# CHAPTER ONE

# INTRODUCTION

# 1.1 BACKGROUND OF THE STUDY

The management of timetable in the educational institution is very crucial as to have well-managed schedule lectures, labs, semester and course. This system is essential in distributing the resources needed in a right way without any faults. As seen nowadays, a lot of solution or application system that help in generating a timetable which has developed in an artificial intelligent way.

Due to the increase in course offerings and enrollment surges, the demand placed on facilities of academic institutions has become relentlessly high. The ability to work within the given constraints of time, facilities and resource persons becomes the greatest asset of any institution.

School timetabling problems vary between different institutions in terms of the constraints specified by the educational system. In most schools, timetables are manually designed and Lecturers usually set aside a week or a weekend for this, thus taking up valuable time in the process. In addition to being a time-consuming process, the manual design of a timetable is subject to human error and may not satisfy all the constraints.

However, because of its inherent challenges, timetable scheduling is still handled manually. For each semester or term, universities, colleges, high schools and several other educational institutions are obliged to produce time tables. It is a very repetitive, tedious and painful job to build timetables manually, and the need to automate this exhausting operation.

Failure to address timetable problems will lead to schedules with a maximum number of disputes that fail to meet a number of side restrictions, allotted times and halls within a restricted period of time. It is within this context that this project seeks to assess and fill this gap by designing and implementing a timetable system for computer science department. In Kenpoly.

**1.2 STATEMENT OF THE PROBLEM**

Because time table has always been done manually in all departments in Kenpoly, it becomes a tedious job prone to errors.

Secondly, this time tables can be altered by anyone due to ease of access.

More, the absence of course curriculum along the time table, making students not to know their area of concentrations.

Lastly, it can only be used as web-based application.

# 1.3 AIMS AND OBJECTIVES

The specific objective of this project is to design and implement a web-based timetable system for higher education institutions that;

1. Allows multiple user access to the system
2. Enforces permissions and privileges to all the students
3. Can be integrated into an existing school management system
4. Has the course curriculum alongside it.

**1.4** **SCOPE OF THE STUDY**

The project is mainly focused on resolving the issue of school lecture timetabling and therefore will not address scheduling of exams. It will be assessed at Kenpoly, Bori in the department of computer science and will also consider different criteria based on the form of classes, lecturers, rooms and laboratories affiliated with the relevant department. Efforts would be placed in motion to ensure that the method built is sufficiently generic to satisfy the different limitations that come with the department.

**1.5** **SIGNIFICANCE OF THE STUDY**

Management will produce conflict-free timetables with less commitment and greater performance, thus meeting all the challenging constraints concerned with far less duration.

Lecturers, students and other users can work and view the timetables on their devices simultaneously due to the availability of the system considering it’s a simple web-based interface which will facilitate ease of use.

The outcomes of this project would provide stakeholders and future researchers in the field of education, information systems and other related technologies useful information which should lead to the generation of new ideas for better implementation of a timetable management system.

# 1.6 RESEARCH METHODOLOGY

The evaluation of the existing system will be carried out using the Joint Application Design (JAD) approach. JAD will facilitate cooperation among all stakeholders to enhance understanding and build upon teamwork. JAD will ensure that both the users and developer have a shared vision of the outcome of the new system.

The life cycle of system creation (SDLC) is a logical structure used in project management that defines the stages involved in the development of an information system project, from a preliminary feasibility analysis to the product lifecycle management.

The type of SDLC to be used for this project is Rapid Application Development (RAD). RAD is an incremental design model for software creation that stresses an incredibly brief period of development. Again RAD being a ‘high speed’ adaptation of the linear sequential model would rapidly develop and achieve the goal of this system by using component-based construction. The justifications for using RAD for this project include;

1. The RAD process would enable me to build a completely operational program within very brief periods of time, after understanding requirements well and the project scope constraints
2. RAD as a software or an organizational tool available provides consistent collaboration in an interactive electronic system for a given activity where users can quickly access it.
3. It is designed to continue providing access to information, directions and support instantly.
4. RAD system provides a selection of tools that allow the design of graphical user interfaces which might usually take a great effort to build.

The tools to be used in the development of the proposed system include;

1. JSON for the Backend database.
2. JAVASCRIPT as the main programming language.
3. CSS as the style sheet language.
4. HTML as complementary programming language
5. BOOTSTRAP as the style sheet framework.

# 1.7 DEFINITION OF TERMS

**Programming:** The act of instructing a computer to do your bidding

**Database:** Organized store computerized data.

**SDLC:** System development life cycle

**JAD:** Joint Application Design

**RAD:** Rapid Application Development

**HTML:** HyperText Markup Language, used to build structure of every site

**CSS:** Cascading Style Sheet, used for the design and look of the system

**JAVASCRIPT:** A programing language used to build the interactivity and dynamicity of the system

**JSON:** A Javascript object notation, is a system used to pass data over the internet

**BOOTSTRAP:** A css framework which comprises of classes used for easy design of the system

**CHAPTER TWO**

# LITERATURE REVIEW

**2.1 INTRODUCTION**

A timetable is an ordered collection that contains details on activities that are scheduled to be held. Timetabling can be divided into many categories, including timetabling for school, scheduling of employees, timetabling for athletics and timetabling for travel. The composition, constraints and specifications of any of these timetabling concerns vary. Timetabling research continues to draw researchers' interest because of constantly enforced additional requirements/restrictions and with the end consumer insisting on more and better solutions.

