ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF EDUCATIONAL SCIENCE AND TECHNOLOGY TEACHERS EDUCATION

TIME TABLE SCHEDULING SYSTEM FOR WELKITE UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF M.SC. IN INFORMATION TECHNOLOGY MANAGEMENT

By

Name	ID
1. Almaz Honja	GSR/5290/06
2. Mehuba Jemal	GSR/5309/06
3. EtaferahuTamiru	GSR/5298/06

Advisors Name: Dr.Sridhar Udayakumar

: Ato Tibebu Beza

June, 2015

Adama, Ethiopia

Approval Sheet

Submitted By

Almaz Honja _____ _____

Mehbuba Jemal _____ _____

Etaferahu Tamiru _____ _____

Student's name **Signature** **Date**

Approved By

1_____ _____ _____

Advisor name **Signature** **Date**

2._____ _____ _____

CO-Advisor **Signature** **Date**

3. _____ _____ _____

DGC chairperson/Dep't Head **Signature** **Date**

4._____ _____ _____

ADRGS/Dean of the School **Signature** **Date**

5_____ _____ _____

Dean, School of Graduate Studies **Signature** **Date**

DECLARATION

We hereby declare that the project entitled Time Table Scheduling System for Welkite University is submitted in partial fulfillment of the requirement for the award of the degree of Master of Science in Technical Teachers Education in Information Technology Management, Adama Science and Technology University, is a record of our own work carried out by us during the academic year 2015 under the supervision and guidance of Dr. Sirdhar Udayakumar and Ato Tibubu Beza , Department of Computing, School of Engineering. The extent and source of information are derived from the existing literature and have been indicated through the project at the appropriate places. The matter embodied in this work is original and has not been submitted for the award of any other degree, either in this or any other University.

This project is our original work and all sources of materials used for the project have been duly acknowledged.

S.No	**Student Name**	**ID**	**Signature**
1	Almaz Honja	GSR/5290/06	_____
2	Mehbuba Jemal	GSR/5309/06	_____
3	EtaferahuTamiru	GSR/5298/06	_____

June, 2015

Adama, Ethiopia

Acknowledgments

First of all, we would like to use this opportunity to express our gratitude and many thanks to God for helping us go through all the challenges we faced.

Next, we want to thank Dr. Sirdhar Udayakumar and Ato Tibebu Beza, our honorable advisors, for their great advice and assistance in making this project a piece of work. Next, we would like to thank Dr. Prathap Mani, TVET student's coordinator, for his support in our project development.

We would also like to thank the Dean of Wku for his great cooperation in our project development.

We would also like to thank the Registrar Office and Instructors of Wku for their kind cooperation in our project. We would also like to give special thanks for Instructor Ato Amanuel Tamirat.

Finally we owe huge personal thanks to Instructor Yealemzewud Mekonnen who continues to make his own inimitable contribution and for his great support by contributing his knowledge to be this work real and we would like to thank our friends in Adama who provided us feedback on this Project Paper.

Contents

List of Tables

List of Figures

Acronyms

WKU - Welkite University

CLP - A constraint logic programming

SDLC- System Development Life Cycle

PR- The probability of crossover

PM- The probability of mutation

SQL- Structure Query Language

CRC-Class-responsibility collaboration

GUI-Graphical-user interface

GSF-Java server face

JDK-Java development kit

UML-unified modeling language

XHTML-extensible hypertext markup language

Abstract

Timetabling is the allocation, subject to constraints, of given resources to objects in space-time domain to satisfy a set of desirable objectives as nearly as possible. Particularly, the university timetabling problem for classes or lectures can be viewed as fixing in time and space a sequence of meetings between instructors and students, while simultaneously satisfying a number of various essential conditions or constraints.

The timetabling problem concerns virtually every educational institution, be it high school, college, or university and thus requires to be solved effectively. This is usually done 'by hand', taking several days or weeks of iterative repair after feedback from lecturers complaining that the timetable is unfair to them in some way.

The basic challenge is to schedule lectures over a limited time period so as to avoid conflicts and to satisfy a number of side constraints.

The manual system of preparing time table in colleges with large number of students is very time consuming and usually ends up with various classes clashing either at same room or with same teachers having more than one class at a time. These are just due to common human errors which are very difficult to prevent in processes such as these. To overcome these problems people usually taking the previous year's timetable and modifying it but still it is a boring job to incorporate changes. To overcome all these problems we propose to make an automated system. The system will take various inputs like details of students, subjects and class rooms and teachers available depending on these input the time table can be scheduled with announcement of any crashing. And also our aim is to develop a general purpose which can efficiently generate optimal solutions.

CHAPTER ONE

1. INTRODUCTION

1.1 Background of the study

Wolkite University is among the 10 Universities that are established in 2011/12 that made the total number of public universities in Ethiopia thirty one. Wolkite University is located at Wolkite town in Guraghe zone that is located 155km from Addis Ababa on the way to Jimma. The University is established after the foundation stone was put by His Excellency the late Prime Minister Meles Zenawi in November 2009 G.C. The campus is located at the Gubre sub-town of Wolkite, i.e., 14 km away on the way to Hosaena or Watchamo.

Time table scheduling has been in human requirements since they thought of managing time effectively. It is widely used in schools, colleges and other fields of teaching and working like clash courses, couching centers, training programs etc. currently in welkite university time table scheduling was done manually with a single person or some group involved in task of scheduling it with their hands, which take lot of effort and time. While scheduling even the smallest constraints can take a lot of time and the case is even worse when the number of constraints or the amount of data to deal with increases. In such cases perfectly designed time table is reused for whole generation without any changes, proving to be dull in such situations. Other cases that can cause problem is when the number of employers/workers are weak, resulting in rescheduling of time table or they need to will on empty seats urgently.

Institutions/Schools/Collages/Universities like welkite university are the regular users of such time tables. The university needs to schedule their course to meet the need of current duration and facilities that are available to them. However, the schedule should meet the requirement of new course addition and newly enrolled students to fresh batches. This may result in rescheduling the entire time table once again for its entire batches and to be scheduled in shortest possible time before the batches course start.

1.2 **Statement of the Problem**

The teaching and learning process is highly controlled and administered by the scheduling system that is provided by the integration of the universities registrar and departments.

Due to the problems observed on scheduling in WKU (Wolkite University) it makes not to run the teaching learning process as expected. The following are among the problems that are observed are:

- ✓ Because of Scheduling timetable is hard to control, peoples post more than one tentative timetable at different time this makes confusion on Students as well as Instructors and boring on workers.

- ✓ The main problem of the class overlapping deals that due to Scheduling process single class may have more than one period assigned at the same time. Means two or more courses assign at one period.
- ✓ This is the problem that arises on and Class Schedule one or more class of student share the same room or laboratory room at the same time. Hence students waste time for searching of other room and the teaching and learning process is disturbed, which makes the current scheduling system difficult for this situation if number of rooms are small the problem is big.
- ✓ The instructors get in a difficult situation because the teacher has to teach two or more classes at the same time.
- ✓ The instructor unable to finish their teaching lesson plan on the specified time.
- ✓ In the current system there is huge wastage of time to plan and retrieve the schedule data. In other word, it wastes time to insert, capture, and process, edit and manipulate the data from the current scheduling system for searching free space period this makes the controlling system is too difficult.
- ✓ Making manually timetable is need more human to come with the best schedule, this makes peoples busy.
- ✓ Planning timetables is one of the most complex and error-prone applications. There are still serious problems like generation of high cost time tables are occurring while scheduling and these problems are repeating frequently. Therefore there is a great requirement for an application distributing the course evenly and without collisions

1.3 Purpose and Description of the Project

The purpose of the study is to develop an automated system for Welkite University which includes:

- ✓ To allows the customer to get services efficiently.
- ✓ Enables the employee to perform many tasks with in a specific time period
- ✓ Reducing data loss because of its recoverability and maintainability. Improving the integrity of the services.
- ✓ Giving schedule delivery services for the university student's, registrar, department and others residents in integrated manner.

1.4 Objective Of The Project

1.4.1 General objective

The general objective of the project to access the current scheduling system and develop a simple, easily understandable, and portable web based application which automates the timetable scheduling for Wolkite University.

1.4.2 Specific objectives

The specific objective of this project is:

- ✓ To design system that schedule accurate class timetable to interact with user
- ✓ To design databases which store information for instructors, rooms, sections and courses.
- ✓ To design attractive and easily understandable graphical user interface to in.
- ✓ To develop an application that can display and print the schedule of each room, section and instructor on the web.
- ✓ To generate report for any user in the system about time table, block, instructors etc.
- ✓ To implement System that can only managed by authenticated users or Administrator.

1.5 Scope of the project

The project Aimed to solve the major problems of time table scheduling service delivery and data integration and management at Welkite University. Taking WKU as a case, the project attempts to assess the problems associated with the classroom assignment, course assignment, notification and other schedule related servicing sections of the University. Based on such assessments, we designed and implemented a system that could be applicable to any government universities in Ethiopia. Only one university in Ethiopia is considered making the scope a bit narrower. .

Specifically, the project focused on:

- ✓ Managing a timetable
- ✓ Managing the course on time table
- ✓ Assign instructors for each course
- ✓ Preparation/generation of reports.
- ✓ We include class or section, room, laboratory and lectures time table for only regular students in Wku. With respect to time and resource management.

Note that the proposed system does not include extension students and or any other time schedule for other departments out of Welkite University.

On the other hand from the results of analysis in this project it can the following limitation

- ✓ The system will not integrate the whole module to create a unified system
- ✓ In case of data handling, processing and manipulation mechanisms like computer virus and system failure have great limitation on our project.

1.6 Significance of the project

The proposed project aimed to improve the overall timetable scheduling system of Welkite University. Since the system deals with time table scheduling (which is the base for course and instructor assignment), online notification (which is the best way to handle course based on course code) and others. Therefore, efficient and careful administration is required to make the best of them. In this regard, the project contributes to the achievement of such a noble goal. And also the newly proposed system will have many advantages in terms of students, teachers, registrar employee and the university in the following ways:-

- ✓ No class room and laboratory room clash.
- ✓ No lecturer clash.
- ✓ No wastage of time due to room finding and class timing while Scheduling.
- ✓ To use fast, attractive and modern service.
- ✓ Students and teacher can access their class easily with respect to room and time.
- ✓ The timing of class will be effective.
- ✓ For registrar workers it minimize work load.
- ✓ To make the time schedule more reliable and flexible by considering changes, registrar employees and students.

1.7 Methodology

1.7.1Data collecting methods

Data gathering is one of the necessities in system development .These enables developers filled with true information and identify the problem easily to propose appropriate solution. Our project is based on the case study (techniques)

To get the necessary data for the effective development of the system the developer team used different gathering methodologies in different context among the different methodologies the following are used:

- ✓ Interviews
- ✓ Observation
- ✓ Document analysis

Interviewing: We used interview in a planned meeting during which we can obtain information from another concerned person. As it is known the personal interview is often the preferred information gathering technique when developing business and user requirements.

We interviewed 10 registrar workers, 5 department heads, 10 room managers, 20 instructors. During which we can contextualise the response by observing body language. Since body language is all of the non-verbal information being communicated by an individual. Part of body language is facial disclosure. Facial disclosure can sometimes enable us to understand how people feel by watching the expressions on their faces. Many common emotions have easily recognizable facial expressions.

We used interview method to gather direct information from users and administrators regarding the procedures of schedule preparation and filing system, announcement, and etc.

Observation: We used Observation technique that enables the analyst to view how processes and activities are being done in the context of the organization. This additional perspective can give a better understanding of procedures of the system. It is sometimes worthwhile to read procedure manuals to find out how things should be done. The developer of the system tries to observe the actual operations of the existing system in Welkite University.

Document analysis: Documents analysis help as for searching, finding and extracting information from documents which have been created and used by the University which we are trying to automate.This method of collecting a data is analyzing the document prepared like forms and report samples of the existing system.

1.7.2 System analysis and design approach

System development methodology is the standard process followed in an organization to conduct all steps necessary to analyze, design, implement and maintain the information system. Here this study uses the common methodology for the system development that is object oriented engineering approach method for planning and management of the system development process**.**

The Systems Development Life Cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development from an initial feasibility study through maintenance of the completed application. Actually, SDLC is the traditional methodology which is used to develop, maintain and replace information system. SDLC is a process of gradual refinement meaning that it is done through several development phases. Each phase continues and refines what's done in the previous phase.

The 5 steps of Software Development Life Cycle (SDLC) process used here are planning & analysis, designing, implementation, testing, and maintenance/ supporting.

1.8 Development Tools

Software tools

The tools we will use are:-

- ✓ NetBeans IDE 8.0 using Java server faces 1.8 JDK for Java programming because of our familiarity and it is best for object oriented approach.
- ✓ Glass Fish Server to run and deploy the system web interface
- ✓ Microsoft project to schedule our tasks.
- ✓ My SQL server because it is the best tool for database designing.
- ✓ Microsoft Visio 2007 especially for UML.
- ✓ Ms Office 2007 and Liber Office writer to document our project data.

Hard ware tools

In doing this project there are different hardware tools that are necessary from the beginning until the end this includes:

- ✓ Computers
- ✓ Flash disk
- ✓ CD/DVD RW

1.9 Feasibility of the Project

The aims of a feasibility study are to find out whether the system is worth implementing and if it can be implemented, given the existing budget and schedule. The input to the feasibility study is a set of preliminary requirements, an outline description of the system and how the system is intended to support business processes. The results of the feasibility study should be a report that recommends whether or not it is worth carrying on with the requirements engineering and system development process.

Depending on the result from statement of the problem the feasibility of this project is reviewed by feasibility measuring parameters such as technical, economical (financial) schedule. These parameters help to decide the proposed project continuation according to its workability, impact on the WKU (Wolkite University), ability to meet the user needs and effective use of resources.

1.9.1 Technical Feasibility

Technical feasibility is a study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not. This system is to be utilized by any users of the organization as they are authenticated to use. The developer team have better knowledge on developing system and give the necessary support to user starting from the development to the maintenance of the system if deployed.

1.9.2 Economic Feasibility

Economic feasibility is measure of cost effectiveness of the project. As per our cost benefit analysis the existing system is not feasible with different perspective so we develop a system which reduce the current cost and improve efficiency.

As per our analysis of the cost benefit analysis, the existing system is not feasible with different perspective, so we developed a system which reduces the current cost and improve efficiency. .For instance benefits from the system that area unquantifiable is

- ✓ Better decision making
- ✓ Better service to users
- ✓ Little job burden to employees

1.9.3 Operational Feasibility

Operational feasibility is the measure of how well the proposed solution of the problems will work in the organization context**.** The schedule management system which we develop is operationally feasible. The user will be giving training on the deployment of the system and familiarize them with the current technologies in hardware and software in that it improve the current operation of the University with great cooperation of the project team.

1.9.4 Political Feasibility

Political feasibility analysis is used to predict the probable outcome of a proposed solution to a policy problem through examining the actors, events and environment involved in all stages of the policy-making process. It is one frequently used component of a policy analysis and can serve as an evaluative criterion in choosing between policy alternatives. Feasible policies must be politically acceptable or at least not unacceptable. Political unacceptability is a combination of two conditions too much opposition or too little support. One common mistake is widespread in practice that feasibility becomes a dominant criterion of preferable alternative.

Policies are implemented by governments, and governments are subject to a variety of political constraints. Within a democratic electoral system governments are unavoidably concerned about the effects of various policies on the electoral blocks upon which they depend; so it is difficult for a government based on the support of rice farmers to abolish input subsidies. But there are political constraints within authoritarian regimes as well. Such regimes rarely have sufficient power internally to dominate the whole of civil society, but depend rather on the support-explicit or implicit-of various interest groups within society. Once again, it is difficult or impossible for such a government to implement policies that are inimical to the interests of the groups upon which it depends.

In this context we may distinguish between varying levels of state autonomy with respect to powerful elites. In some instances a powerful group-for example, a landholding elite-may be in a position to simply determine state policy-through its dominant presence in a legislature, its ability to capture the presidency, or through other familiar mechanisms. In a wider range of cases, however, powerful elites will be in a position to block various

1.9.5 Schedule Feasibility

Schedule Feasibility is defined as the probability of a project to be completed within its scheduled time limits, by a planned due date. If a project has a high probability to be completed on-time, then

its schedule feasibility is appraised as high. In many cases a project will be unsuccessful if it takes longer than it was estimated: some external environmental conditions may change, hence a project can lose its benefits, expediency and profitability. If a work to be accomplished at a project does not fit the timeframes demanded by its customers, then a schedule is unfeasible (amount of work should be reduced or other schedule compression methods applied).

If the project managers want to see their projects completed before they can lose their utility, they (project managers) need to give proper attention to controlling their schedule feasibility: to calculate and continually reexamine whether it is possible to complete all amount and scope of work lying ahead, utilizing the given amount of resources, within required period of time. Schedule feasibility study includes use of the following matters

1.10. Plan

1.10.1 Budget Plan

The following table shows of the items list and their associated cost which are important for the completion of this project.

