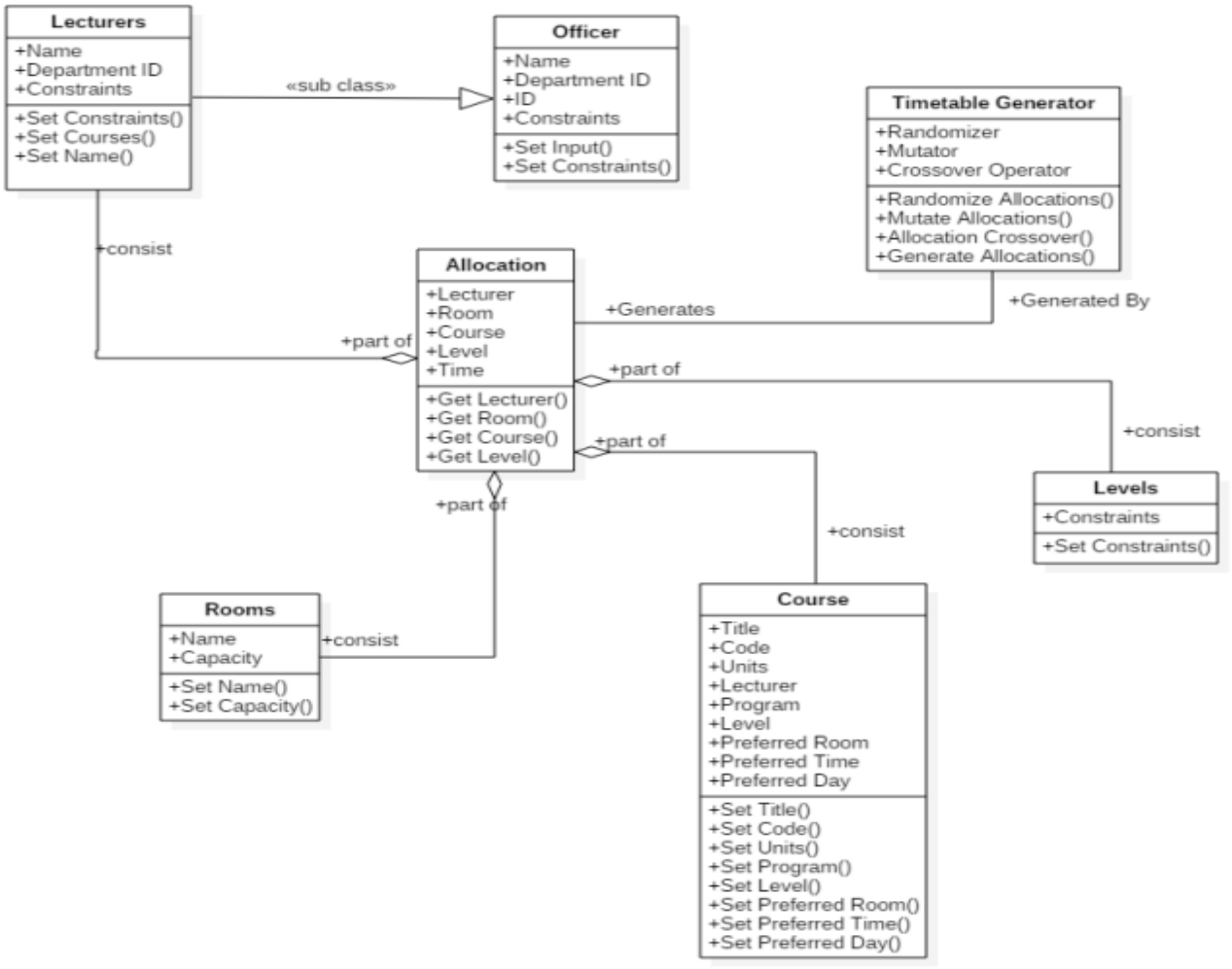
System Block Diagram



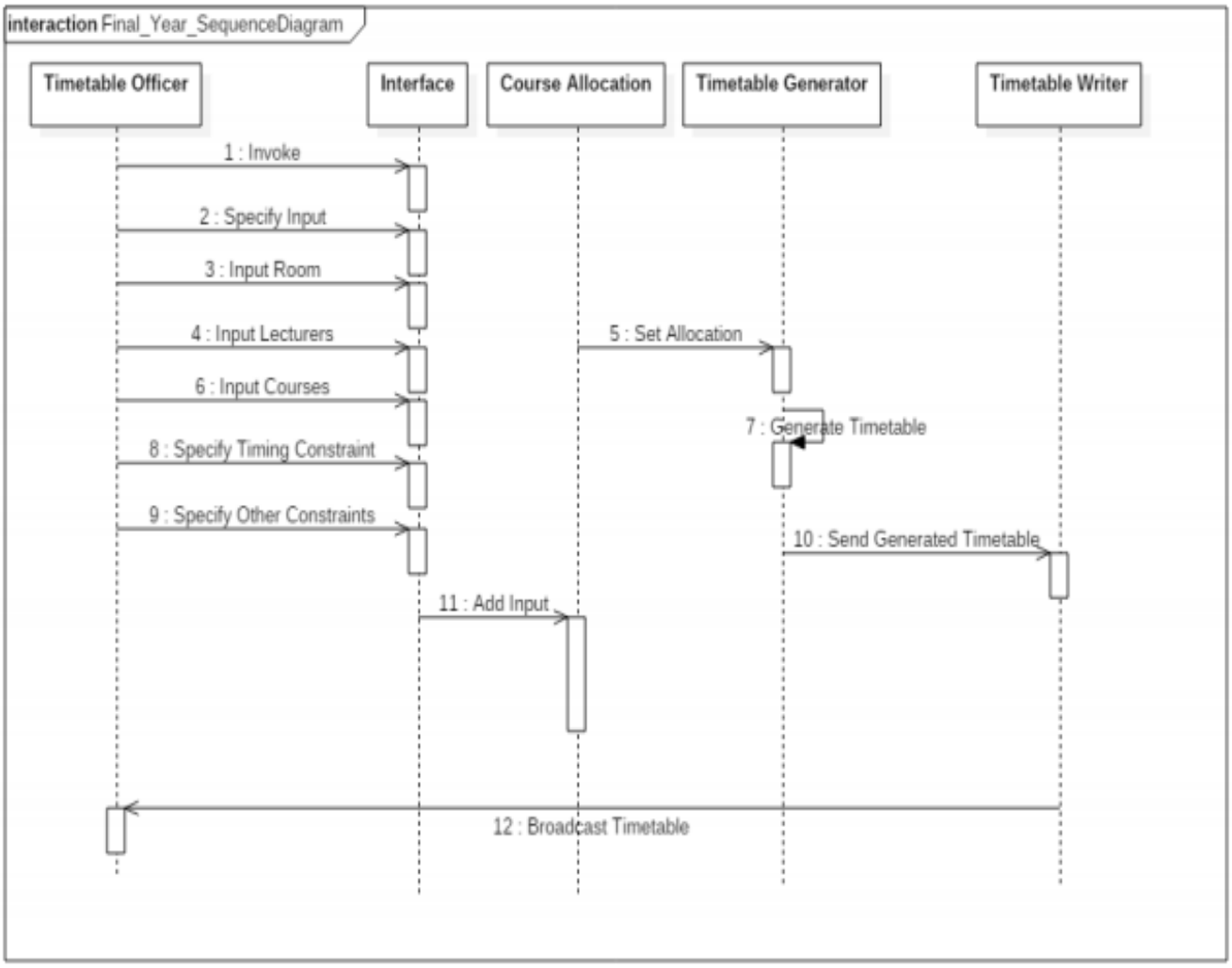
Use Case Diagram



Class Diagram



Sequence Diagram



APPENDIX D

SOURCE CODE

```
<<noscript></noscript>
<noscript id="yui_modules_entrypoint"></noscript>
<title>A System for Automatic Construction of Exam Timetable Using Genetic Algorithms (PDF Download Available)</title>
<meta name="description" content="Official Full-Text Paper (PDF): A System for Automatic Construction of Exam Timetable Using Genetic Algorithms"/>
</head>
<body class="logged-out">
<div id="page-container">
<script nonce="rSQzAFMxjxTac5cwlvitSZ1EG0JM1BEb">
var googletag=googletag||{};googletag.cmd=googletag.cmd||[];
(function(){var
gads=document.createElement("script");gads.async=true;gads.type="text/javascript";gads.src="https://www.googletagservices.com/tag/js/gpt.js";var
node=document.getElementsByTagName("script")[0];node.parentNode.insertBefore(gads,node);
})();
</script><div id="main" class="logged-out-header-support fluid">
<div id="content" class="">

  <div class="c-box-warning full-width-element" style="text-align: center; ">
    <div style="margin: auto; padding:10px;" class="container">
      <b>
        You are using an outdated version of Firefox which is not supported by ResearchGate anymore. For a faster, safer browsing experience, <a href="://whatbrowser.org/" rel="nofollow noopener"  target="_blank">upgrade your browser now</a>.
      </b>
    </div>
  </div>
  <noscript>
```

For full functionality of ResearchGate it is necessary to enable JavaScript.
Here are the
instructions how to enable JavaScript in your web browser.

</noscript>

<meta itemProp="datePublished" name="datePublished" content="2008-06-01"/>
<meta itemProp="image" name="image" content="https://i1.rgstatic.net/publication/228617160_A_System_for_Automatic_Construction_of_Exam_Timetable_Using_Genetic_Algorithms/links/555b58fd08aec5ac22322acf/largepreview.png"/>

See all › <span class="publication-resource-link-text" data.