In the words of Suresh et al. time tabling is the assignment of a given resource to objects that are put in space time, subject to constraints, in such a way as to achieve a collection of desired goals as nearly as practicable"

Another definition given by Burke & Carter is that: "A timetabling problem is a problem with four parameters: T, a finite set of times; R, a finite set of resources; M, a finite set of meetings and C a finite set of constraints. The problem is to assign times and resources to the meetings so as to satisfy constraints as far as possible".

Based on these definitions, timetabling challenges include allocating activities into appropriate time slots and services whiles fulfilling the constraints with the objective of improving the problem's fitness function.

**2.2 REVIEW OF RELATED WORK**

There are numerous free and commercial application that enables the user to automate or semi-automatically generate timetables (iteratively). Many applications provide us with graphical interface to direct the user through the timetable development process that allows users to quickly identify school resources and timetable constraints. The most widely used programs and their internal workings are discussed in this section.

### **2.2.1. ACTIVETIMETABLE 2003**

ActiveTimetable 2003 is a timetable preparation application for a variety of educational establishments. Absolute automated scheduling of all the most difficult timetables in a limited period is the biggest advantage. From data input, one can schedule many different timetables and then choose the best suited to the institution's specifications. The data entry method is streamlined by the schedule wizard, which helps you to easily construct the skeleton of the schedule.

The main features for the ActiveTimetable 2003 application are as follows;

1. Schedule blocks and multi-classes lessons
2. Spreads each topic's lessons equally throughout the week
3. Allows one to specify the full amount of hours a day for lecturers and classes.
4. Permit the timetable to be exported to html

However, the following limitations have be identified

1. Lack of multiuser use
2. Lack of independent platform
3. No accessibility from remote locations
4. Students cannot view timetable schedules by themselves

### **2.2.2. CMIS SCHEDULER (CCM)**

An event-based planner that is able to coordinate not just the school schedule, but any kind of activities, such as workshops, conferences, visits and other school activities, it is also possible to use as a professional resource manager. The preparation matrices allow a user to conveniently organize the school curriculum by appointing lecturers and facilities, from developing the initial timetable to ensuring that certain rules or constraints are not violated. The features for CMIS Scheduler include;

1. Color coding for displaying blocked and allocated routine status
2. Ordering of user-controlled courses, lecturers, subjects and facilities
3. Different planning layout
4. There is a drag and drop from one class to another in order to copy curricular elements

However, the following limitations have be identified

1. No support for other languages
2. Lacks export to other file-based applications
3. Lacks capabilities to combining multiple classes for a given period
4. No accessibility from remote locations

### **2.2.3. CYBER-MATRIX CLASS SCHEDULER**

Cyber-Matrix Class Scheduler is a perfect application for universities and other colleges that need to plan classes easily. Block scheduling encourages classmates to be arranged instead of only single students. A narrower resource footprint was also substituted by the database engine. The features for Cyber-Matrix class scheduler include;

1. It is possible to join a group of required classes and pick a list of potential schedules from.
2. You can schedule complete classes of students at one time.
3. Share scheduling details for all of your students and lecturers across ones network.
4. A search mechanism allows all students taking a single class or sharing a similar classification, such as band students, to be found.
5. Non-English speaking users can quickly translate the text of the app into the equivalents of their very own language.

However, the following limitations have be identified;

1. Lacks export to other file-based applications
2. Lacks capabilities to combining multiple classes for a given period
3. No accessibility from remote locations
4. Students cannot view timetable schedules by themselves

**2.2.4. iMagic** iMagic Timetable allows one to build schedules for academic institutions. The main features for the iMagic application are as follows

1. Save time inserting details manually
2. Has a search and replacement of entered data
3. Easy to use interfaces
4. Functionality to export output to CSV file format or a web page

However, the following limitations have be identified

1. Lack of multiuser use
2. No support for other languages
3. No accessibility from remote locations
4. Students cannot view timetable schedules by themselves

**CHAPTER THREE**

# ANALYSIS AND DESIGN

**3.1 METHODOLOGY**

Methodology is the systematic, theoretical analysis of the methods applied to a field of study, or the theoretical analysis of the body of methods and principles associated with a branch of knowledge.

Many methodologies can also be found for developing time tables. The various algorithms for designing time tables have been illustrated below.

### **3.1.1 GRAPH HEURISTICS**

The use of graph colouring which has been identified as an efficient and fastest algorithm where Colors are assigned to vertices, such that no neighboring vertices attains exact or similar color. Other scholars have concluded that graph heuristics, according to their own, are not really suitable techniques for solving global timetabling problems and they have not even created feasible solutions for some of the problem cases. Latest evidence, however, has shown that they are successful in generating an actual meta-heuristics remedy.

### **3.1.2 HILL CLIMBING (HC)**

Hill Climbing (HC) or simple climbing is a classical local quest technique. Hill climbing is fast and quick to execute. The downside, however, is that the localized Optima is quickly attained. Recently Hamed et al. suggested a late acceptance technique for scaling the HC. The mechanism postpones the step of comparison between the candidate solution and the actual (best) response.