No	**Item**	**Total(Eth birr)**
1	**Stationery**	
	a. Printing	450.00
	b. Paper	200.00
	c. Binding	200.00
	d. Pen, Pencil	100.00
	e. Copying	500.00
	f. CD/DVD	100.00
	g. Note Book	50.00
2	**Transportation**	2,000.00
3	**Per diem**	3,000.00
4	**Communication(mobile air time fee, CDMA)**	1,500.00
5	**External Storage Device**	4,000.00
7	**Miscellaneous**	500.00
8	**Contingency**	1,160.00
9	**Total Cost**	15,760.00

Table 1.1 budget plan

1.10.2 Communication Plan

Project Communication Plan within our project is described in table 1.2

What	Who/Target	Purpose	When/Frequency	Type/Method(s)
Initiation Meeting	All stakeholders*	Gather information for Initiation Plan	First Before Project Start Date	Meeting
Team Meetings	Entire Project Team. Individual meetings for sub-teams, technical team, and Functional teams as appropriate.	To review detailed plans and to Work the project (tasks, assignments, and action items).	Regularly Scheduled. Weekly is recommended for entire team. Weekly or bi-weekly for sub-teams as appropriate.	Meeting Detailed Plan
Project Advisory and Group Meetings	Project Advisor Group and Project Manager	Update Project Advisory Group on status and discuss critical issues. Work through issues and change requests here before escalating to the Sponsor(s).	Regularly Scheduled. In 2 week is recommended.	Meeting
Sponsor Meetings	Sponsor(s) and Project Manager	Update Sponsor(s) on status and discuss critical issues. Seek approval for changes to Project Plan.	Regularly scheduled Recommended biweekly or monthly and also as needed when issues cannot be resolved or changes need to be made to Project Plan.	Meeting
Others Team Meeting	Team Members (not All)	Share Ideas about projects		Demonstration

Table 1.2 communication plan

1.3 Time Plan

	Task Name	Duration	Start	Finish
1	Proposal	5 days	Mon 3/16/15	Fri 3/20/15
2	Requirement Analysis	10 days	Mon 3/23/15	Thu 4/2/15
3	System Design	15 days	Fri 4/3/15	Wed 4/22/15
4	Implementation	25 days	Thu 4/23/15	Wed 5/27/15
5	Testeing	4 days	Thu 5/28/15	Mon 6/1/15
6	Conclusion	4 days	Tue 6/2/15	Fri 6/5/15

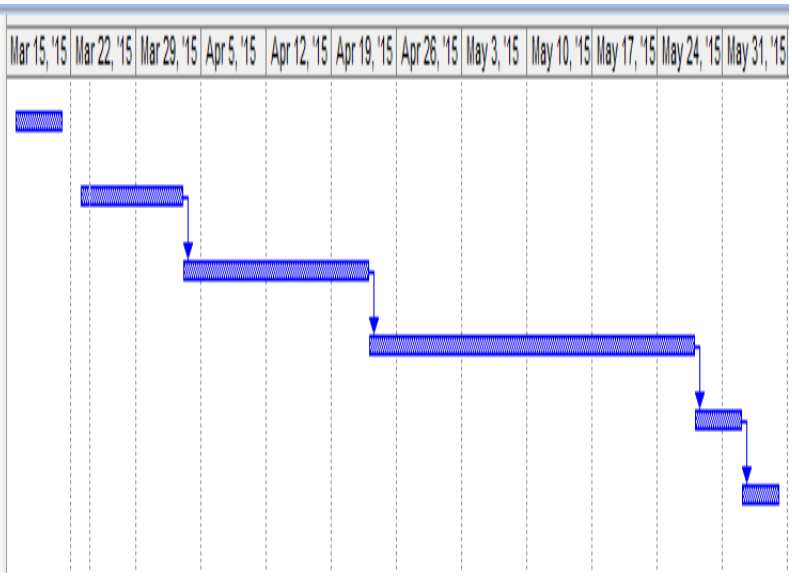

Table 1.3 Time Plan

CHAPTER TWO

2. LITERATURE REVIEW

2.1 Theoretical Background

Automated SMS plays a great role in simplifying the job of employees at the school and satisfying the need of customers and stakeholders of the school. Even though no documentation is found in Ethiopia to be reviewed, products have been observed at some schools to help understand the problem of managing schools and handling school data. This chapter reviews these products.

2.2 Observed Products

In the year 2003 City Government of Addis Ababa Education Bureau (CGAAEB) was very much interested to have automated school management system to get uniform and quick access to the students' data for administrative purpose on promoting the students' achievement and related issues. The bureau has selected Wondrad Preparatory School for pilot test. At the time the school principals together with officials from CGAAEB signed a contractual agreement with some software developer company. The developers installed their first version of the product which can register a student offline and generate official transcript with some level of difficulty. As the system is not fully automated, it does not support management of attendance, does not support generating report cards and other important functions such as generating school timetable and a web based report for parents. Due to the lack of follow up by the government officials at CGAAEB, the company was unable to complete the project. The school currently is unable to use the partially developed system because of lack of trained person and lack of hardware and software maintenance.

Another product that is in use is transcript generator system. The transcript generator system at Menelik II Preparatory School generates official transcript of students. In order to generate transcript the record officer enters the student information along with the grade marks for the grades completed per year and per semester. Then the system generates the required official transcript. Currently the school is using the system to generate official transcript even though the data entry format has unnecessarily many fields which are not applicable for the record office but can be used for continuous assessment by the course teacher.

2.3 Manual Timetabling

Manual timetables are prepared by dedicated teachers. In manual timetabling, it is common to proceed in an iterative fashion where each iteration selects and schedules a lesson [3]. Scheduling a lesson requires to choose a classroom (fixed for each section of students) and a time slot such that the commitment to the choice will not violate any constraint.

In school timetabling, we are required to schedule a given set of meetings such that the resulting timetables are feasible and acceptable to all people involved. Humans are able to prepare the timetable using some hit/miss approach. So it is possible to automate the timetable based on a simulation of the human way of solving the problem. Such techniques, that we call direct heuristics, were based on a successive augmentation. That is, a partial timetable is extended, lecture by lecture, until all lectures have been scheduled. The underlying idea of all approaches is to schedule the most constrained lecture first.

Usually some responsible teachers are assigned to schedule subjects and teachers. The number of teachers available per each subject is predefined and the load that each teacher has is calculated. With these data the timetable constructor assigns each teacher-subject association to the appropriate classes with the available time slots.

The manual solution of the timetabling problem usually requires many person-days of work. In addition, the solution obtained may be unsatisfactory. The lessons should be fairly distributed satisfying the identified constraints.

2.4 Genetic Algorithms

Genetic algorithms are a solution technique for optimization problems (Davis, 1991; Michalewicz, 1994). Differently from tabu search and simulated annealing, they are not based on local search.A genetic algorithm starts with a set of solutions randomly chosen $\{s^0 1,\ldots, s^0 n\}$, which is called the population at time 0. The core procedure is a loop that creates the population $\{s1^{t+1},\ldots,sn^{t+1}\}$ at time t+1 starting from the population at time t. To this aim, the value of the objective function is computed for each solution sit.

Based on a weighted randomization, n elements of the population at time t are selected. Obviously, some solution may be selected more than once. The randomization is biased by the value of the objective function so as to assign a higher probability to be selected to the solutions that result in a better value of the objective function. In this way the best solutions get more copies, and the worse ones probably die off.

At this point each solution is selected for recombination with a given probability (PR). The recombination is done by the crossover operator. That is, two selected solutions are mixed by swapping corresponding segments of their representations. One of the most common ways to do the crossover is by selecting a fixed number of positions in which the swapping takes place (fixed-point crossover).For example, if two solutions are represented by the strings abcdef and uvwxyz and we choose two crossover points after the second and the fifth character, then the new solutions would be abwxyf and uvcdez.

In addition, mutation arbitrarily alters randomly some part of some solutions randomly selected, based on a given probability value (PM). The method terminates either when it generates a fixed number of populations, or when the best solution reaches a certain value of the objective function, or when the algorithm does not make any progress for a certain number of iterations. The main control parameters of the method are the population size n, the probability of crossover PR, and the probability of mutation PM.

2.5 Drawbacks of the Reviewed Systems

The reviews described have the following problems:

- ✓ Generate official transcript with some level of difficulty,
- ✓ Do not sufficiently produce the required reports to allow parents to view status of their children and reports for officials of Kebele and Kifle-Ketema to help them participate in decision making,
- ✓ Do not generate timetable for the schools
- ✓ Do not facilitate attendance record keeping by the home room teachers

From the above four characteristics, huge problems may be settled using the Al Khwarizmi (2009) Genetic as problem parameter will be coded as a fixed length sequence. Al- Khwarizmi Genetic operates directly using only the objective functions without specific additional information. In addition, the use of `database' or population will reduce the incidence of being trapped in local optimum. Finally, by using probability as a method of selection, it will actually assist Al-Khwarizmi in moving freely in the search space, with the intention of obtaining better development. Al-Khwarizmi Genetic may be moved several times using the same parameter, to possibly obtain the best solution.

The twin problems of timetabling and classroom assignment that educational institutions face have been widely studied in the literature (de Werra 1985, Mooney et al. 1996, Schaerf 1999a, Asratian

and de Werra 2002, Lewis 2008). Both problems are classified as NP-hard (Schaerf 1999b, Pongcharoen et al. 2008). Many techniques to solve them are available, including a series of optimization models based primarily on linear and integer linear programming (Daskalaki et al. 2004, Daskalaki and Birbas 2005, MirHassani 2006, Hernandez 2008) and various types of heuristics and metaheuristics (Elmohamed et al. 1998, Rossi-Doria et al_ 2002, Lewis 2008, Pongcharoen et al_ 2008). Many researchers have used specificheuristics such as tabu search (Alvarez-Valdes et al.2002), simulated annealing (Elmohamed et al. 1998), ant colony algorithms (Socha et al. 2003), genetic algorithms (Wang 2003), multineighborhood local search (Gaspero and Schaerf 2003), multiagent methods (Oprea 2007, Strnad and Guid 2007), and hyperheuristics (Burke et al. 2003, 2007; Rattadilok et al. 2005). Other approaches use hybrid heuristics that combine two or more of the above techniques (White and Zhang 1998, Chiarandini et al. 2006) or offer designs built around constraint programming (Goltz and Matzke 1999, Abdennadher and Marte 2000, Rudova and Murray 2003, de Werra, 1985) describes a series of problems that arise in the mathematical modeling of course scheduling, instructor assignment, and examination timetabling. By adopting a theoretical perspective, de Werra's paper lays the basis for developing future applications that address similar problems and actual study cases.

Some researchers have systematized their solution approaches and implemented computational systems to support decision makers at specific educational institutions. (Bloomfield and McSharry, 1979) propose a system that uses a heuristic procedure to solve a course scheduling problem. They discuss its low implementation costs and significantly reduced execution times. Whereas the equivalent manual process took three weeks to deliver a solution, their system took two days. Ten years later, Chahal and de Werra (1989) implemented a system on a PC to resolve a course scheduling problem at an adult training school. They solve their network-flow model using a heuristic procedure that generates initial solutions in a matter of minutes

Johnson (1993) implemented a computational course scheduling system at the Lough borough University Business School. The paper describing the system, which allows the user to estimate the effects and consequences of a given scheduling solution, shows that efficient management of course databases leads to better-quality schedules. The schedules generated are not necessarily feasible or optimal; however, Johnson stresses their system's usefulness in permitting greater automation of the processes.

Ferland and Fleurent (1994) present *SHAPIR*, a system that uses heuristic algorithms to solve an IP

model and allows the user to manually adjust the schedules it generates. Three educational institutions of different sizes have tested the system; in each case, the results were good and execution times were reasonable. Stallaert (1997) developed a system to automate course timetabling at UCLA's Anderson School of Management. In this approach, the courses are separated into two sets, core and noncore. The system uses an IP model whose objective function maximizes the schedule preferences of the core-course instructors; it then schedules the noncore courses using a heuristic procedure.

Foulds and Johnson (2000) developed *Slot Manager*, which has a similar structure to our model, for use with a microcomputer. It has three components— interface, database, and model base. Its interface enables a user to enter information, e.g., type of student group, list of classrooms, and course characteristics, into the system using a series of menus. Its database is relational and contains the information that defines the schedules for a given semester or year. Finally, the model base contains the system's "intelligence," which solves instructor-schedule conflicts, classroom-capacity violations, and resource underuse. Sprague and Carlson (1982) originally presented this structure. Hinkin and Thompson (2002) created *Schedule Expert*, a system that also incorporates an IP model that they implemented for course timetabling at Cornell University's School of Hotel Administration. The set of decision variables in the model assigns courses to patterns, which are combinations of class days, time slots, durations, and classroom-equipment requirements.

The type of variable used is similar to the type that we used in our formulation. Their model addressed all the schedule-conflict conditions that Cornell imposed; in addition, the time required to generate the timetables dropped from several weeks to a few hours. Other studies (Carter 2001, Dimopoulou and Miliotis 2001, Alvarez-Valdes et al. 2002, Haaseet al.2004, Oprea 2007) have reported diverse decision support systems that incorporate mathematical programming models for generating course timetables and classroom assignments. In each case, the system contributed numerous improvements to the decision-making process.

The applied runtime system to a class of tiled linear algebra algorithms: Cholesky factorization that are introduced in. In the tiled algorithms, each task computes subroutine. The tiled algorithms can fully utilize the operations such that the cache hit rate is maximized and data movement is minimized. Our practical analysis shows that the algorithms have a provably good expected execution time. We conducted experiments on both shared and distributed machines. Our project results demonstrate that the decentralized task scheduling approach is efficient and scalable.

The paper we have designed runtime system to schedule tasks dynamically on both shared and distributed systems. Linear algebra programs are written with a task-based library and can be executed by the local system automatically. We mainly focus our system design on the scalability metric. The paper has the following contributions: (i) proposes a task-based library to support dynamic scheduling of linear algebra algorithms automatically; (ii) presents a distributed algorithm to resolve scheduling dependences without external process communication; (iii) demonstrates the scalability and practicality of the dynamic scheduling approach on both shared and distributed systems; (iv) proves that class schedules linear algebra algorithms have sufficient degree of parallelism and can offer scalability guarantees. The current system prototype does not optimize schedule. For small departments, we plan to add new features to the system to improve the scheduling like randomization of schedule affinity as well as large departments. Another research beyond this line is to study the dynamic scheduling approach on organizational architectures.

Accordingly, having in mind that automated scheduling have very good opportunities for educational institutions and the experience of other developing countries it will be unquestionable to apply the same technology in other areas also. As all have been said, a study should be made on education workers (*i.e instructors, lab attendants*) to examine the schedule time condition opportunities to apply and see the challenges as well.

CHAPTER THREE

3. REQUIREMENT ANALYSIS

3.1. Current system

In this section we reviewed the current system followed by a brief description of the problems associated with the current system and a proposed system that aim to solve these problems The Current System of the Wolkite University timetable scheduling is manual work .The registrar worker collect the data from Departments about their instructors, courses, sections and also get the data about class room available for learning process. Based on thus data the class scheduling process is stared with first test schedule by providing the time for each department section, course and instructors within room .After that try to avoid any collision that occurs on the period next the revised or temporary schedule is printout, it is temporary available until the problem of period collision get solution and it is ready to accept any complain comes from the Instructors. If any complain is come the new temporary revised time table schedule will be post.

The main issue of current system in regarding with input interaction from different stakeholder (actors) is manual work. The registrar workers collect data such as instructors, courses sections as input just spurting across different departments. Existing system have no more option to engage other department rather than registrar. Means a single department did not collect data and working with the existing system scheduling process does not process further more scheduling process. Enhancing the availability of processed data for other use is too difficult.

The main problem of the existing system is class overlapping due to Scheduling process single class may have more than one period assigned at the same time. Means two or more courses assign at one period are class overlapping.

3.2. Proposed System

3.2.1. Overview

The need to develop automated Timetable scheduling system of the Wolkite University is that the current activities of the university are time consuming due to manual system which results delay on the delivery of course on time.

The automated system is developed with the aim to overcome the drawbacks of existing manual system. The proposed system will have many advantages.

- ✓ This system automates all data on the web to the user gives information.
- ✓ To keep database up to date by regular updating.

- ✓ To provide various report facility.
- ✓ Cost effective and less manpower required.
- ✓ Login authentication
- ✓ The system can generate to create new time table based on the name that the administrator specify for it.

3.2.2. Functional Requirement

The functional requirements are features that the system must include to satisfy the users need, some of the functional requirements of the system are:

- ✓ Login authentication
- ✓ The System can show free periods and display based on the inputs given about lecturer, room, and Section of class.
- ✓ The system should allow modifying the class schedule.
- ✓ The System should generate randomly schedule for the selected Section of class.
- ✓ The System should generate reports the final result of Class.
- ✓ The System should display the timetable on the web application

3.2.3. Nonfunctional requirements

3.2.3.1 User interface and human requirements

The system provides web based application user interfaces that are compatible with most platforms. When we consider the user interface, since there is going to be different type of users it should have different type of user interfaces. Some of the nonfunctional requirements for the system are described as follows.

3.2.3.2 Security issue

The system should be protected with different security features (techniques) on both the software and hardware part. On the software part, any user who wants to login to the system must have a user name and password. This protects the system from any unauthorized access. The administrator in any section to access and/or modify the data related to that section.
Generally, the system should be secure to a level that, even when it is available Web, critical information should not be available to non-authorized users. On the hardware part, the Server is expected to be placed in a secured room. The room should establish physical access procedures to ensure only system administrators obtain physical access. In addition, the system should be in session controlled way to permit URL based system access.

3.2.3.2 Performance characteristics

The system is accessible by specified actors in the university. It should be given more emphasis for the speed to access it.

3.2.3.2.1 Error handling

The system is expected to handle errors encountered during run time. Errors could rise from users and from the system. Errors that occurred from the wrong doing of users will be handled by appropriate exception handling mechanisms. Moreover, in case of the system's database failure, back up database will be used to resume normal operation. Generally, if an error occurs, the system will identify the error and notify the user so that he/she can take the appropriate corrections. When we say this, we are not saying that all errors will be handled with notifications on how to correct them.

3.2.3.3 Quality issue:

Information in database should be accurate and updated. When additional information comes through new semester add batch, staff member, add courses, rooms, school establish academic year.

3.2.3.3.1 The system modifications

The system should be easily modifiable. System modification is based on the interest and the need of the users as well as the developers. As of the users' comment (feedback), the system will be modified to meet their requirement. In fact, the developers or any trained person that knows the code behind the system does this modification. Moreover, modifications can be achieved without a problem since the system is designed using an object oriented approach.

3.3. Hardware and Software Requirement

3.3.1 Hardware Requirements:

The University should have the following hardware components for efficient use of the proposed system

- ✓ Personal computers having required storage capacity and processing speed.
- ✓ Network devices such as cable, Switch and different connectors as per the standard.
- ✓ Backup devices and AC in the server room.

3.3.1 Software requirements

The university should have server software for the deployment of the system. In addition, it should have Up-to-date web browser, Glass Fish Server for hosting the system locally, anti-spy wares in order to protect the database from spy and others for the effective implementation of the system.

My sql server to deploy the database.

3.4. Software Engineering Methods and Techniques

We use object oriented methodology to develop our system. Because it is a popular technical approach to analyzing, designing an application system, or business by applying the object-oriented paradigm and visual modeling throughout the development life cycles to foster better stakeholder communication and software product quality. Object Oriented Analysis is selected since it has so many advantages and which can make our system more effective.

The advantages of Object Oriented Analysis are:

- ✓ The ability to tackle more achieving problem domains.
- ✓ Increased consistency among analysis, design and programming activities.
- ✓ Common design and coding methodologies, including Object-Oriented Design (OOD), Design Patterns, Refactoring, and the Unified Modeling Language (UML)
- ✓ Explicit representation of commonality among system components.
- ✓ Increased consistency among all the models developed during Object Oriented Analysis, design and implementation.

But for better implication of requirements of the new system analysis and design methods are used in addition with the object oriented paradigm.

3.1 Use case diagrams

An actor specifies a role played by a user or any other system that interacts with the subject.

Primary Actor: The primary actor of a use case is the stakeholder that calls on the system to deliver one of its services. It has a goal with respect to the system – one that can be satisfied by its operation. The primary actor is often, but not always, the actor who triggers the use case.

Supporting Actors: A supporting actor in a use case in an external actor that provides a service to the system under design. It might be a high-speed printer, a web service, or humans that have to do some research and get back to us.

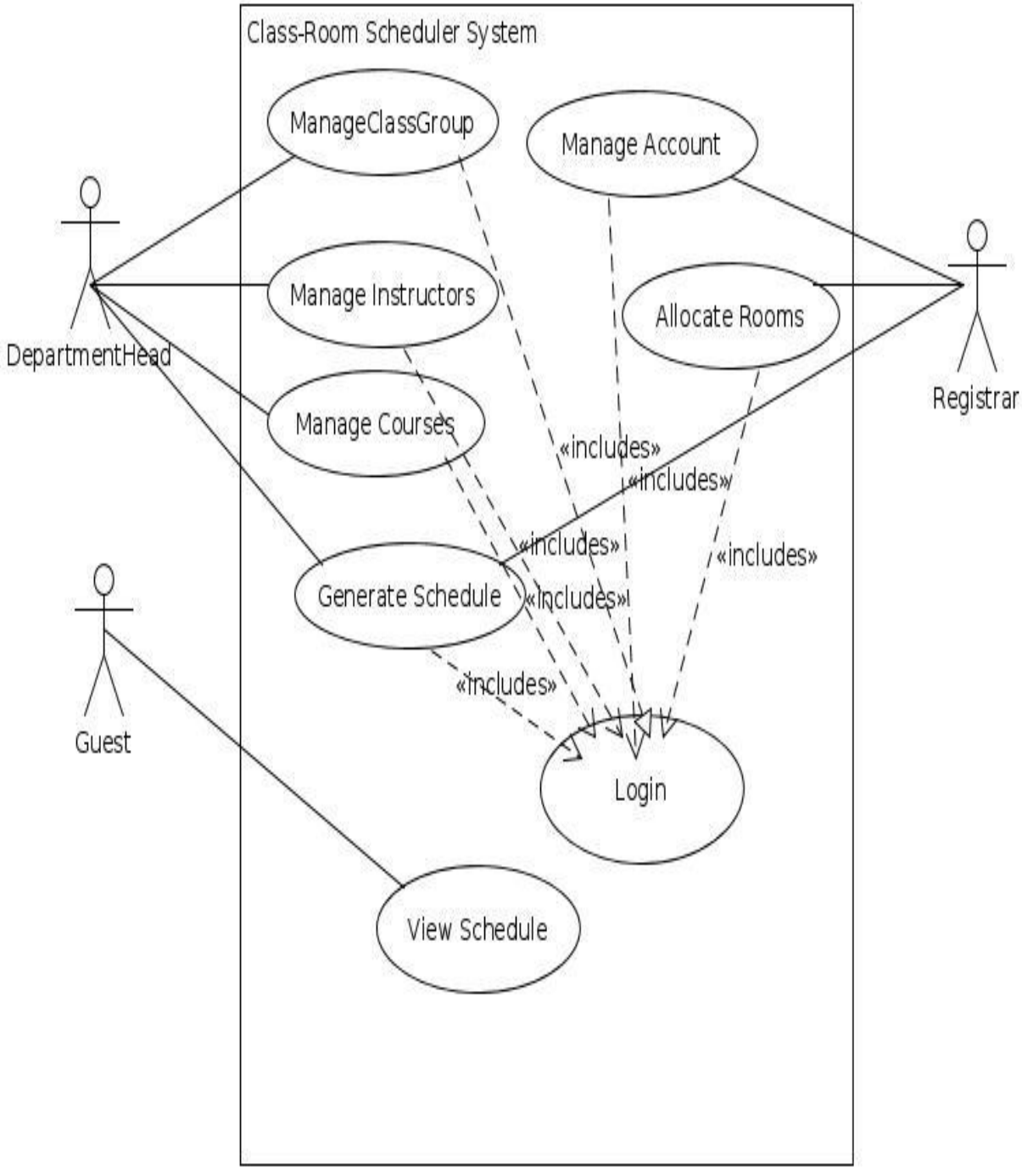

Fig 3.1 System Use case modeling

The main actors of the systems are registrar, department and guest.

Registrar activates are manage account, allocate room and generate schedule.

Department activities are mange instructor, mange course, generate scedule and manage class group. Guest activities are view seclude.

3.1.1 Use case descriptions

1. Use case description for: **Managing class group**

Use Case Name	Managing class group
Use Case ID	UC-1
Description	Manage department of batch and section groups
Actors	Department, Registrar
Basic Flow	✓ The Registrar creates departments ✓ The department head prepare the list of group and batch ✓ The department head start manage for each section group
Alternative Course of Action	
Precondition	The Registrar should create department.
Post Condition	Each group of the department include in the schedule

Table 3.1 managing class group table

2. Use case description for: **Managing Course**

Use Case Name	Managing Course
Use Description Case ID	UC-2
	The use case deals with managing and offering courses of department for the batch.
Actors	Department
Alternative Course of Action	
Precondition	The department head must be login The Department should create batch of department and courses of department
Post Condition	Each course will include in the schedule.

Table 3.1.2 managing course table

3. Use case description for: **Manage instructor**

Use case name	Manage instructor
Use Case ID	UC-3
Description	The Department Head manages instructor that are found in the department and their course
Actors	Department
Basic Flow	✓ The Department head prepare the list of the staff. ✓ The department head selects course offering to section of batch and assigns an instructor in to it.
Alternative Course of Action	If the department head doesn't assign instructor for sections : The system will assign default instructor, which is TBA .
Precondition	The department head must be login The department head should create instructor data. The department head should assign the course for each of batch.
Post Condition	Each Instructors of the department will get optimal schedule.

Table 3.1.3 managing instructor table

4. Use case description for: **allocate rooms & Blocks**

Use Case Name	Allocate Rooms
Use Case ID	UC-4
Description	Deals with allocate & providing Available Rooms and Blocks for each Department of section group.
Actors	Registrar
Basic Flow	Started by a Registrar by creating rooms. The registrar worker assign each room for each group of section in each department.
Alternative Course of Action	
Precondition	The Registrar must be login The Department should create a list of section of batch.
Post Condition	Every available room include in the schedule according to their functionality

Table 3.1.4 allocate room table

5. Use case description for: **Class Schedule Process**

Use Case Name	Generate Schedule
Use Case ID	UC-5
Description	Deals with generate class schedule for section group of department.
Actors	Department head, Registrar worker
Basic Flow	✓ Started by selecting one section of group from the list ✓ perform automatic function
Alternative Course of Action	If the section group does not have course and If the section group don't have a room allocate The system will notify warring.
Precondition	The department head or registrar must be login All the data required for scheduling purpose has to be entered.
Post Condition	All data get to be scheduled.

Table 3.1.5 schedule process table

6. Use case description for**: View Class Schedule**

Use Case Name	View Schedule
Use Case ID	UC-6
Description	Deals with viewing schedule of rooms, instructor and section of group
Actors	Guest
Basic Flow	✓ Started by a guest by selecting one room or instructor or section group from the list. ✓ The system should display the schedule of the selected object.
Alternative Course of Action	
Precondition	The Department head or registrar should finish schedule process.
Post Condition	All schedule ready to view by in any platform.

Table 3.1.6 view and generate report table

3.5.3 Class Diagram

The relationship among the classes. It is a logical model, which evolves into a physical model & finally becomes functioning information system. Class diagram evolve into code modules, data objects, & other system components.

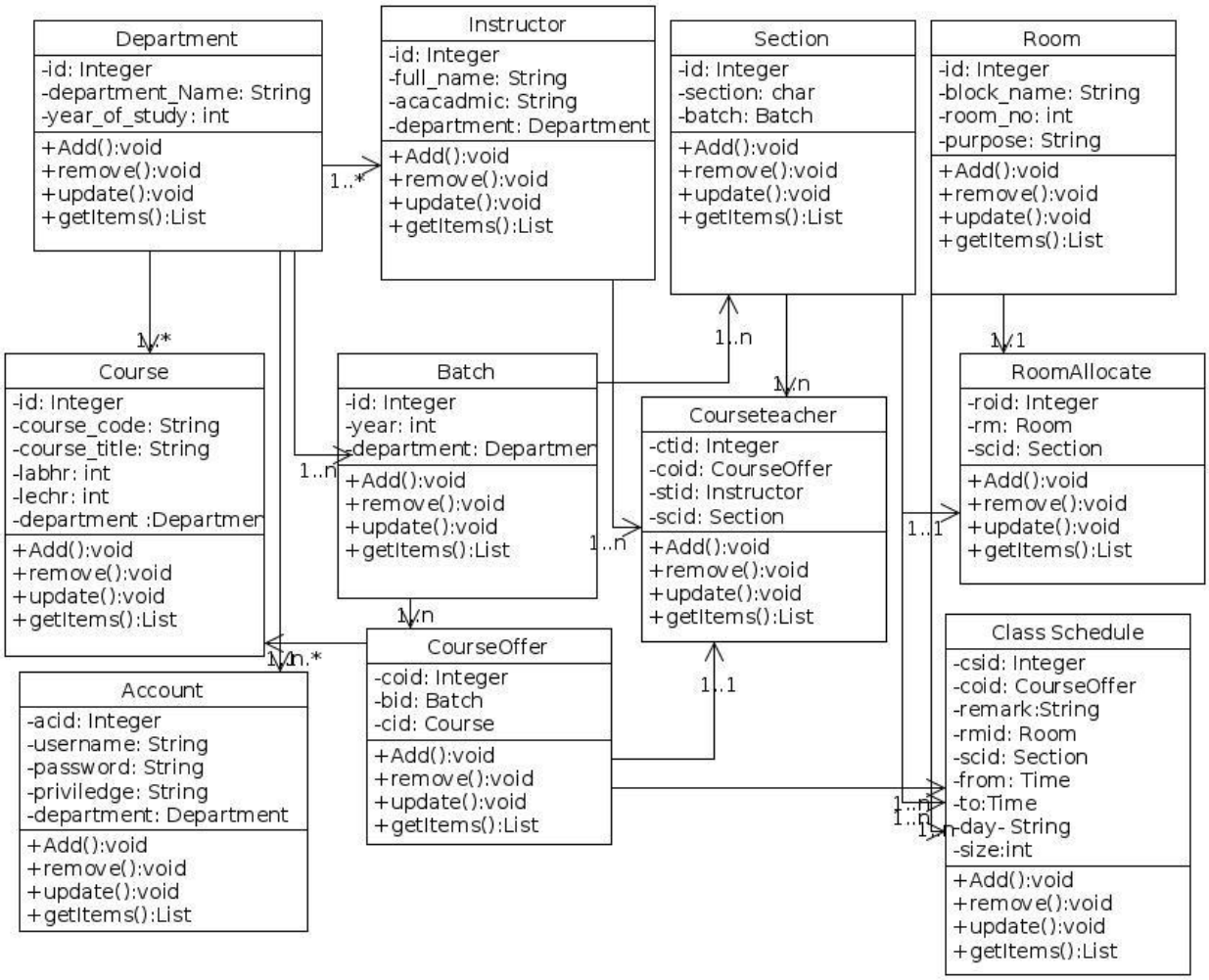

Fig 3.2 Class Diagram

3.5.4. Sequence Diagrams

While there is the assumption that Sequence Diagrams were made for developers, the truth is that a school's business staff could use such diagrams to communicate how exactly the business presently currently works by illustrating how the different business objects interact. Sequence diagrams display the time sequence of the objects participating in the interaction.

3.5.4.1. Sequence diagram for announce rooms and blocks

This diagram shows the interaction between the registrar worker and the institute to insert the available rooms into the system.

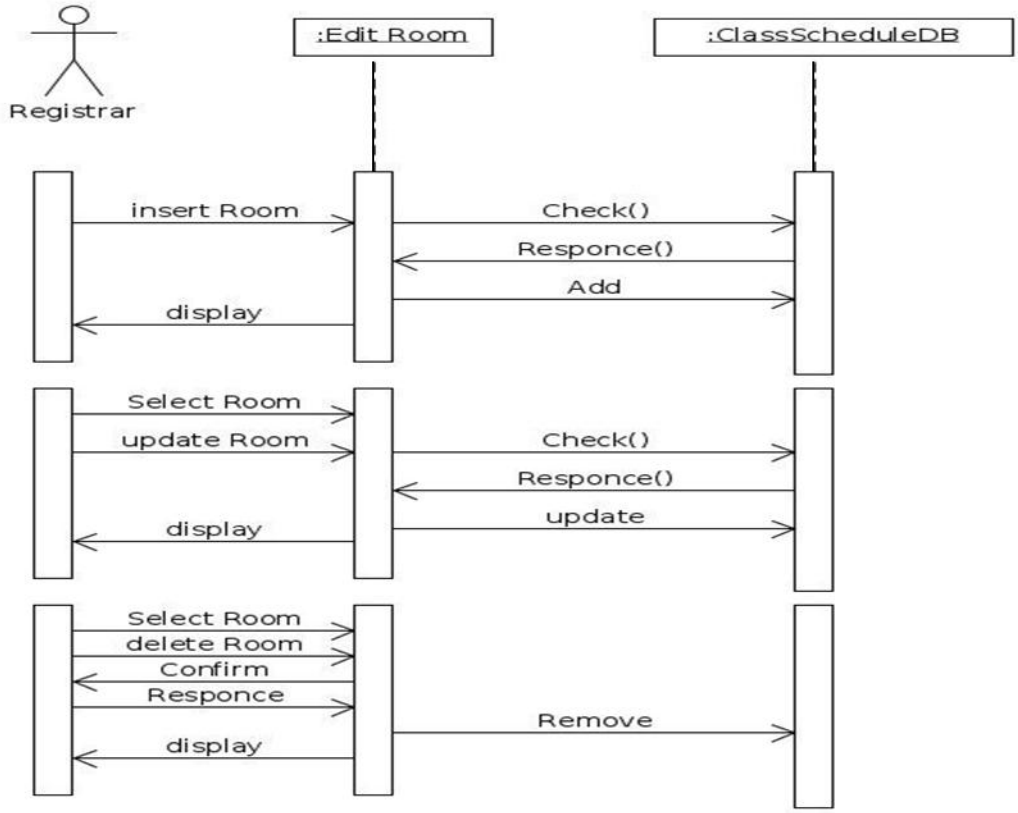

Fig.3.3 Edit Room

3.5.4.2. Sequence diagram to Manage Sections, Instructor and Courses

This diagram shows the flow of data between departments and Registrar worker to insert the data that includes Section, course and instructor into the database.