### **3.1.3 TABU SEARCH (TS)**

Tabu search suggested by Glover, works in a manner comparable to HC but adding a memory to enable analysis of the differentiation of search space. Tabu search has been enforced as a search algorithm that generates course schedules by minimizing losses heuristically over an utterly impossible solution.

### **3.1.4 SIMULATED ANNEALING (SA)**

Simulated annealing (SA) has been suggested by Kirkpatrick, Gelatt Jr. and Vecchi. It was inspired by the physical annealing mechanism to heat up a solid to an extreme temperatures and cool it off slowly until everything crystallizes and there are no more changes. SA begins with an initial solution built using a positive heuristic and an optimal solution are always approved, whereas a much worse solution is only acknowledged with certain probabilities.

### **3.1.5 GENETIC ALGORITHMS (GA)**

Genetic Algorithms is a population-based search that produces appropriate results from one generation to another through the theory of biological evolution. In order to avoid infeasible solutions Esraa & Ghada, advised using only the mutation operator to produce solutions for the offspring. Experimental results found that the genetic algorithm used by a generic crossover operator was less effective than their approach. Because of the direct chromosome representation that created infeasible offspring solutions, they introduced a repair mechanism to solve the infeasibility.

# 3.2 ANALYSIS

In order to overcome the timetabling challenges affecting universities, the proposed web-based schedule systems will be built. The system would include the option to input the different classes, course codes, lecturers, curriculum, and the identification of a few constraints.

### **3.2.1 STAKEHOLDERS**

They are key people, groups of people who are interested in a project or initiative or that may be influenced actively or passively by the process or the outcomes. The list of stakeholders are explained below

As the sponsor for the project, the Rector will be responsible for monitoring the progress of the project to ensure that they meet the stated goals and objectives as well as general mission of the academic institution.

The Dean/ Head of Departments are responsible for ensuring that lectures and students under them comply with schedule policies and are identified as key stakeholders.

Lecturers are going to be the active users of the output from the proposed System and are identified as key stakeholders.

Students are required to adhere to the schedules specified in timetables and are identified as a stakeholder

### **3.2.2 REQUIREMENT GATHERING**

The method for the requirement gathering will be a Joint Application Design (JAD) to facilitated cooperation among all relevant stakeholders

The JAD session facilitated discussion among stakeholder in order to elicit requirements. Below are some of the discussion topics or tools used in the session.

1. What is the effect of the inability to solve the problem?
2. What is your guess about how the new system will solve the current problem?
3. What features do you think when added to the web-based timetable system will solve the current problem?
4. Where would the user be located physically when using the web-based timetable System?

#### **3.2.2.1 FUNCTIONAL REQUIREMENTS**

Functional requirements describe what the proposed Web based timetable system should do or the behaviors of the system and below are the requirements it should satisfy.

1. The proposed system should provide a login interface through which only authorized users are allowed to use based on their roles (Administrators, Heads of departments (HOD), Lecturers, students etc.).
2. The system should allow administrators to manage, that is add, modify, delete lecturers, time, dates, even curriculum.
3. The proposed system should be online to allow access from anywhere
4. The proposed system should support any device that can access a webpage online
5. The system should handle multiple users simultaneously
6. The proposed system will allow administrators to monitor activities of logon users
7. The system will support all web browsers with cookies and JavaScript enabled.
8. The proposed system will use JSON as the DBMS for database.
9. The propose system will use JAVASCRIPT as the server side script

#### **3.2.2.2 NON-FUNCTIONAL REQUIREMENTS**

A non-functional requirement places constraint on how the proposed web-based timetable system should work or elaboration of the performance characteristics of the system. Below are the requirements it must satisfy

1. The proposed system should have an interface that is user friendly, easy to follow and indicate the name and subject of the authorized user
2. The proposed system should have an interface that produces relevant error messages related to user login and system usage
3. Access to features such as marks entry and backup of database should depend on users designated role. The web server software should offer good performance by employing server-side caching
4. The proposed system should be available 24 hours, 7 days a week and can be accessed anytime
5. The proposed system should have a server-side backup of 12 hours duration and UPS support for 24 hours, 7 days a week to limit server down times
6. The proposed system should not cost the student a dime.
7. The proposed system should implement SSL (https) and RSA encryption to secure the contents of the system
8. The proposed system should use HTML, CSS, JS, etc. which are platform independent and can be implemented with minimal efforts
9. The proposed system should run on a Linux-based cloud system with at least 4GB RAM and 100 GB hard disk space

### **3.2.3 MAJOR FEATURES/COMPONENTS OF THE PROPOSED SYSTEM**

The key components of the system that seeks to replace the manual way of timetable preparation have been outlined below.

A backend database is an essential component of the proposed system. JSON is a common database of open sources used in the Javascript world. Its success derives from the potential to reuse the code of several other user groups, which boosts DBMS performance and efficiency. The proposed web-based timetable system will maintain JSON objects to store information relating to timetable building. Tables will be created for the various assets that are required in timetable preparation such as lectures details, etc. For this new system, this DBMS would be a reasonable alternative and the current mechanism could then be used, removing a need to implement a new structure.