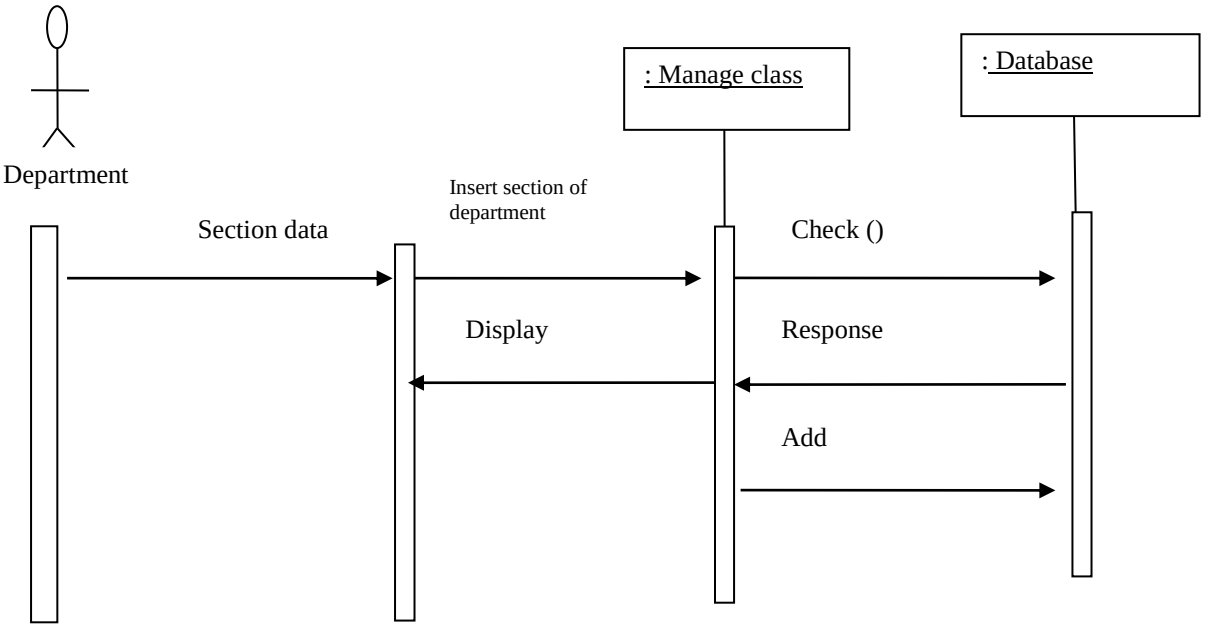

Fig3.2. Manage class

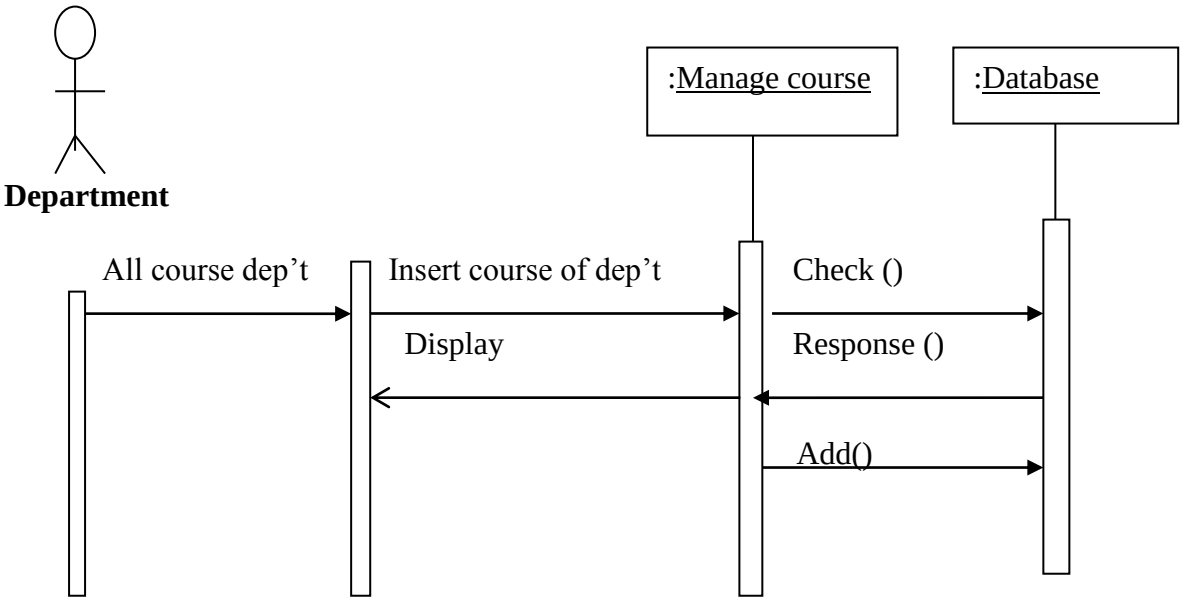

Fig3.3 Manage course

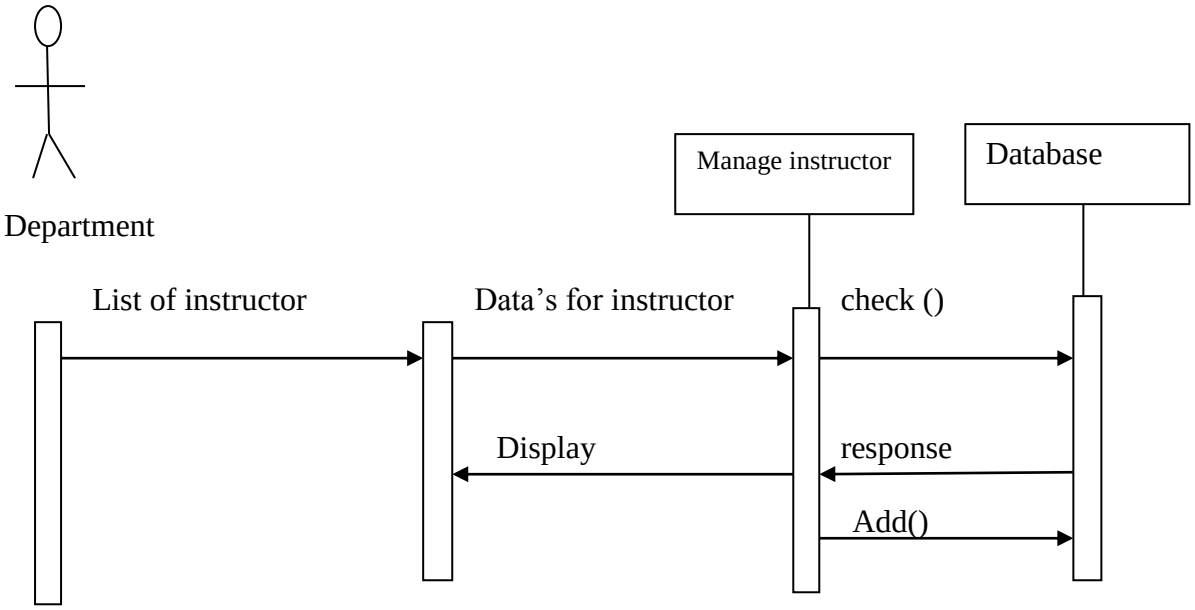

Fig3.4 manage instructor

5.4.3 Sequence diagram for Schedule Process

The following sequence diagram shows how to schedule class timetable, Database is needed for fetching data that have been data stored of each departments section, course and instructors also it is important to handle the collision occurrence for students and other data by referring the schedule weather they have class at the same time or not. If everything is good it saves to schedule table database.

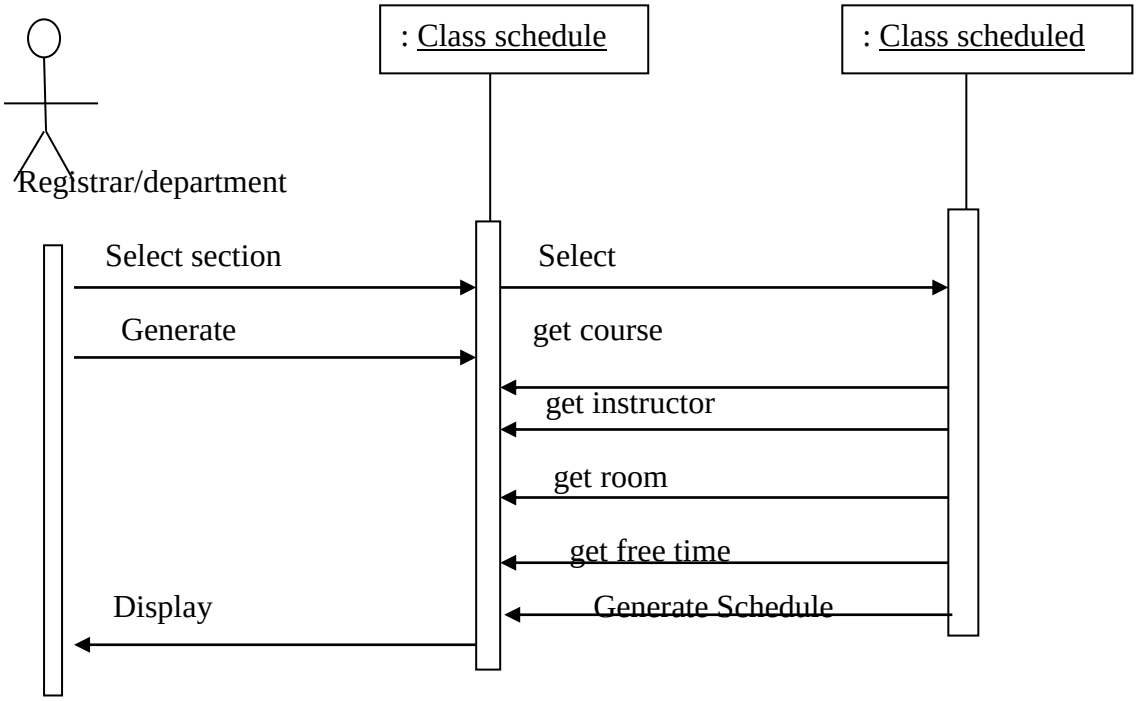

Fig.3.5 schedule

5.4.4 Sequence Diagram for User Account

The following sequence diagram shows how to login and request in timetable scheduling system database.

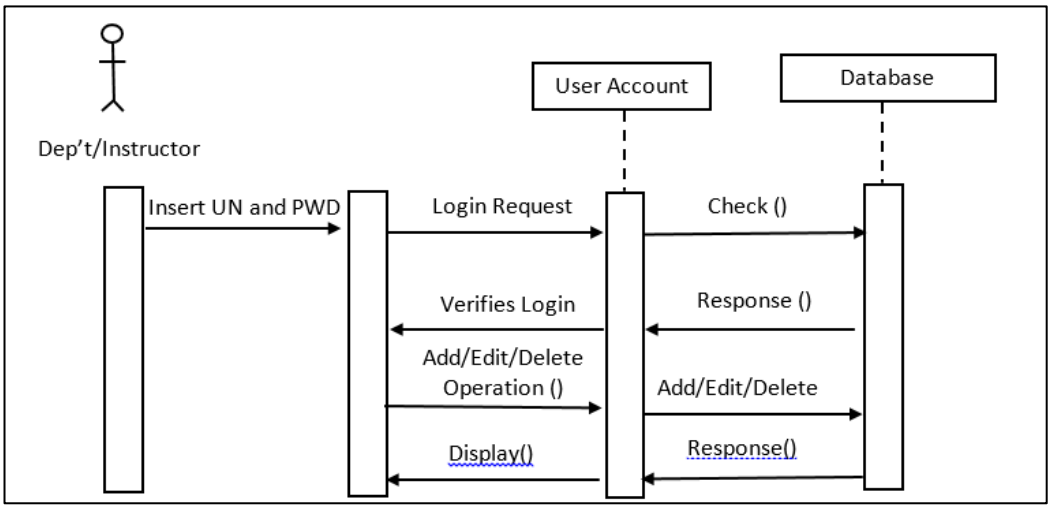

Fig. 3.6 user account

5.4.5 Sequence Diagram for View Schedule

The following sequence diagram shows to view the schedule

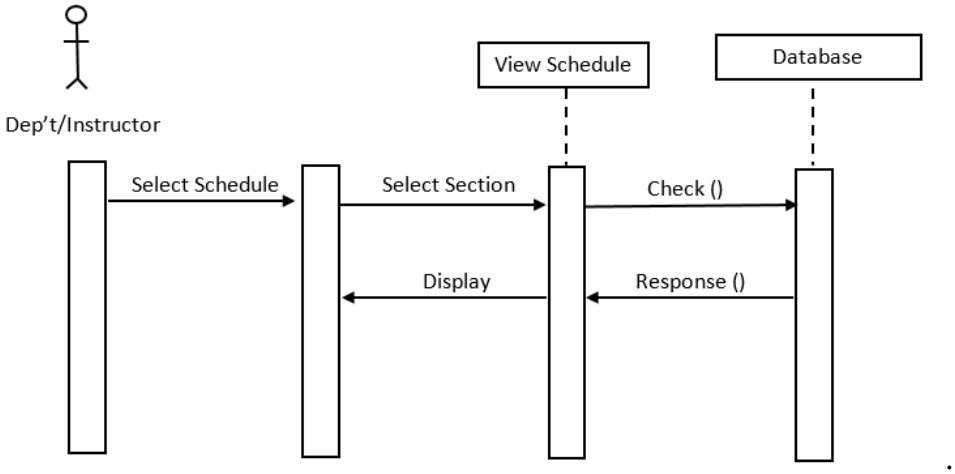

Fig.3.6 view schedule

5.4.6 state digram

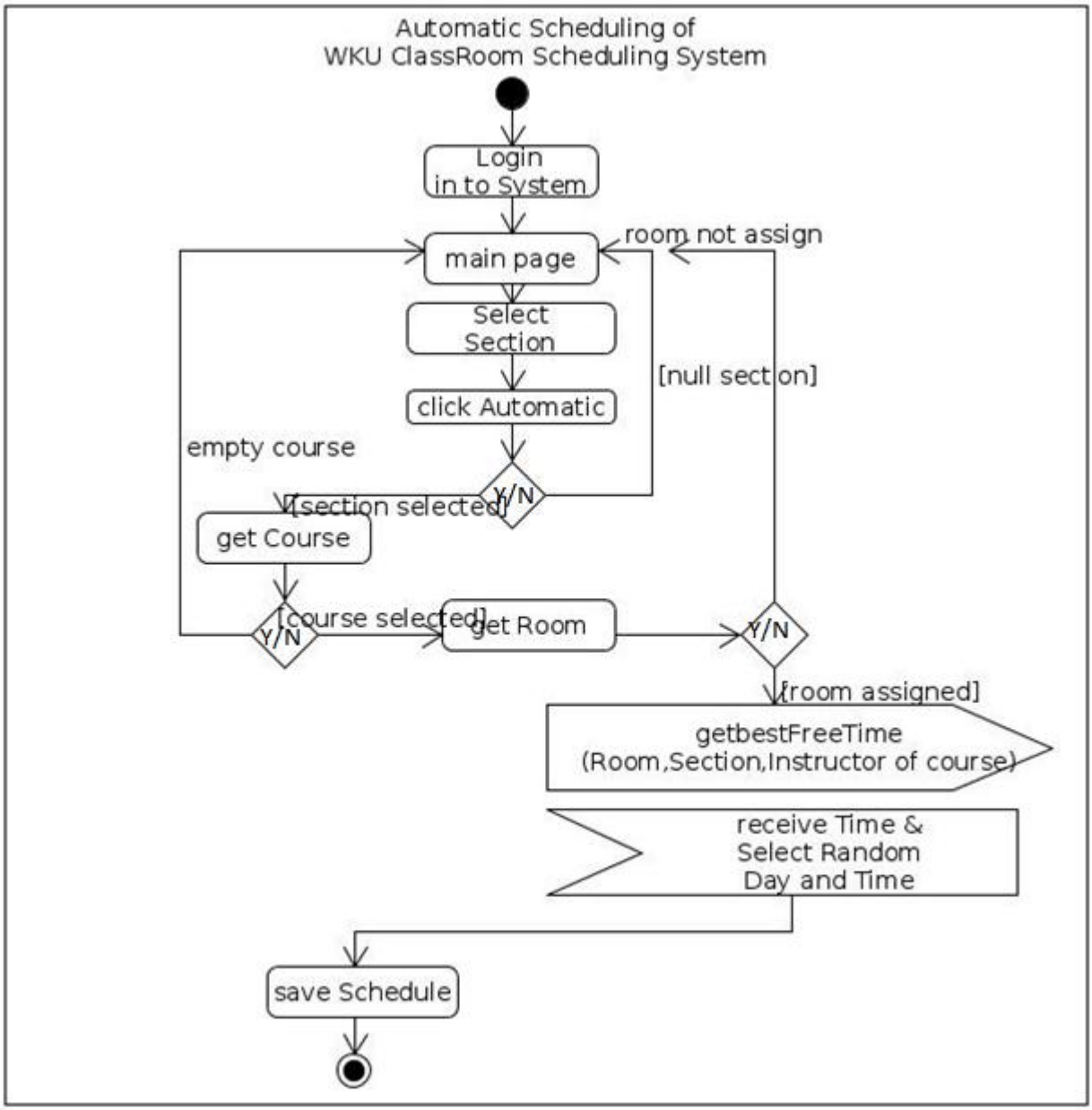

Fig3.7 state schedule diagram

5.4.7 Activity Diagram

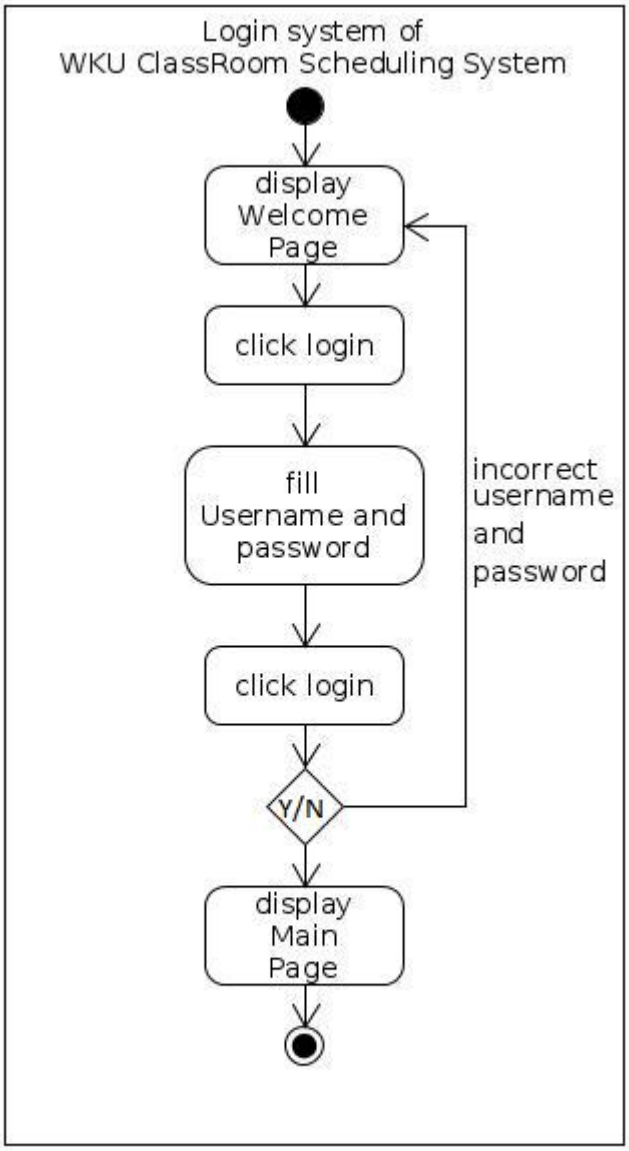

Fig 3.8 activity diagram

CHAPTER FOUR

4. SYSTEM DESIGN

4.1 Introduction

This chapter presents the Time table framework design for Wolikite University. Particularly, we discuss about the proposed Time table framework in detail with its components.

4.1.1 Purpose of the system

The purpose of this system is to design Time table framework for Wolkite University and integrate instructor, Department, and class group section modules based on the components of the proposed Time table framework.

4.1.2 Design Goal

Design goals describe the qualities of the system that developers should optimize. Such goals are normally derived from the non-functional requirements of the system.

Design goals are grouped into five categories. These are:

- ✓ Security
- ✓ Extensibility
- ✓ Reliability
- ✓ Maintainability
- ✓ End-user criteria

4.1.2.1 Security

Only those who have administrative privileges should access the system and any unauthorized access to the system should be restricted through using user name and password. Moreover, the data stored in the database should also be kept secure. The system uses group level security to protect unauthorized personnel from accessing the system. The system security level privilege are defined in the system those are the user account granted to access the system administrator, the user account granted to access specific module and the user account granted to access related module for report purpose.

4.1.2.2 Extensibility

Extensibility refers the ability to add new functionality without requiring major changes to the existing code. This project shall be extensible the new features or any functionality can be easily

added.

4.1.2.3 Reliability

The system shall be reliable it should be validate all input data if the user enter correct information the system allow the user to access the service , if not the system should be display an error message and should order the user to use the correct information.

4.1.2.4 Maintainability

The system should be designed in such a way that it can be easily maintained by the system developer or any authorized professional. In addition, adding new functionality or modifying some parts of the system will not affect the other parts of the system. Because at the design level the system allowed scalability which is open for modifying sub system

4.1.2.5 End User Criteria

Usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. The system should have simple and understandable graphical user interface such as forms and buttons which have descriptive names. It should give reliable response for each user request.

4.1.3 Definitions, acronyms, and abbreviations

Button: one of the GUI components used to trig an action.

Case Study:study everything in the system.

Combo Box:GUI components used to select items from drop down options.

CRC: Class-Responsibility-Collaboration. A technique described in object oriented software by Wires-Brock.

Database: such system that can manage large structured sets of persistent data.

Developer: a person who write/develop/design a computer system/ software codes.

Domain Modeling (Sec 2.3.4):

GUI: Graphical-User-Interface which programs and files appears as icons and data are displayed in windows (rectangular areas).

Hardware: Mechanical, electrical and electronics components of computer system, as opposed to the various programs, which consists software.

Interface: point of contact between user and programs.

Java (Sec 1.5.3): an object oriented programming language from Sun Micro system for a set of technologies for creating and running software programs.

Java server Faces (JSF): a framework to easy build Java web application using XHTML and EJB

JDK: Java-Development-Kit .A software development environment for writing applications in java.

Manual: the work done by paper or file based system.

Menu Bar: List of options displayed on the screen, from which the user may take a choice, usually menu bar appeared on the top of the windows frame.

Microsoft PowerPoint: Software developed by Microsoft Company used to present project documents for presentation.

Microsoft Word: Software developed by Microsoft Company used to write soft copy files.

Microsoft Visio: Software developed by Microsoft Company used to draw UML diagrams and other drawings.

My SQL Server: a data base server for managing and manipulating large data.

Net Beans IDE: Software IDE (Interactive-Development-Environment) used to write and compile java and other programming language.

Object Oriented: the basic concept on objects, which consists of data structures encapsulated with a set of routine called "methods" which operate on data.

Operating System: a program that controls the basic operation of the computer and can able to run others programs.

Query: question asked of the database to get a data form data.

Stakeholders: all persons that participate in computer system development including customers and sponsors.

Sun: Sun Microsystems, a US workstation manufacture.

Software: collection of programs and procedures for making a computer perform a specific task, as opposed to hardware.

System Use Case: the activity of the newly developed system in diagram.

Text Fields: one of components used by a user to input data to the computer system.

UML: Unified-Modeling-Language .a language used to communicate a customer and developer in

diagrams

Virus: A harm full piece of software that can replicate and transfer itself from one computer to another, without the user being aware of it.

WKU: Wolkite University

XHTML: (eXtensible Hypertext markup language) a more powerful and flexible framework to build web application.

4.1.4 Overview

Timetable Management System is a web based system which is developed for lecturers and Class Rooms for Wolkite Universityas their online timetable. It would contain four modules:

- ✓ Administrator(registrar) module,
- ✓ Department Head module
- ✓ Anonymous(Guest) Module

The functions of the registrar module are to handle the entire administrator task. Administrator also has to handle the additional, editing and deleting class rooms, departments and subject as well. For the department head module it contains the function maintain lecturer, courses, and their relation to make a schedule and Guest Module contain view timetable for the specific lecturer and room timetable for that semester. Lecturer can inquire for the class available and can book for the free classes. Students' module contains the functionality subject. Timetable Management System also contains a database, which stores the courses, rooms, and department and lecturers personal details. Only the administrator and department head can view, alter the data in the timetable schedule.

4.2 Current Software architecture

The software design of the class room timetable scheduling system is based on web architecture. The design is three-tiered architecture. That is the front-end, middle and back-end tiers. The front-end tier is the client side running on web browsers. The middle tier contains the web server and/or application server,

Web servers are a program that runs on a network server (computer) to respond to HTTP requests. The most commonly used web servers are Glass Fish Server or Apache Tomcat. The web server used in this system is Glass Fish. HTTP is used to transfer data across an Intranet. It is the standard protocol for moving data across the internet.

The back-end tier is the database server, which has the feature of distributed database. The client tier is the applications user interface containing data entry forms and client side applications. It

displays data to the user. Users interact directly with the application through user interface. The client tier interacts with the web/application server to make requests and to retrieve data from the database. It then displays to the user the data retrieved from the server.

4.3 Proposed software architecture

4.3.1Overview

The developed software system is designed to work on the web so the software architecture of the system is design based on the web architecture .the subsystems activities the database persistency and different hardware and software relation should be designed in such format of the web.

4.3.2 Subsystem Decomposition

Subsystem decompositions will help reduce the complexity of the system. The subsystems can be considered as packages holding related classes/objects. This subsystem decomposition is to maximize cohesiveness of components with subsystem component and coupling among subsystems. Based on the functional requirement of this project, the sub-system decomposition includes <<manageclassgroup>>, <<manage course>> and<<manage instructor >> <<allocate room and block>> <<class schedule process>>subsystems. The sub system of this project are defined in web services using different methods each subsystem invoke the method and communicate with the service and this methods are accessible by specified departments using web reference of web services. For instance create time table method manage by the department sub-system to manage the start date of the class.

The web service defined the registrar sub-system to be invoked by the registrar department to perform the following tasks:-

- ✓ Managing class group: is Managed by departments group based on schedule.
- ✓ Managing course: this method deals with the receiving data of courses of department to be scheduled with credit hour, course name, lecture hour, lab hour.
- ✓ Managing Instructor: The registrar worker manage the list of the instructors found in each department and what course they teach.
- ✓ Allocate room and blocks: Deals with allocate & providing Available Rooms and Blocks that are in the institution for each Department or group.
- ✓ Class schedule process: This use case describes scheduling process based on the above collected data.
- ✓ Generate class schedule report: finally the system generate class report based on the users need. That means if the user needs course break down he generate the course report and another also the same.

4.3.3 A decomposition diagram

A decomposition diagram shows a high-level function, process, organization, data subject area, or other type of object broken down into lower level, more detailed components. For example, decomposition diagrams may represent organizational structure or functional decomposition into processes.

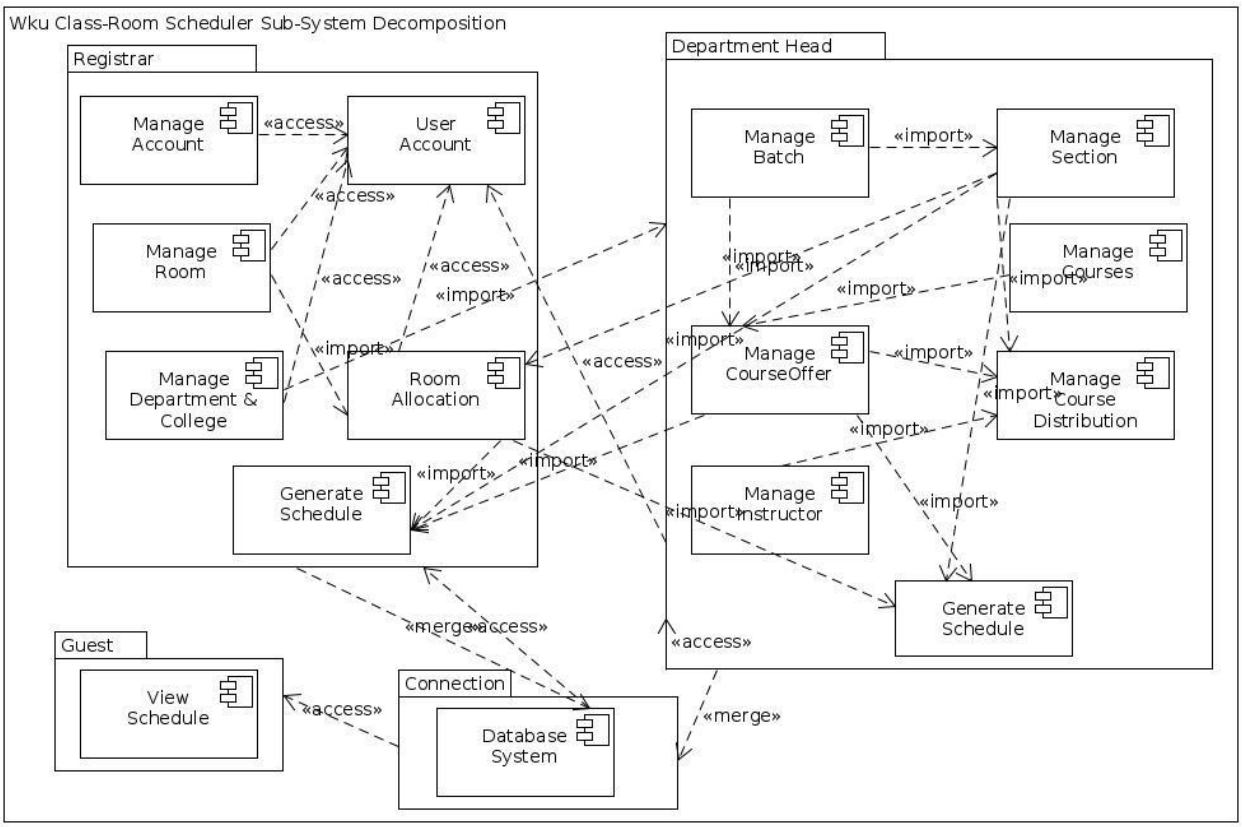

Fig4.1. Sub system decomposition

4.3.2 Hardware and Software Mapping

4.3.2.1 Hardware Mapping

The hardware required for class room timetable scheduling system services of Wolkite University is personal computers which fulfill the minimum requirements.

4.3.2 Software Mapping

The system is installed in any kind of (Linux, Windows, Mac or Solaris) operating systems and the user need web browser software.

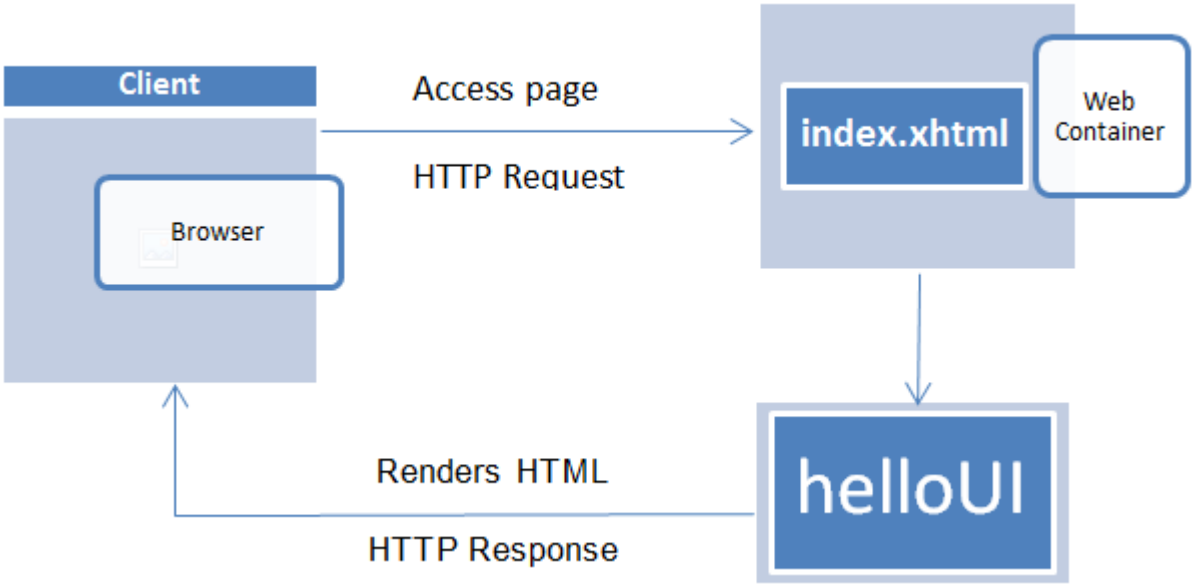

Fig 4.2 Request / Response Diagram

4.3.4 Persistent Data Management

This part of the document depicts the persistent data management rooted from the original class diagram that was produced in analysis phase. Persistence are used the schema of database. The strength of persistence data models is that data entities are conceptually the same as the table of relation data base and that attributes are the same as table columns in due process, relationships, which are logical but irrelevant to the system, are eliminated and those relationships that are important are added.

Persistence layered capsulate the capability to store, retrieve, and delete objects/data permanently without revealing details of the underlying storage technology. In the current database system we have used different tables as object and each object is related to each other and enforced by referential integrity. This schema enables as data manipulation activity such as select, search, delete, update on the data base.