Again, an interactive user interface design will ensure that actors easily make choices on the system. Relevant feedback on the status of processing will be communicated to ensure a user-friendly application usage.

The architecture and structural design component will leverage on programming languages which are architecture independent to guarantee successful integration into other applications. The primary programming language for the system will be Javascript. A Linux environment is a preferred choice for the web-based system due to its secure architecture.

|  |
| --- |
| ***Figure 3.1:*** *Use case diagrams Context Diagram for the Web-based Timetable system.* |

### **3.2.4 ADVANTAGES OF THE PROPOSED SYSTEM**

The web-based timetable system is proposed to solve the problem of paper and manual way of preparing time table. The benefits to be derived from the new system are outlined below.

1. The system provides consistency, with the exception of the manual timetabling system.
2. It almost eliminates paperwork by utilizing minimal processing/computing power.
3. It greatly reduces the time needed to generate timetables.
4. With an intuitive interface, it offers a simple means for entering data and modification.
5. It simplifies the timetabling process.
6. It has a One-click total system backup and restore across different servers
7. Generate ready-to-print timetable image snapshots
8. Create routine backup copies of database.
9. Secure management and use of passwords for authentication purposes.
10. Restrict access to administrative interfaces

## 3.3 DESIGN

The functional processes of this system are the major features that the system has or performs. A functional model or functional process in system development is a structured representation of the functions, behaviors, activities or processes of the system or subject area. The functional process of the system include all the working and operational processes of the entire system

## 3.3.1 USE CASE DIAGRAMS

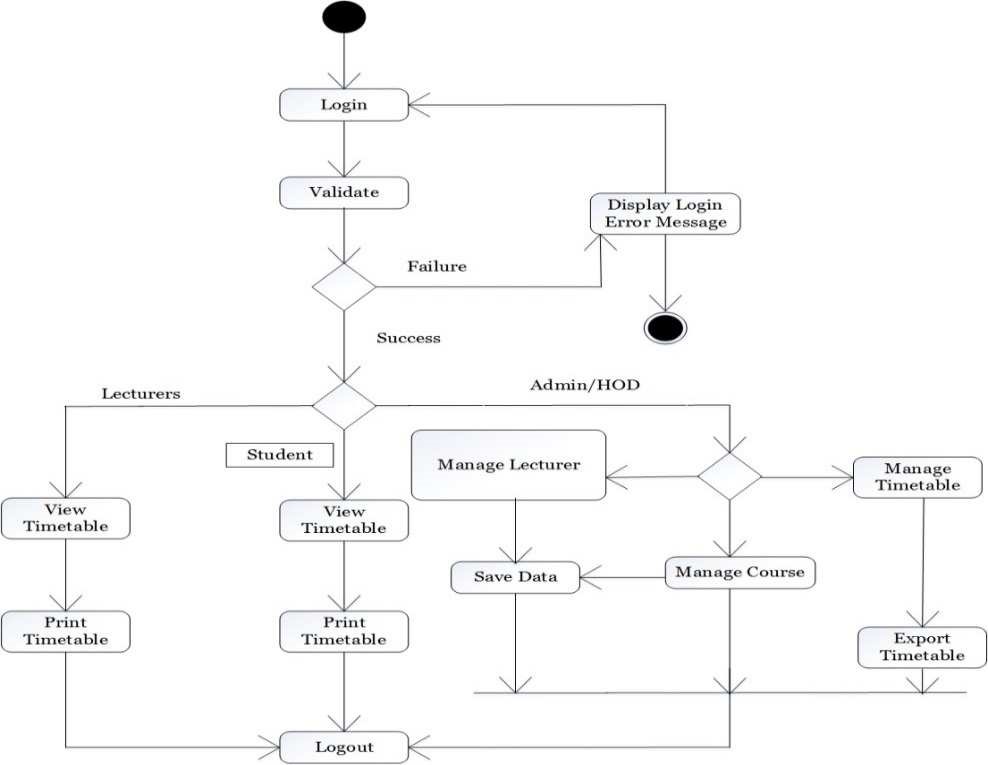
Use case diagrams are used to describe the main processes and functionality of the proposed Web-based Timetable System. The main objective for the use case diagram is to outline the scope of the system. Use case diagrams have been created for the various actors of the proposed Web-based Timetable system that is lecturers, administrators and students of the system.