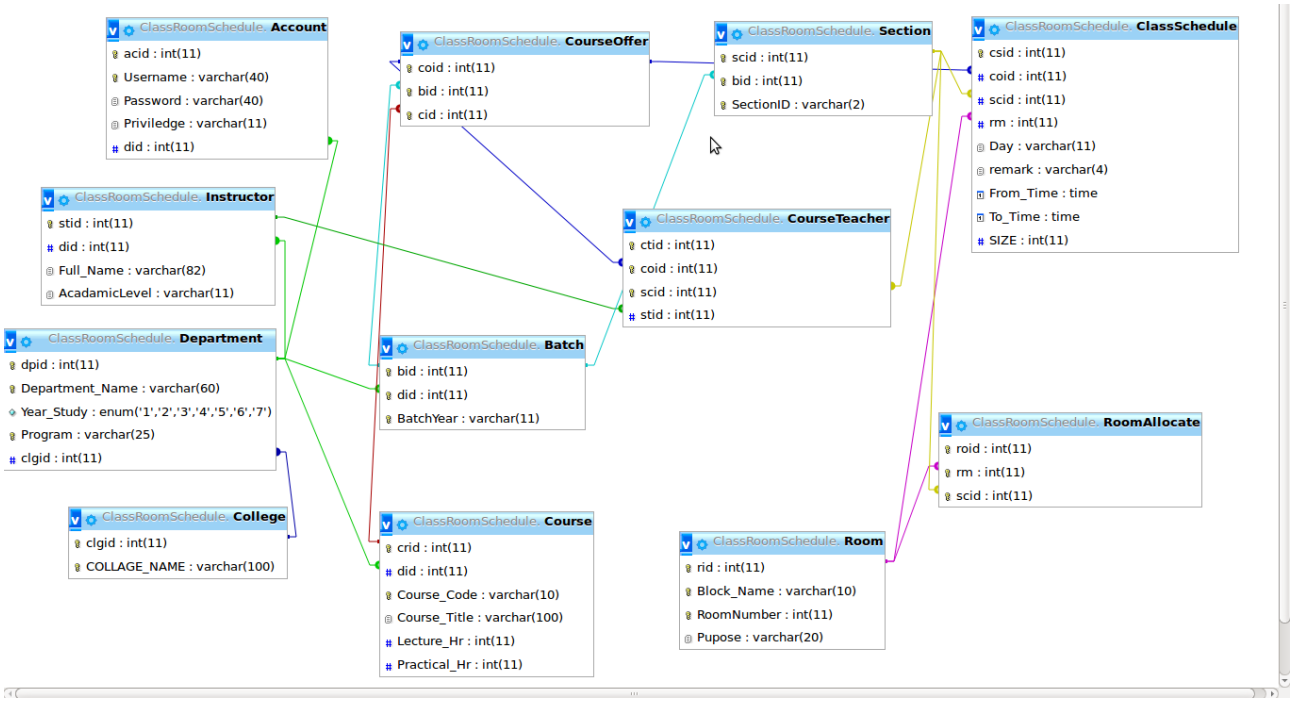

Fig 4.3 Persistent Data modeling

4.3.5 Access control and security

When the user wants to access the system the login screen asks to enter the correct username and password then the system validate the entered user name and password whether correct or not. If the system verifies valid user name and password, the user can access the system based on the security level in which the system grant.

List of operations	Actors		
	Department head	Registrar	Other users
Authentication ()	✓	✓	
Course register()	✓		
Instructor register()	✓		
Section register()	✓		
Batch register()	✓		
Course assign to teacher()	✓		
Course assign to section ()	✓		
Room register()		✓	
Arrange Class Schedule()	✓	✓	
Assign room to department()		✓	
Prepare schedule()		✓	
Create account for user()		✓	
View room schedule()	✓	✓	✓
View section schedule()	✓	✓	✓

Table 4.1 Access control and security

4.3.6 Boundary conditions

Boundary conditions of the system decide how the system is started, initialized; shut down and it defines the major failures within the system.

This project is to be functional to the user the system administrator should first start-up the server and the database engine load the necessary files into memory to start up the server. Then; the browser at the presentation tier makes connection with the SQL server database. The system provide exception handling for example the user can login into the specified module by entering the correct user name and password otherwise the system handles the fault and display error message for wrong user name and password and the system order the user to re-enter the correct user name and password. The system also halts if a hardware failure occurs such as failure of network link between client and server computer, and the failure hard disk.

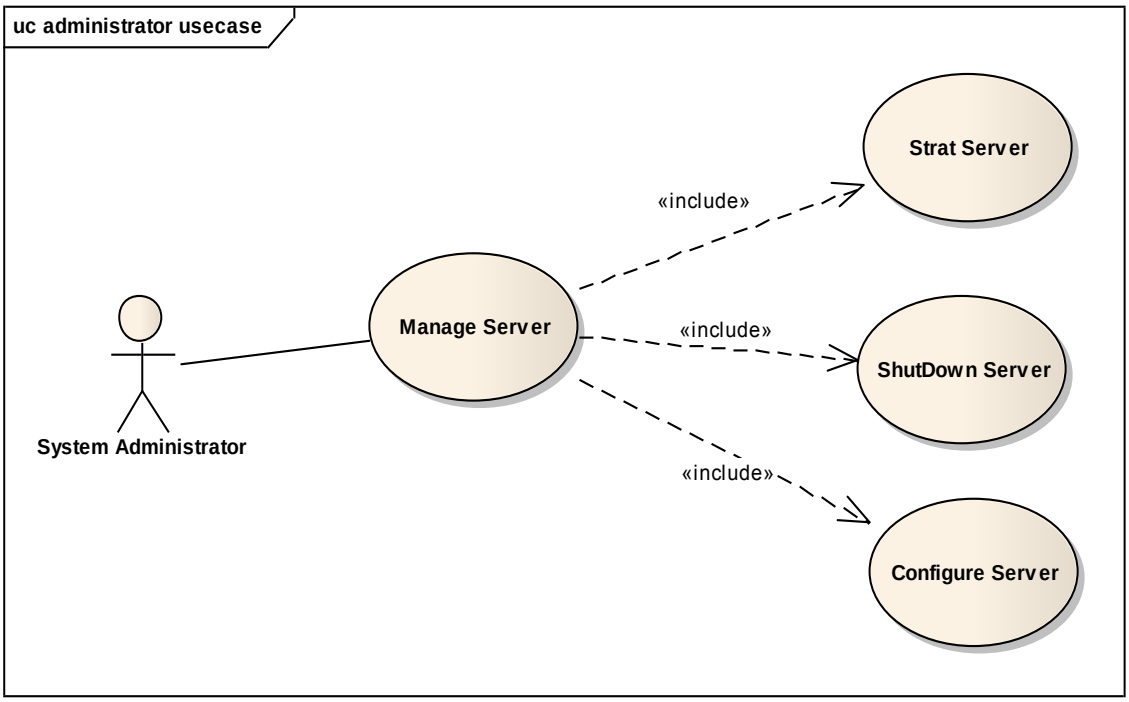

Figure 4.4 Boundary Conditions

4.4 Sub system service

The sub system specification specifies the sub-system we are going to implement and their operations.

Registrar sub-system: this sub-system is responsible to manage user accounts to create and update. It handles the number of operation to manage the user role.

Operation	Service
Authenticate()	Check the correct user name and password entry
Create Account()	Create new account for module user
Specify Roll()	Grant access privilege to module user
Block register()	Register blocks to the system
Room register()	Register rooms in the system
Assign block()	This sub system have the grant to assign blocks for departments
Assign room()	Assign room to the department

Table 4.2 Registrar Sub-system

4.4 Department sub-system:- this sub-system responsible to enroll course, sections and batches

Operation	**Service**
Authenticate()	Check the correct user name and password entry
Course register()	Register all the courses in the department to the system
Section register()	Register all sections in to the system
Batch register()	Register batches to the system
teachers register()	Register teachers in to the system
Assign teachers ()	Assign teachers for specific course and section

Table 4.3 Department Sub-system

CHAPTER FIVE

5. IMPLEMENTATION AND RESULT

5.1 Algorithms used

Here some Algorithms are designed to show the flow of programs in the system. They are semantic driven rather than syntax driven. That means, the rule of syntax is not respected as other programming language but it has a complete meaning as that of syntax based programming language. In addition, Algorithms show the flow and steps of logic in each function. This design part is important in the coding. Some of the algorithms are listed below.

5.1.2 Algorithm for log in to the WKU Timetable Scheduling system

Display home page
view any timetable information
click on login
Fill user name and password in the appropriate interface
If the filled input is valid
    If user is admin or department head
Open administrators' home page if user-name is admin or open heads page if it is head
    Else if user is data clerk
 Open a data clerks' home page
   Else if user is casher
Open casher home page   End if
Else if filled input is invalid
While the trial is less than three
Display error message and return login page
Otherwise
Display Error Message and return to home page
End if

5.1.3 Algorithm room offering

Display main Timetable page
        If   data clerks' selection is room offer
            Display room offer page on the main frame (Disp)

click on create button

select section group and select the room in appropriate interface

If input is valid

Register to Ro table database ()

Else if input is invalid

Display Error message () and return to the room offer page with error message.

End if

End if

5.1.4 Algorithm Course offering

Display main Timetable page

If data clerks' selection is course offer

Display course offer page on the main frame (Disp)

click on create button

select department batch and select the course for them in appropriate interface

If input is valid

Register to co table database ()

Else if input is invalid

Display Error message () and return to the course offer page with error message.

End if

End if

5.1.5 Algorithm Course Teachers

Display main Timetable page

If data clerks' selection is course distribute

Display course distribute page on the main frame (Disp)

click on create button

select department batch course (course offer),

select the section group from department batch

select Lecturer that teach the course

If input is valid

Register to co table database ()

Else if input is invalid

Display Error message () and return to the course offer page with error message.

End if

End if

5.1.6 Algorithm Class Schedule

Display main Timetable page

select the section group

click on Automatic button

If section not null

select courses under department batch of section (course offer),

for each course in list of course offer

getInstructor

get remark size of Lab or Lec Class

select Room based on the remark

if room not null

for each remark size

get list of mutual free-time and day for selected room, instructor and section

random select time and day from list

insert schedule

end loop

else

Display Error message () and return to the class main page with error message.

break

End if

Else

Display Error message () and return to the class main page with error message.

End if

5.2 Methods

To design the linear algorithmic view of the use case functions we used pseudo code that can represent the schema that could not run it in any programming language, Just for view of the overall implemented functions.

5.3 Sample Input/output Screen

5.3.1 Log in page

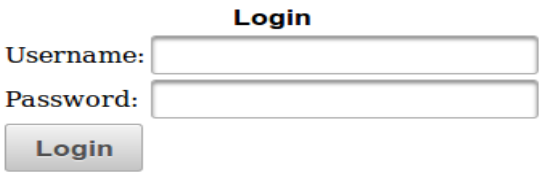

Fig5.1 login page

5.3.1 Timetable main page

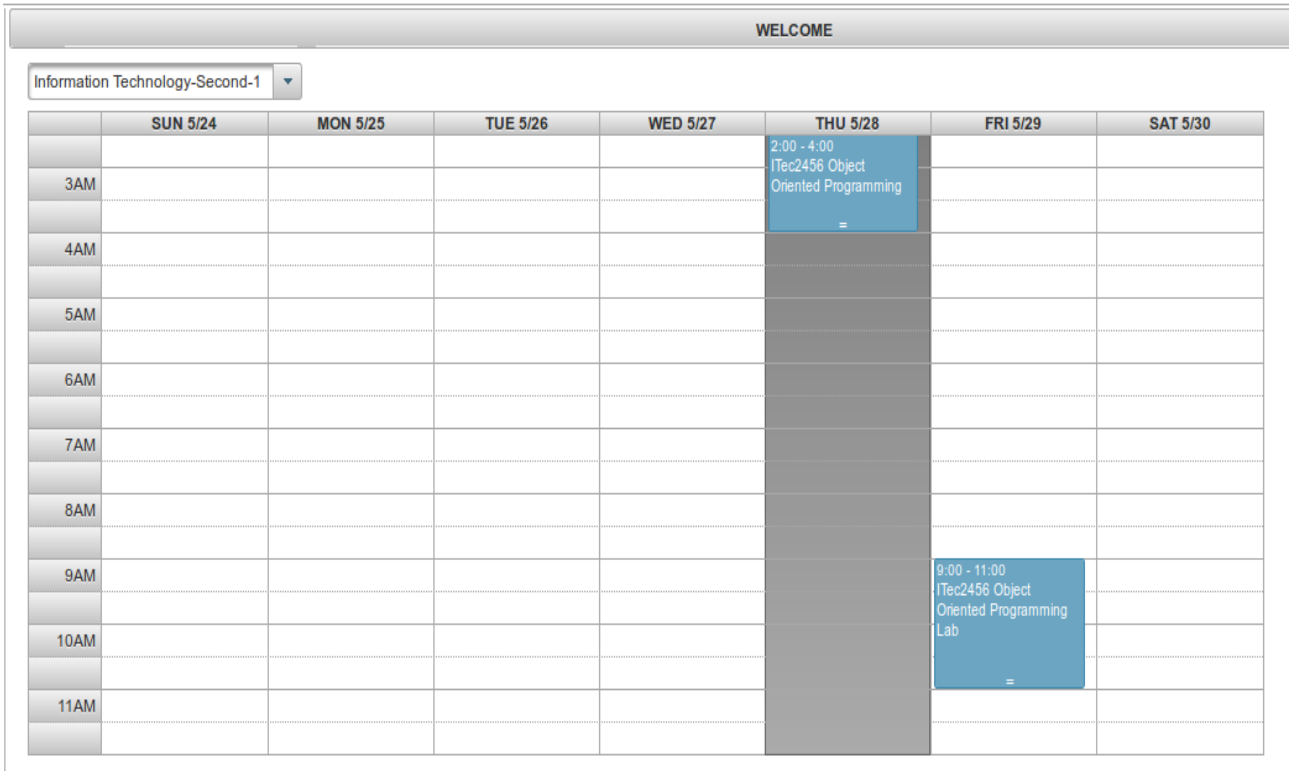

Fig 5.2 Timetable page

In this page shows the Visual Timetable Information site able the users to view any class schedule based on the selection.

5.3.2 Department Page

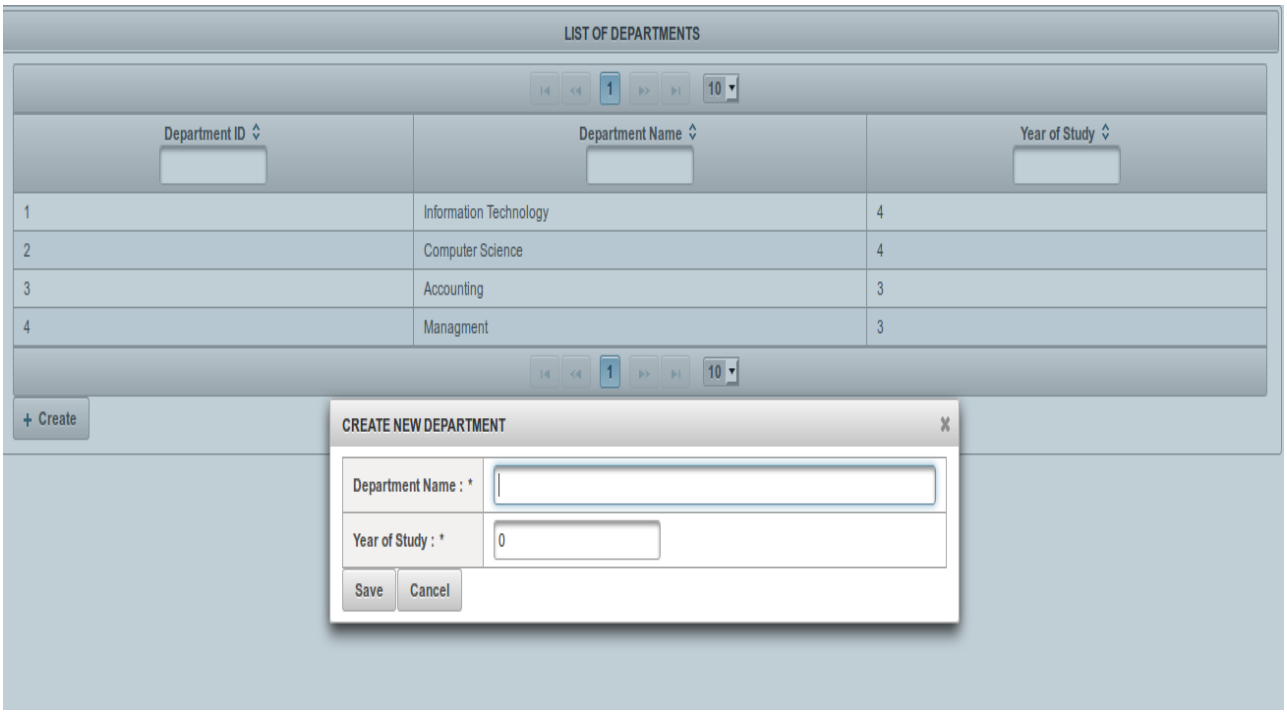

Fig 5.3 Department page

In this page shows the list of department and a dialog that popup after click the create button site able the admin (Registrar office) to create and edit department data.

5.3.3 Course Page

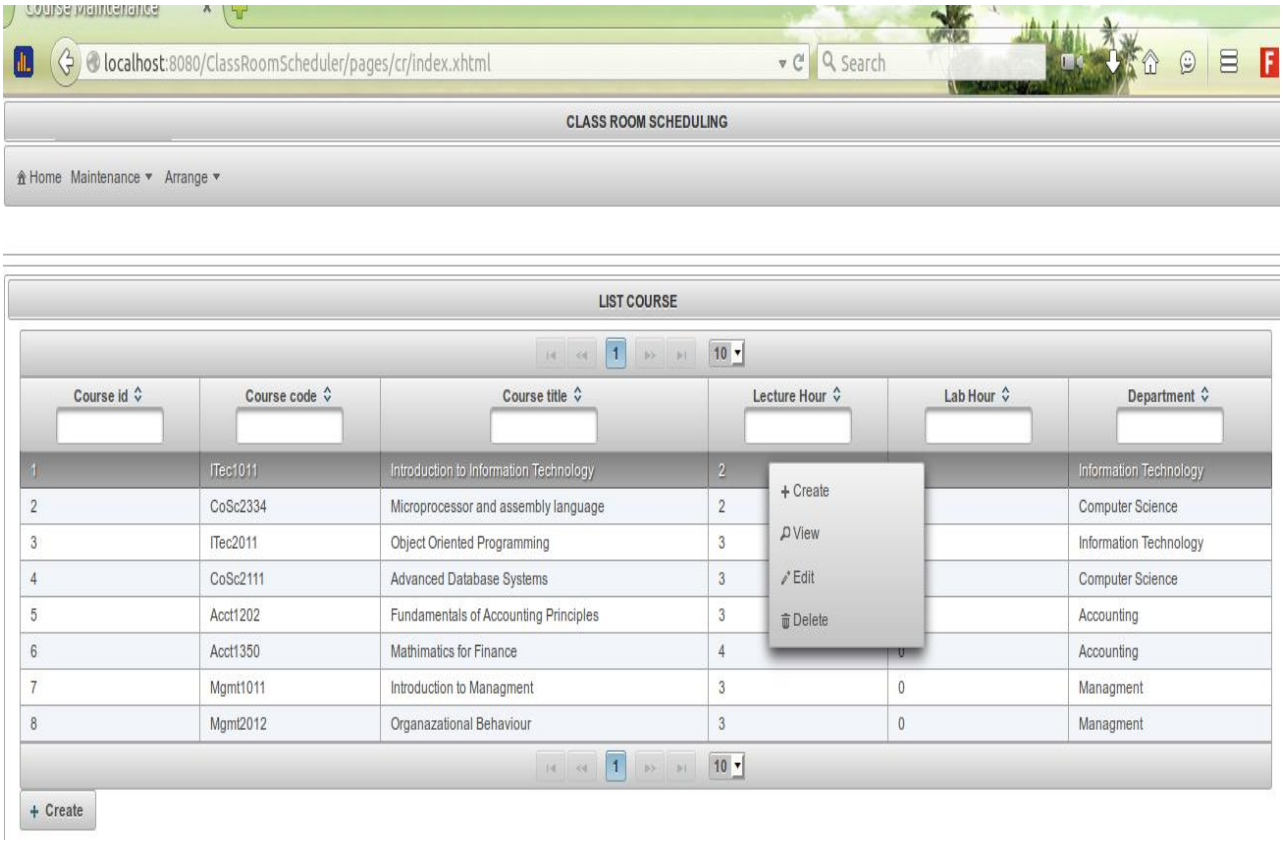

Fig 5.4 Course page

A page that shows the list of all courses with their data each this page also can able to edit delete and create data courses based on the user input dialog.

5.3.4 Course Teacher Page

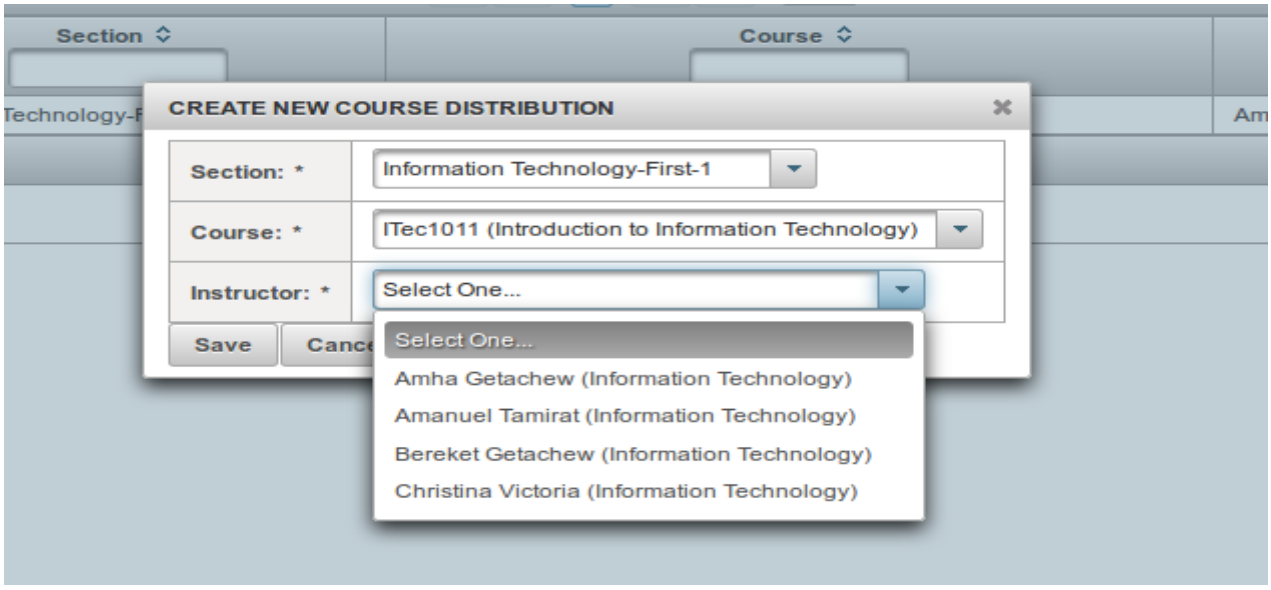

Fig 5.5 Course distribution page

A page that shows the input dialog to distribute courses for each teachers.

5.3.5 Class Schedule Arrangement Page

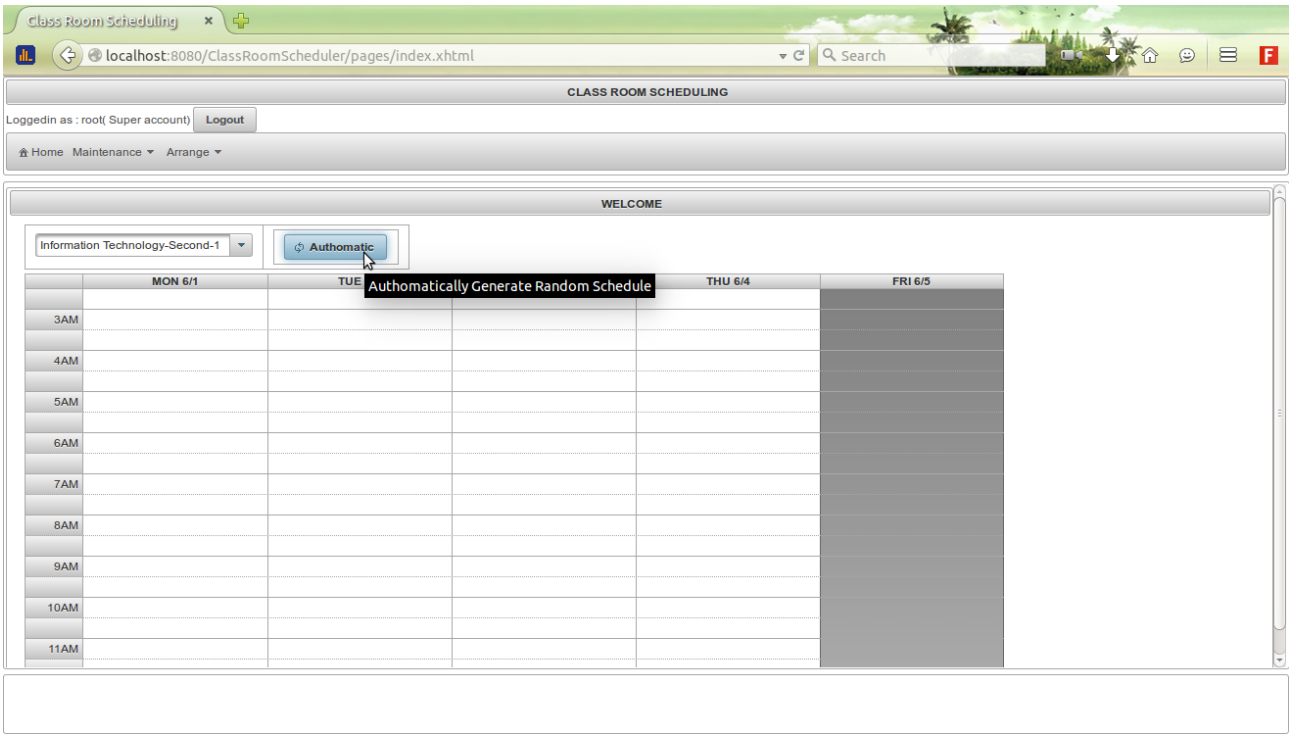

Fig 5.6 Class Schedule page

A page that shows a set of input option for making class schedule.

5.4 Sample codes

5.4.1 Create new department dialog page

<p:dialog id="DpCreateDlg" widgetVar="DpCreateDialog" modal="true" resizable="false" appendTo="@(body)" header="Create New Department" closeOnEscape="true">

<h:form id="DpCreateForm">

<h:panelGroup id="display">

<p:panelGrid columns="2" columnClasses="column">

<p:outputLabel value="Department Name :" for="dn" />

<p:inputText id="dn" value="#{dpController.selected.dn}" title="Department Name" required="true" requiredMessage="Department Name is required" size="60" maxlength="60"/>

<p:outputLabel value="Year of Study :" for="ys" />

<p:inputText id="ys" value="#{dpController.selected.ys}" title="Year of Study" required="true" requiredMessage="Study Year required"/>

</p:panelGrid>

<p:commandButtonactionListener="#{dpController.saveNew}"value="#{myBundle.Save}"
update="display,:DpListForm:datalist,:growl"
oncomplete="handleSubmit(xhr,status,args,DpCreateDialog);">

<p:confirmheader="#{myBundle.ConfirmationHeader}"
message="#{myBundle.ConfirmCreateMessage}" icon="ui-icon-alert"/>

</p:commandButton>

<p:commandButton value="#{myBundle.Cancel}" onclick="DpCreateDialog.hide()"/>

</h:panelGroup>

</h:form>

</p:dialog>

5.4.2 A Model Class called Room.java that map to rm table and its attribute

packagecom.project.model;

importjava.io.Serializable;

importjava.util.Collection;

importjavax.persistence.Basic;

importjavax.persistence.CascadeType;

importjavax.persistence.Column;

importjavax.persistence.Entity;

importjavax.persistence.GeneratedValue;

importjavax.persistence.GenerationType;

importjavax.persistence.Id;

importjavax.persistence.NamedQueries;

importjavax.persistence.NamedQuery;

@Entity

@Table(name = "RM")

@XmlRootElement

@NamedQueries({

@NamedQuery(name = "Rm.findAll", query = "SELECT r FROM Rm r"),

```
@NamedQuery(name = "Rm.findByRid", query = "SELECT r FROM Rm r WHERE r.rid = :rid"),
@NamedQuery(name = "Rm.findByBn", query = "SELECT r FROM Rm r WHERE r.bn = :bn"),
@NamedQuery(name = "Rm.findByRn", query = "SELECT r FROM Rm r WHERE r.rn = :rn"),
@NamedQuery(name = "Rm.findByPur", query = "SELECT r FROM Rm r WHERE r.pur = :pur")})
public class Rm implements Serializable {
private static final long serialVersionUID = 1L;
    @Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
@Basic(optional = false)
@Column(name = "rid")
private Integer rid;
@Basic(optional = false)
    @NotNull
@Column(name = "BN")
private String bn; // a room block Name
@Basic(optional = false)
    @NotNull
@Column(name = "RN")
privateintrn;// the room number
@Basic(optional = false)
    @NotNull
@Column(name = "PUR")
privatebooleanpur;//purpose of the room true if its laboratory room else class room
publicRm() {
    }

publicRm(Integer rid) {
this.rid = rid;
```

```
    }

publicRm(Integer rid, String bn, intrn, booleanpur) {
this.rid = rid;
        this.bn = bn;
this.rn = rn;
this.pur = pur;
    }
public Integer getRid() {
return rid;
    }

public void setRid(Integer rid) {
this.rid = rid;
    }

public String getBn() {
returnbn;
    }

public void setBn(String bn) {
        this.bn = bn;
    }

publicintgetRn() {
returnrn;
    }
public void setRn(intrn) {
```

```
this.rn = rn;
    }
publicbooleangetPur() {
returnpur;
    }
public void setPur(booleanpur) {
this.pur = pur;
    }
     @Override
public String toString() {
if(!pur)
return "R "+this.bn+""+this.rn;
else
return "Lab "+this.bn+""+this.rn;
    }
}
```

5.4.3 Instructor Staff List table XHTML

```
<p:dataTable id="datalist"
value="#{stController.items}"
var="item"
rowKey="#{item.stid}"
paginator="true"
rows="10"
rowsPerPageTemplate="10,20,30,40,50"
selectionMode="single"
selection="#{stController.selected}">

<p:ajax event="rowSelect"
```

```
update=":StListForm:createButton :StListForm:viewButton :StListForm:editButton :StListForm:del
eteButton"/>
<p:ajax event="rowUnselect"
update=":StListForm:createButton :StListForm:viewButton :StListForm:editButton :StListForm:del
eteButton"/>
<p:ajax event="rowDblselect"
onsuccess="document.getElementById('StListForm:viewButton').click();"/>

<p:columnsortBy="#{item.stid}" filterBy="#{item.stid}">
<f:facet name="header">
<p:outputLabel value="Staff ID"/>
</f:facet>
<h:outputText value="#{item.stid}"/>
</p:column>

<p:columnsortBy="#{item.fn}" filterBy="#{item.fn}">
<f:facet name="header">
<p:outputLabel value="Full Name" />
</f:facet>
<h:outputText value="#{item.fn}"/>
</p:column>

<p:columnsortBy="#{item.al}" filterBy="#{item.al}">
<f:facet name="header">
<p:outputLabel value="Acadamic Level" />
</f:facet>
<h:outputText value="#{item.al}"/>
```

```
</p:column>

<p:columnsortBy="#{item.did.dpid}" filterBy="#{item.did.dpid}">
<f:facet name="header">
<p:outputLabel value="Department "/>
</f:facet>
<h:outputText value="#{item.did}"/>
</p:column>
</p:dataTable>
```

5.4.4 A save new Class Schedule function (method)

```
public void saveNew(ActionEvent event) {
boolean f = false;
FacesContext context = FacesContext.getCurrentInstance();
for (Cs xsc : ejbFacade.getOverlapSchedule(getSelected())) {
System.out.println(xsc);
if (Objects.equals(getSelected().getRm().getRid(), xsc.getRm().getRid())) {
          f = true;
context.addMessage(null, new FacesMessage(FacesMessage.SEVERITY_ERROR,"Not Good "+"Room Overlapping occurs",  "") );
         // break room overlap occur;
          }
if (Objects.equals(getSelected().getScid().getScid(), xsc.getScid().getScid())) {
             f = true;
context.addMessage(null, new FacesMessage(FacesMessage.SEVERITY_ERROR,"Not Good Section Group Overlapping occurs",  "") );
             // break section overlap occurred;
          }
       }
// if everything is good then save it.
```

```
if(!f){
super.saveNew(event);
    }
    }
```

5.5 Results

After accomplishing the overall implementation process the result achieved is the automated wku timetable systems which register, retrieve and update the information of class schedule in efficient way. The developed system have user friendly pages that can be accessed with any kind of web browser , integrated database which is easily accessible, a secured service delivery and others which was designed above.

CHAPTER SIX

6. TEST

6.1 Test plan

6.1.1 Introduction

The test plan outlines the standard tests that should be applied to the group of program files (Java,XHTML, MySql code etc) that are used to build each Web page i.e. the tests listed in this document used as a template when building the test plan for each Web page.

This testing phase uses a number of testing techniques to ensure that the code matches the required program specifications. Through those techniques we tested the function in different way by unit testing, integration and System testing.

6.1.2 Test Plan Overview

This document includes different activities. In the first part testing constraints and precondition are described this include functions to be tested and not to be .next to that the test plan description is documented including the use case functions to be tested in the test case specification.

The last part of this test document is test case specification in which the unit, integration and system testing is done. Features to be tested/not to be tested

6.1.3 Features and Functions to Test

Accessibility: The colors used on this Web page are friendly to color blind viewers

Compatibility: All on the general release versions of the following Windows based browsers should accurately render the Web page (with/without style sheets enabled) when viewed via 800x600 screen resolution in full screen mode (e.g., no horizontal scroll bars):

- ✓ Mozilla Firefox
- ✓ MS Internet Explorer
- ✓ Netscape Navigator
- ✓ Opera
- ✓ Google chrome

The content of the Web page is clearly readable when printed with a black and white printer using Letter sized paper and also color printer using A4 sized paper. The Web page is understandable when viewed through a text only browser

Content

- ✓ All Times, string and number amounts should be displayed using standard format

- ✓ Input fields should be able to accept international variations
- ✓ All text must be spell checked with a US English spell checker but this does not included the words.