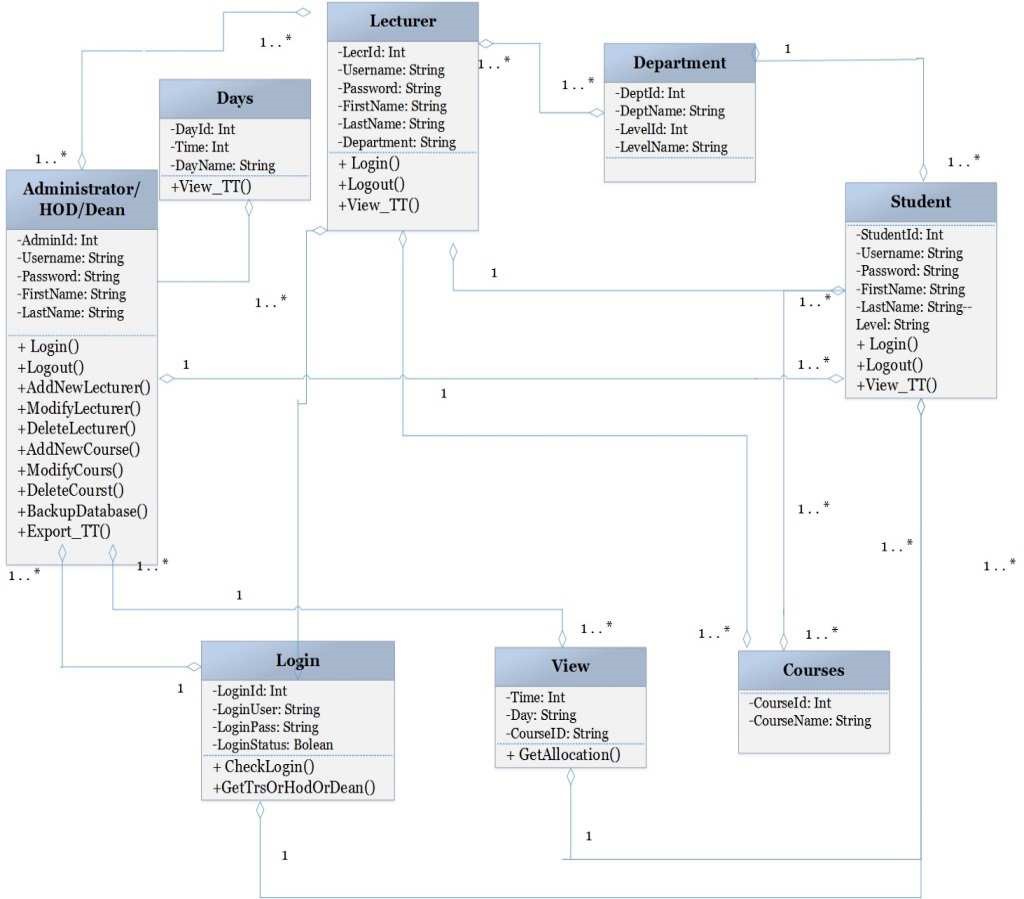
The key actors for the proposed Web-based Timetable System include lecturers, administrators and students. The clients for the system include students and lecturers. The secondary actors for the system include Head of Departments, Dean and the Examination Committee. The governing bodies for the system include the Board of service providers for the proposed Web-based Timetable is Governors for the University, Academic Board and the System Developer. Student Representative Council

|  |
| --- |
| ***Figure 3.2:*** *Use case diagrams* |

## 3.2. ACTIVITY DIAGRAM

Unified Modeling Language like activity diagrams is purposely designed to model out both computational and organizational process [13]. The essence of Activity Diagram for the proposed Web-based Timetable System is to model the workflow of its Use Case to show paths within it. It will show the sequence of an activity from the other. The Activity Diagram will illustrate existing functionalities and how those functionalities coordinate with other pieces functionalities of the system. The general overview or description of the for the Timetable System activity diagram is depicted in figure 3 below.

 ***Figure 3.3:*** *Activity Diagram for the Web-based Timetable System.*

 ***Figure 3.4:*** *Class Diagram for the Web-based Timetable System.*

## 4.3. CLASS DIAGRAM

A sort of static structure diagram is a class diagram in objects-oriented modeling that illustrates the component or structure of a system depicting attributes, operations as well as the relationships that exist among the classes of that system. The relevance of Class diagram in the Online Result System is to graphically show what the various objects will perform by showing the interactions amongst the various classes on the system. The web-based timetable System has been divided into classes with strong coherence. The main system shows the existence of the following classes;

1. Administrator/HOH/Dean class
2. Lecturers class
3. Student class
4. View class
5. Courses class
6. Department class
7. Days class

The Administrator/HOD/Dean class is responsible for adding, modifying and deleting lecturers, courses, departments, rooms and levels. The class can also generate timetable and backup information from the whole system. The attributes are AdminId, Username, Password and Name. The operations or methods include login(), logout(),

AddNewLecturer(), ModifyLecturer(), DeleteLecturer(), AddNewCourse(), ModifyCourse(), DeleteCourse(), SetupTT(), AllocateTT(), BackupDatabase() and View\_TT(). Administrator class is associated with Lecturers and students in a many-to-many relationship because there can be more than one administrator with many Lecturers and students.