Functional: The Web page implements all of the business requirements described in its' associated requirements specifications. Acceptable functional should be checked for under the following adverse client-side conditions.

Navigation: All the links on the Web page should be checked to ensure that they meet the following specifications: The link should not be broken (unless the target has not yet been developed) and goes to the most appropriate location.

- ✓ The link did not alter the browser's default link colors.
- ✓ The link must have an associated "Title" link tag specified.
- ✓ The link text does not wrap to two lines (this may confuse visitors into thinking that there are two links instead of one).
- ✓ If the link points to a directory (instead of a specific Web page), the link ends with a slash.
- ✓ Internal URL links should reuse the existing browser instance (the exception would be any help pop-up Windows).
- ✓ Internal links must use relative addressing rather than absolute.
- ✓ Internal links must use lowercase characters for the address.
- ✓ External URL links should spawn a new browser instance.

Performance: Exceptions to this rule must be approved.

Security

- ✓ Input data received from the user must be parsed to make sure that it does not contain "out of bounds" or "buffer overflow" input data.
- ✓ Input data received from the user must be parsed to make sure that it does not contain inappropriate meta-character sequences e.g. &&

Usability

- ✓ Critical information has not been placed on the lower portion of the Web page (if the position of this information requires the user to use scroll down, the majority of visitors are unlikely to ever read it)
- ✓ Related information is grouped together on the Web page to minimized eye movement
- ✓ There are no competing/duplicate actions/options on the Web page, which might confuse the user or cause them to make an error
- ✓ Multiple key combinations can also be entered sequentially or are mapped to a single key.

6.1.3.1 Feature or Combination of Features Not To Be Tested

The following list of features not tested:

- ✓ Ability to cope with volume, load, and hardware faults. The tests case should not include testing the systems abilities to deal with hardware issues that may arise.
- ✓ Time related bugs like a session timeout. Once the session times out, the variables are lost, therefore the user must start over. Although most errors are unanticipated, tests cases should not test for error conditions that may arise outside the DBMS.
- ✓ User interface inconsistency, where the users resolution or PC settings may cause some difference in appearance to the web page.
- ✓ User-friendliness, although the system is designed to be easy to use, the computer experience of the end users cannot be measured.

6.1.4 Approach

The following documentation was used to prepare the test design, case and procedure specifications. The tests are intended to verify the accuracy and completeness of the information in the documentation in the areas covered by the tests.

The testing approach accounts for the following:

- ✓ Interface Testing
- ✓ Security Testing
- ✓ Performance Testing
- ✓ Accessibility Testing
- ✓ Constraints

The test is performed by entering test data into the web pages, and observing the displayed output. Validations require that the database be queried to ensure that the proper data was created, retrieved, updated or deleted.

6.1.5 Testing materials (Hardware/Software requirements)

For the proper functioning of the system the following hardware and software are required.

6.1.5.1 Hardware requirements

Web application server

- ✓ Processor: Intel(R) core 2DU, CPU 550@3.24GHz processing speed.
- ✓ RAM **:** >1GB.
- ✓ Hard disk: >=320 GB.

Database server

- ✓ Processor: Intel(R) core™i3 CPU 550@3.20GHz processing speed.

- ✓ RAM: >=4 GB**.**
- ✓ Hard disk: > =500 GB.

Other

- ✓ Computer (Desk top) or laptop.
- ✓ CD
- ✓ Removable hard disk (if any).

6.1.5.2 Software Requirements

- ✓ Browsers (one required client side).
  -Mozilla fire fox with JavaScript enabled active flash player.
- ✓ Web application server (one required, server-side).
  -Glass Fish server
  Relational Database (one required, server-side).
- ✓ MYSQL Server.

6.2 Test Case Specification

6.2.1Test Procedures:

The testing procedures are listed for each use case. The tester followed every step in the following test cases. If no errors occurred, then the tester needs to validate that the outcome is correct. If errors occur, a Software Problem Report written.

6.2.2 Test Procedure Conventions:

Each test case has given specific instructions describing the steps which need to be taken to complete the test cases. All test cases start at the home page.

6.2.3Test Data:

All test data is contained within the test cases below. The following are test cases for each use case described in this document.

6.3 Unit testing

6.3.1 Login

6.3.1.1Test Case #1

This use case emulates a user logging into the system.

STEPS:

1. Load the WKU Timetable site.

2. Click on Login button

I. Test Items

The program files that are needed to perform this test:

- ✓ login.xhtml

II. Input Specifications

STEPS:

1. At the login screen, display

enter the correct Username

enter the correct Password

2. Click on the Log In button.

The main page is loaded. Displayed on the left frame is a menu system for navigating through all of the components of the system. In the center frame the content for the selected page they are on, at first being only a welcome screen.

3. Inter case Dependencies: None.

6.3.1.2Test Case #2

This use case emulates an attempt to enter an incorrect password when logging in.

STEPS:

1. Load the WKU Timetable site.

2. Click on Login button

I .Test Items

The program files that are needed to perform this test:

- ✓ index.xhtml
- ✓ useraccount.java

II. Input Specifications

STEPS:

1. At the login screen, enter:

Username: admin

Password: WRONG

2. Click on the Log In button.

III. Output Specifications

The user returns to the login screen, with a new message stating that login failed because the username or password was incorrect.

IV. Inter-case Dependencies: None

6.3.1.3Logout

This test case emulates a user logging out of the system.

STEPS: None

I. Test Items

The following program files that are needed to perform this test:

- ✓ menu.xhtml
- ✓ useraccount.java

II. Input Specifications

STEPS:

1.From the current page, click on the Log Out link, which is located in the menu on the left frame. You should be logged out of the system and taken back to the login page.

III. Output Specifications

If the logout was successful, then the page displays the Login page, but with an added message stating that the user was successfully logged out.

III. Inter case Dependencies

The login test case must be completed prior to the log out use case, as only logged in users can log out of the system.

6.3.1.4 Use Case: Generate Reports

This test case emulates a user generating a report.

STEPS:

1. Starting at the home page, click on the 'Report' button.

The generate reports page is now displayed, and awaits menu entry.

I. Test Items

The following program files that are needed to perform this test:

- ✓ index.xhtml
- ✓ report.java

II. Input Specifications

STEPS:

1. click "Report " button

III. Output Specifications

If everything worked, then the page displayed all of the information in the database forreport.

IV. Inter-case Dependencies

The login test case must be completed prior to the generate reports use case, as only logged in users can generate reports.

6.3.1.4 Use Case: Create User

This test case emulates a user who has administrator privileges to create a new user.

STEPS:

1. Starting at the login page, enter the following username and password:

Username: root

Password: secret

2. Click on the 'Login' button.

The web page will now display the administrator menu

3. Select Maintenance menu under that select "User Account.." menu item

3. Click on the 'Create' button.

The web page pop up a dialog now displays a form that contains text-box two password fields , one combo-box to select roll and also another combo-box to select from staff Instructor which is optional to capture the entire required field for a user profile to be store in the database

I. Test Items

The following program files that are needed to perform this test:

- ✓ /pages/acu/index.xhtml
- ✓ /pages/acu/create.xhtml
- ✓ AcuController.java

II. Input Specifications

STEPS:

1. Enter the following information

Username: Clerk Name

Password: secret

Verify Password: secret

Privilege: select" root"

Staff : leave it

2. Click on the 'Save' button.

The web page displays confirmation message that the operation was completed successfully. It also displays two options, back to main menu and makes another update, for the next user action

III Output Specifications

1. Click on the 'Home' link and the web page now displays menu for the administrator.

IV. Inter-case Dependencies: Not Applicable

6.3.1.5 Use Case: Update User

This test case emulates a user who has administrator privileges to change the user and type.

STEPS:

1. starting at the login page, enter the following username and password:

Username: root

Password: secret

2. Click on the 'Login' button. The web page displays the administrator menu

3. Select Maintenance menu under that select "User Account.." menu item

I. Test Items

The following program files that are needed to perform this test:

- ✓ /pages/acu/index.xhtml
- ✓ /pages/acu/edit.xhtml
- ✓ AcuController.java

II. Input Specifications

STEPS:

1. Select 'user1' from the list of table and right click on the selected user account
2. pop up option will be there
3. Select the "Edit' menu item

The web page pop up a dialog now displays a form that populate a data of the selected user account contains text-box two password fields , one combo-box to select roll and also another combo-box to select from staff Instructor which is optional to update the entire required field for a user profile to be store in the database

The web page displayed confirmation message that the operation was complete successfully. It also displays two options, back to main menu and makes another update, for the next user action.

III. Output Specifications

1. Click on the 'Back to Main Menu' link and the web page now displays menu for the administrator.

IV. Inter-case Dependencies

This test case requires that the create user test case be completed before this test can be perform.

6.3.1.6 Use case Remove User

This test case emulates a user who has administrator privileges to delete a user from the database.

STEPS:

1. starting at the login page, enter the following username and password:

Username: root

Password: secret

2. Click on the 'Login' button. The web page displays the administrator menu

3. Select Maintenance menu under that select "User Account.." menu item

I. Test Items

program files that are needed to perform this test: i.e. the program is check all the input and output.

- ✓ /pages/acu/index.xhtml
- ✓ confirmation.xhtml
- ✓ AcuController.java

II. Input Specifications

STEPS:

1. Select 'user1' from the list of table and right click on the selected user account
2. pop up option will be there
3. Select the "Delete' menu item

The web page displays confirmation message that the operation is warning click "Yes" to continue

The web page displays confirmation message that the operation was complete successfully. It also displays two options, back to main menu and makes another update, for the next user action

III. Output Specifications

1. Click on the 'Back to Main Menu' link. The web page will now display menu for the administrator.

IV. Inter-case Dependencies:

This test case requires that the create user test case be completed before this test can be perform

6.4 System testing

System testing is based on process description and flows, emphasizing pre-driven process links and integration points.

In essence system testing is not about checking the individual parts of design as a unit test, but about checking the system as a whole. In effect it is one giant component. System testing ensures the fulfillment of the following:

- ✓ Functional requirements of the system are tested by checking the SRS and object modeling with the implementation of it.
- ✓ Non Functional requirements such:
  - ✓ Accessibility □Compatibility □ Content □Functional
  - ✓ Navigation □Performance □ Security □Usability

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

University is the large organs of FDRE Government which is empowered to handle most of public administration services mainly focused on teaching and learning process / educational sector which needs accurate free error class room schedule .

From the depth investigation made in Wolkite University, it was founded that current class room scheduling activities of the university was time consuming, costly, error prone and non- integrated due to manual service delivery mechanisms which results delay on the development of the university.

Automation system in class room scheduling refers to the varied computer machinery and software used to digitally create, collect, store, manipulate, and relay information needed for accomplishing basic tasks. So, class room scheduling system is crucial for universities. We designed the database management system based on requirements of the new system in way that it achieves those noble goals of automation technologies.

The system we developed creates convenience for customers to get fast and efficient timetable services. These services include view the timetable in different platform view, lecturer free clash schedule, room free-clash schedule and collection of courses gained from those services which are stored in well integrated centrally administered database.

This web based system has the capability to be used in the organization Intranet site. And we assessed the implementation issues that should be considered when introducing the system like privacy issues, security of the system, and others.

Overall, class room scheduling System has been successfully built and has achieved and fulfilled the objectives and requirements that are stated in the project proposal.

7.1.2 Limitation of the project

This project has its own limitations these are the system is internet connection dependent and mostly attacked by viruses .

7.1.3 Future works

In the future we plan to develop more enhanced by adding some futures in this time table scheduling system such as adding the schedule of regular as well as extension students, exam

schedule, add and drop course.

Appendix

Appendix 1: User's manual/guide

I. Getting Started

To do this getting started you should have to follow the following procedures. Those procedures are:-

- ✓ First you should have to Activate Glass Fish server server from the Desktop or Start up Menu
- ✓ Then you should have to launch NetbeansIDE go to under services tab
- ✓ Select Web server , right click on GlassFish server and click start
- ✓ In the run Dialog box type the URL of the page (example http://localhost/ClassRoomScheduler/) and then click "OK" button.
- ✓ After the application have been installed and tested the system is ready to be functional and then preparation could begin to place the new system in to operation.

II. Installation

After installing both NetbeansIDE and MySQL Server do the following procedures. Those procedures are:-

- ✓ First install NetbeansIDE server on your computer c: /
- ✓ Copy the folder Class Room Schedule to your c:/MyDocument/NetBeansProject/ folder
- ✓ Run NetbeansIDE
- ✓ Select menu file and select import then select your folder
- ✓ Now run MySQL server
- ✓ Explore Mozilla and type http://localhost/ClassRoomScheduler/init.xhtml to create database called Class Schedule DB and tables system
- ✓ Now you can access the system freely but, type http://localhost/ClassRoomScheduler/ in the internet search box or on your computer RUN dialogue box, you can get your home page.
- ✓ Finally, log in with your account so, Installation is finished.

III. System Flow

In this section we tried to show some of the flow of the system.

Department register

System flow

The data clerk opens his home

Select Maintenance menu

The root press department... Menu Item

The system takes to the list of departments

Then click “Create” button

The system takes all inputs from the user

Check the validity

System Registers department’ profile on the database and displays success report.

Course register

System flow

The data clerk opens his home

Select Maintenance menu

The head press courses... Menu Item

The system takes to the list of courses of the department

Then click “Create” button

The system takes all inputs from the user

Check the validity

System Registers course’ profile on the database and displays success report.

Instructor Registration

The data clerk opens his home

Select Maintenance menu

The head press Instructor... Menu Item

The system takes to the list of instructors under the department

Then click “Create” button

The system takes all inputs from the user

Check the validity

System Registers instructor’ profile on the database and displays success report.

Room Registration

System flow

The data clerk opens his home

Select Maintenance menu

The root press rooms... Menu Item

The system takes to the list of rooms

Then click “Create” button

The system takes all inputs from the user

Check the validity

System Registers room’ profile on the database and displays success report.

Batch registration

The data clerk opens his home

Select Maintenance menu

The head press Batch... Menu Item

The system takes to the list of Batch of the department

Then click “Create” button

The system takes all inputs from the user (department and the year of the batch)

Check the validity

System Registers batch’ profile on the database and displays success report.

Section group registration

The data clerk opens his home

Select Maintenance menu

The head press Section... Menu Item

The system takes to the list of Section groups of the department

Then click “Create” button

The system takes all inputs from the user (department batch and the section identifier of batch)

Check the validity

System Registers section’ profile on the database and displays success report.

Course teacher offering

The data clerk opens his home

Select Arrange menu

The head press Course distribution... Menu Item

The system takes to the list of Course assessment groups of each section

Then click “Create” button

The system takes all inputs from the user (courses from courses offer to batch and the section identifier of batch and instructor)

Check the validity

System Registers course teacher’ profile on the database and displays success report.

Class scheduling offering

The data clerk opens his home

Select Arrange menu

The head press Class scheduling... Menu Item

The system takes to the list of Class Schedule timetable assessment groups of each section

Then click “Create” button

The system takes all inputs from the user (courses from courses offer to batch and the section identifier of batch, room, and day, from and to time)

Check the validity

System merge class schedule’ profile on the database and displays success report.

Bibliography

[1]. E. Burke and W. Erben. Practice and Theory of Automated Timetabling, Third International Conference, Germany, Springer Private Limited, August 2000

[2]. J. G. Hedberg et. al. (1992). Educational information systems: Problems of the small educational organisation. Australian Journal of Educational Technology, 8(2), 132-160. http://www.ascilite.org.au/ajet/ajet8/hedberg.html

[3]. M. Marte. Models and Algorithms for School Timetabling, A Constraint-Programming Approach, Ph.D dissertation, an der Fakult¨at f¨ur Mathematik, Informatik und Statistik der Ludwig-Maximilians-Universit ¨at M¨unchen, July, 2002

[4] R.J. Willemen. School Timetable Construction: Algorithms and Complexity, Thesis, Faculty of Mathematics and Computer Science, Technische Universiteit Eindhoven, 2002.

[5]. S. Petrovic and E. Burke. University Timetabling, School of Computer Science and Information Technology, University of Nottingham, 2002, pp. 1-4

[6] T. Willis and B. Newsome. Beginning Visual Basic 2005, Wiley Publishing, Inc., 2006 T.

[7] Larman C. (2004), Applying UML and patterns: An introduction to Object-Oriented Analysis and Design, Third Edition, Addison-Wesley, Boston, USA

[8]. Johnson. Computational course sceduling system. 1993.

[9]. et.al, Aluarez valdes. simulated annealing . Madrid : Madristan inc., 2002.

[10]. et.al, Elmohamed. Colony algorithis. Turkey : Alvarez inc., 1998.

[11]. et.al, Socha. Genetic algoritms. Florida : Elservi publishers, 2003.

[12]. Genetic Alogorith. Florida : Elsirvirinters, 2003.

[13]. Johnson, Foulds and. Slot manger. Britain : Britanica inc., 2000.

[14]. schaerf, Gaspero and. Multiagent Methods. Rome, Italy : Italio printers, 2003.

[15]. Thompson, Hinkin and. Created sceduling expert. Massachuset : Massachuset publisher group, 2002.

[16]. Worg. Multineighborrood local searher. China : Ting Publisher inc., 2003.

[17]. Zhans, White and. Chiar and inin. Arizona : Elsivir Pulishers, 1998.