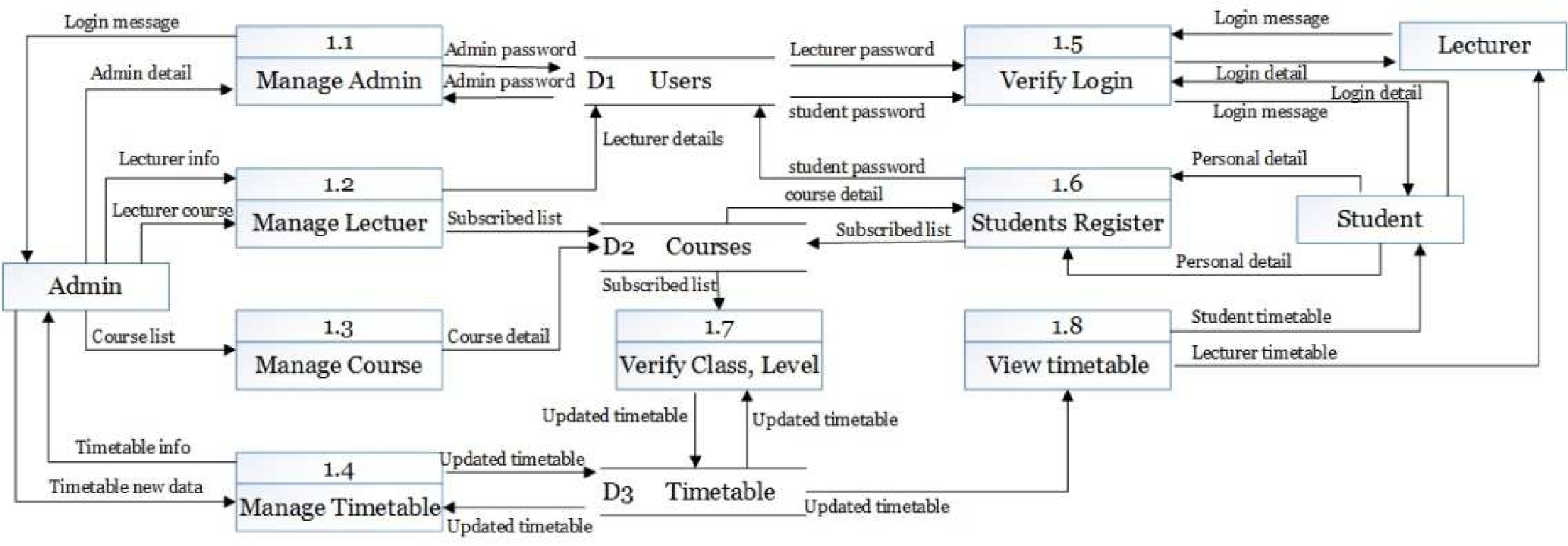
The lecturer class is responsible for viewing of the completed timetable and printing them out. The attributes are LecturerID, Username, Password, FirstName, LastNAme, Gender, Department and LevelTeaching. The operations or methods include login(), logout() and View\_TT(). The Lecturer class is associated with students, levels, courses and login classes. Lecturers have one-to-many relationship to students because single lecturers can teach many students.

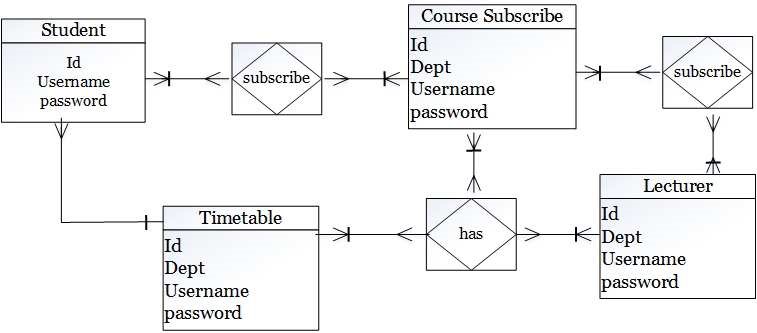
The student class is responsible for accessing the generated timetable. The attributes of the student class include StudentId, Username, Password, FirstName, LastName and level. The operations or methods are login(), logout(), View\_TT(). The student class is associated with Lecturers, administrator, levels, courses and login classes.

The view class is accessed by administrators, lecturers and students to view the completed timetable. The attributes of the view class include Time, Day, CourseId. The operations or methods of the generate report class are GetAllocation(). It is associated with students in a one-to-one relationship because a student is entitled to a view of the timetable.

The course class holds the data for the various subjects or courses. The attributes of the subject class are CourseID and CourseName. Courses have a one-to-many relationship with students and a one-to-one relationship with Lecturers.

The Department Class holds the data for the various classes under the Online Result System. The attributes of the class are DeptID, LevelID, DeptName and LevelName. It has a one-to-one relationship with students and a one-to-many relationship with Lecturers





***Figure 3.6:*** *Entity Relationships of the Web-based Timetable System.*

***Figure 3.5:*** *Data Flow Diagram level 1 for Web-based Timetable System.*

|  |
| --- |
| ***Figure 3.7:*** *Sequence Diagram of the Web-based Timetable System.* |

The login class is responsible for verifying login details of users. The attributes include LoginId, Login Username, LoginPassword, LoginStatus. The operations or methods are CheckLogin(), GetLecORHODOrDean(). It has a one-tomany relationship with students, Lecturers and administrators because many students, Lecturers and administrators can have only one login details.

**CHAPTER FOUR**

**IMPLEMENTATION AND DISCUSSION OF RESULT**

**4.1 REASON FOR CHOICE OF PROGRAMMING LANGUAGES**

The programming language used in the development of this game is Python. This is because python is one of the most demanding and most developing language. Also, its simplicity, efficiency and robustness make it a good option for 2D game development. Python is open source, and cross platform independent. It has vast libraries and is very extensive. the module used in designing this game is known as pygame. As defined in the earlier chapter, Pygame is a cross platform set of python modules designed for writing 2D games. It includes computer graphics and sound libraries designed to be used with the python

## 4.2 DOCUMENTATION

For the new system to be successful there is the need for documentations or manuals aimed at the following

1. Educating users on the current system's objectives and advantages.
2. Educating users of the current system on management roles or tasks and obligations.
3. Educating users on the procedure for undertaking tasks and functionalities on the new system
4. Most importantly, making the system documentations available to users.

### **4.2.1 LAUNCHING THE WEB-BASED TIMETABLE SYSTEM**

Please follow the sequence of steps to launch the application

1. Double click on a index.html
2. Wait for it to load up on your default browser

**4.2.2 LOGIN USING USERNAMES AND PASSWORDS**

Please follow the sequence of steps to enter access credentials for validation.

1. Click on login hyperlink
2. Locate and enter usernames and passwords in their appropriate fields
3. Click on Login button or press enter on the keyboard
4. Wait for the credentials to be validated to access the interface.

## 4.2.3 ADDING/MODIFYING LECTURERS

Please follow the sequence of steps to enter access credentials for validation;

1. Click on ‘Manage Lecturer’ hyperlink
2. Fill the form that appears by inputting the full name, department, username, password and role played at the department
3. The entries can be deleted by selecting it from the delete form
4. Click on the save button to complete the task

### **4.2.4 ADDING/MODIFYING DEPARTMENTS**

Please follow the sequence of steps to add a department to the system

1. Click on ‘Manage Department’ hyperlink
2. Fill the form that appears by inputting a numeric code and a name for the department (eg 002, Mathematics and Science Education)
3. The entries can be deleted by selecting it from the delete form
4. Click on the save button to complete the task

### **4.2.5 ADDING/MODIFYING COURSES**

### Please follow the sequence of steps to include courses on the system

1. Click on ‘Manage Courses’ hyperlink
2. Click and select the name of a lecturer
3. Fill the form that appears under add course by inputting the course id, course name, allowed levels and whether conflict allocations are allowed for that course
4. Click on the save button to complete the task
5. Existing courses can be deleted from the delete course form by selecting the appropriate course and clicking on the delete button.

**4.2.6 ADDING/MODIFYING LEVELS**

Please follow the sequence of steps to add levels under the various departments

1. Click on ‘Manage Levels’ hyperlink
2. Fill the form that appears under ‘add level’ by inputting the level name, selecting the department it falls under and an anticipated size for that level.
3. Click on the ‘Add’ button to complete the task
4. Existing levels can be deleted from the ‘delete level’ form by selecting the appropriate level and clicking on the delete button.

### **4.2.7 ADDING/MODIFYING TIMETABLES**

Please follow the sequence of steps to configure a new timetable

1. Click on ‘Manage Timetables’ hyperlink
2. Under configure timetable, type the name of the timetable or select existing timetables.
3. Under Number of slots enter the total number of hours the timetable can span and specifying the exact time under the ‘Start Time’ form including AM or PM
4. Under Number of Days, input the total days the timetable will be used
5. Conflict allocations and current active timetable, freeze timetable checkboxes can checked depending on the needs of an administrator
6. The save button is used to save current configurations
7. The ‘Update’ button implements effected changes on the interface
8. The ‘Download’ button is used to backup the database
9. The ‘Restore’ button is used to load an existing backup file by clicking on the ‘Browse’ button and locating the file on a harddisk or resource

## 4.2.8 ALLOCATING TIMETABLE

Please follow the sequence of steps to assign lecturers, courses and rooms to lecturers.

1. Click on ‘Allocate Timetables’ hyperlink

2. Under ‘lecturer’, select and click the name of a lecturer.

1. Under Courses, drag and drop course at appropriate day and time on the timetable preview.
2. Under ‘Assign Room’ select and click a room to be used for the lecture.
3. Click on the save button to store the current timetable.
4. The ‘Freeze’ or ‘Disable’ buttons can be clicked to prevent modification on the timetable

## 4.2.9 VIEW TIMETABLE

Please follow the sequence of steps to view the completed timetable

1. Click on ‘View Timetables’ hyperlink
2. Under ‘Timetable’, select and click the name of a timetable.
3. Under Department, click and select the department you wish to view its timetable
4. Under ‘Lecturer’ select and click a name for a lecturer.
5. Under ‘Level’, select the level you wish to view the timetable for.
6. The ‘Freeze’ or ‘Disable’ buttons can be clicked to prevent modification on the timetable
7. The ‘Export’ button can be used to export the final timetable in an image format.

**4.3 DISCUSSION OF RESULT**

The objectives of this system were fully realized.

# 4.4 SOURCE CODE

Refer to Appendix A

**4.5 SAMPLE OUTPUTS**

Refer to Appendix A

# CHAPTER FIVE

# SUMMARY, CONCLUSION AND RECOMMENDATIONS

# 5.1 SUMMARY

At the end of this project work, the software that can successfully manage our time table for the department of computer science, Kenpoly, was successfully built.

# 5.2 CONCLUSION

This study was undertaken to minimize the intensive manual effort that is being made to establish and build timetables for school lectures. The Web-based Timetable system is capable of producing near-optimal schedules depending on courses with reduced course constraints. The system timetable system allows multiple users access (administrators, HOD, Dean Lectures, students) irrespective of their location, provided there is an internet enabled device. School timetabling problems vary between different institutions in terms of the constraints specified by the educational system. In most schools, timetables are manually designed and Lecturers usually set aside a week or a weekend for this, thus taking up valuable time in the process. In addition to being a time-consuming process, the manual design of a timetable is subject to human error and may not satisfy all the constraints. But with this system approach, timetable management has been made simpler as it is a web application to generate timetable that can be easily shared on internet enabled devices.

# 5.3 RECOMMENDATIONS

The following was recommended after having understudied the difficulties found in the manual timetable system.

1. The timetable system built should be extended to handle the timetable management in the higher institution of the entire department (academic) and made available to all students.
2. In order to use the incremental model of software development, further work on developing a timetable system should be based on this research study.
3. A shared timetabling system model should be developed that uses mobile operating systems.

# REFERENCES

Ahmed, K., & Keedwell, E. (2017). "A Hidden Markov Model

Approach to the Problem of Heuristic Selection in Hyper

Heuristics with a Case Study in High School Timetabling Problems," Massachusetts Institute of Technology, 2017.

Burke & Carter (1998). Practice and Theory of Automated

Timetabling: *Selected Papers from the 2nd International Conference. Lecture Notes in Computer Science*, 14 (08), 115 129.

Esraa, A., & Ghada, E., K. (2016). "A Utilization-based

Genetic Algorithm for Solving the University Timetabling Problem (UGA)," *Alexandria Engineering Journal*, vol. 55, no. 2, pp. 1395-1409, 2016.

Fatih, C. & Merve, K. (2015). "A Fuzzy Logic and BinaryGoal Programming-Based Approach for Solving the Exam Timetabling Problem to Create a Balanced-Exam Schedule," *International Journal of Fuzzy Systems*, pp. 119-129.

Hamed, B., Jaber, K. & Amin, H. (2019). "Generating an optimal timetabling for multi-departments common lecturers using hybrid fuzzy and clustering algorithms," *Methodologies and Application*, vol. 23, p. 4735–4747.

Iosif, K., Ioannis, T. & Grigorios, B. (2015). "A Comparative Study of Modern Heuristics on the School," *Algorithms,* vol. 8, no. 3, pp. 723-742.

Joselynn, H., F. (2019). "The Effect of the Gainful

Employment Regulatory Uncertainty on Student Enrollment at For Profit Institutions of Higher Education," *Research in Higher Education*, p. 1 25.

Kheiri, A., & Ed, K. (2017) "A Hidden Markov Model

Approach to the Problem of Heuristic Selection in Hyper

Heuristics with a Case Study in High School Timetabling

Problems," *Evolutionary Computation*, vol. 25, no. 3, pp. 473501.

Mahmoud, S., Mahmoud, N., & Mohammad, M., M. (2020).

"Optimal Localization of Shopping Centers Using

Metaheuristic Genetic Algorithm," *Journal of Optimization in Industrial Engineering*, vol. 13, no. 1, pp. 167 176.

Nelishia, P., & Ender, O. (2019). "Automated generation of constructive ordering heuristics for educational timetabling," *Annals of Operations Research*, vol. 275, no. 1, p. 181–208.

Rakesh, B., Gupta & Pallav, I., (2014). "A new hybrid algorithm for university course timetabling problem using events based on groupings of students," Computers &

Industrial Engineering, vol. 78, pp. 12-25.

Schaerf A. (1999). A survey of automated timetabling. Artificial Intelligence Review, 13 (2), 87-127.

Siau, K., & Tian, Y. (2009). A Semiotics Analysis of UML Graphical Notations. *Requirements Engineering*, 14 (1), 15 26.

Suliadi, S., Ahmad, I., Maselan, A., & Siti, R. (2018). "A

Heuristics Approach for Classroom Scheduling Using Genetic Algorithm Technique," *Journal of Physics*: Conference Series, vol. 995.

Suresh, L., Ajinkya, K., Akash, T., Manasi, M., & Yogesh, C.

(2014). "Genetic Algorithm: Paradigm Shift over a Traditional

Approach of Timetable Scheduling," *in Proceedings of the 3rd*

*International Conference on Frontiers of Intelligent Computing: Theory and Applications* (FICTA) 2014,

Switzerland, 2014.

Tan, L., Joe, O., Yu-Beng, L. and Jetol, B. (2018).

"Implementation of Constraint Programming and Simulated

Annealing for Examination Timetabling Problem," *Computational Science and Technology*, vol. 481, pp. 175 184.

Zekang, L., Shiwei, H., Rui S., & Sijia, H. (2019). "Optimizing Vehicle Scheduling Based on Variable Timetable by Benders-and-Price Approach," *Journal of Advanced*

*Transportation*, vol. 2019, pp. 1 13.

Glover, F. (1986) “Future Paths for Integer Programming and Links to Artificial Intelligence,” *Computers and Operations Research*, Vol. 13, pp. 533-549.

E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and J. Woodward. Exploring hyperheuristic methodologies with genetic programming. In C. Mumford and L. Jain, editors, *Collaborative Computational Intelligence*. Springer, 2009.

Kirkpatrick, S., Gelatt Jr., C. D. and Vecchi, M. P. (1983) Optimization by Simulated Annealing. Science, 220, 671-680. https://doi.org/10.1126/science.220.4598.671

**APPENDIX A**

Source code

# APPENDIX B