Class Scheduling System for Tikur Anbessa Senior Secondary School

Henok Feleke, Abdulquyyuma Ali, Biniam Gizaw and Yoseph Tadese
St. Mary's University

Abstract

Tikur Anbessa Senior Secondary School's class scheduling system is based on a backward manual data processing system. This manual data processing system has created many problems like: data redundancy, inconsistency and problem of sharing; assignment of an instructor to different classes at the same time; assignment of the same class room for two or more classes; and lack of information on free class room and lab. The main objective of this project is to analyze the existing classes and exam scheduling and design, implement a new and better automated information system. The Tikur Anbessa Senior Secondary School's scheduling system, focuses on assigning room and instructors to different sections and the network design and implementation. In relation to the methodology, the tools applied include: interview, self administered questionnaire, observation; and development of program tools. The primary tools used are grouping similar tasks WBS (work break down structure) and decomposition technique. Using these tools we divided a software project in to different phases. The software project schedule is proposal according to the arrangement of the phase; and the tools comprise: SQL -for implementation; WORD - for documentation part; and C# or PHP TO WRITE THE CODE.

It is hoped that the thesis will have a beneficial impact in the academic environment in terms of affording: insight into the systems nature of school scheduling, in particular into the role and importance of variables involving time, and opportunities for, and justification of, improvements to schedules, through time pattern analysis. We also believe that it is time to realize the need for modern information system and work towards making the most of it. The decision to have one as soon as possible will determine the schools future success.

Key Words: Class Scheduling, Software Project System, Tikur Anbessa, Secondary School

1. Introduction

1.1. Background of Tikur Anbessa Senior Secondary School

Tikur Anbessa Senior Secondary was established in 1972 E.C. Tikur Anbessa Senior Secondary School was a popular school in Addis Ababa, since the opening of the school, Students used to come from nearby, the Cathedral School after completing grade 8. The students were very good at English. They could listen and speak English very well. There were some Ethiopian students capable of speaking French as well. Those students came from the nearby school "Lycee G/Mariam" French school. They joined the school because they were unable to take the Ethiopian National Examination at Lycee French School which was a requirement to join universities.

Tikur Anbessa Senior Secondary School became very much popular in late seventies. The popularity came as a result of the ESLCE (National Examination) result of students. The school used to stand first from Addis Ababa on the National Examination results. These achievements were repeated again and again.

1.2. Statement of the Problem

Tikur Anbessa Senior Secondary School's system's main problem is caused by a backward manual data processing system. This manual data processing system has created many problems like:

- Data redundancy, inconsistency and problem of sharing;
- Assignment of an instructor to different classes at the same time;
- Assignment of the same class room for two or more classes; and
- Lack of information on free classrooms and labs.

1.3. Objective of the Study

1.3.1. General **Objective**

The main objective of this project is to analyze the existing classes and exam scheduling problems of Tikur Anbessa Senior Secondary School and as a result design and implement a new and better automated information system.

1.3.2. Specific Objective

To achieve the above general objective, the following specific objectives have been set.

- Computer exam scheduling with appropriate information.
- Computerized the record office with available information.
- Computerized class scheduling with information (free rooms, laboratory and class).
- Computerized the data base in a manner that allows each building to have access to the necessary data creating a first data processing system.

1.4. Feasibility Study

During our visit to the school we have been able to realize the difficulties and complications of manual scheduling system. So the system will be useful and applicable to the problem in hand.

1.5. Significance of the Project

Several times each year, an academic institution is concerned with the process of choosing times for each of its classes and then assigning instructors, rooms, and students to these classes. This project is concerned with a central aspect of the school scheduling process, especially the configurations of time used in establishing a school's schedule. Like most decision-making processes, much can be said for taking a systems approach. A systems approach to school scheduling requires careful-attention to the variables involving time, since the times chosen for classes directly affect many quantitative and qualitative measures: conflicts between classes, personal work schedules, resource utilization and loading, preparation time (for instructors), individual study time (for students), etc.

The process of scheduling a school program must, of course, be subordinated to the overall educational objectives of the school. The unfortunate paradox of school scheduling is that the highest education aspirations of a school can be frustrated by a schedule supposedly designed to implement them. Many schools are faced with the mechanics of scheduling at odds with the objectives of scheduling, creating a problem unduly out of proportion.

Typically, a school has enough problems to cope with without getting bogged down in scheduling concerns.

1.6. Beneficiaries of the Project

The purpose of the project is to draw the attention of Tikur Anbessa Senior Secondary School's authorities to the importance of careful time pattern analysis, and to assist these authorities in performing a proper time pattern analysis. So, there will be a fully functional and reliable timetable that can maximize the productivity of the learning system.

2. Methodology

2.1. Interview

Interview was conducted with the Tikur Anbessa Senior Secondary School's staff and efforts were made to ask various questions. Accordingly, we have grasped the necessary facts from our interview.

2.2. Questionnaires

Self administered questionnaires were used in order to find additional facts about the present information system and opinions about how this system can be improved.

2.3. Observation

Some of our team members have tried to personally observe how activities are carried out within the information system. Some of us did try to receive explanation on the spot.

2.4. Development Environment/Programming Tools and other Tools

The primary tools we use to group similar tasks are the WBS (work break down structure) and decomposition technique. Using these tools we can divide a software project into different phases. The software project schedule is proposed according to the arrangement of the phase.

We may use one of the following tools in our Project

- SQL -for implementation,
- WORD - for documentation part,
- C# or PHP TO WRITE THE CODE.

2.5. Scope

The scope of this project is limited to the Tikur Anbessa Senior Secondary School's scheduling system, which focuses on assigning room and instructors to different sections and the network design and implementation.

Due to time limitation, the network design and implementation stage will only go as far as the design stage. In other words, the network layout will be designed but will not be implemented.

2.6. Risks, Assumptions and Constraints

In the process of our system development the risk or the big challenge is the shortage of time in hand considering the vastness of the problem to tackle down.

The other problem is the human and time resource needed to make the school environment familiar with the system.

2.7. Phases and Deliverables of the Project

Phase 1: Preparation of proposal,

Phase 2: Requirement analysis,

Phase 3: System modeling,

Phase 4: Designing,

Phase 5: Implementation.

Table 1: Work-break Down Structure, Coordination and Deliverables

TASK	START	FINISH	DELIVERABLES
Proposal of the new system	1/10/2015	7/10/2015	Proposal document
Requirement analysis	8/10/2015	23/11/2015	Requirement definition document
System analysis	24/11/2015	14/12/2015	System model document
System design	15/12/201	12/1/2016	Braked down system
Conclusion and recommendation	13/1/2016	23/1/2016	Conclusion and recommendation document

3. Analysis

3.1. Introduction

The use of information and information technology is becoming immensely important in today's social and business world. Failure to efficiently and effectively use information resources results in critical challenges for survival. As a matter of fact, we have reached a stage in which most activities require the proper usage of information, processing, organization, storage, etc.

Because of this reason organizations nowadays need to have high performance information systems. Many organizations or business and public institutions in developed countries currently have their own established computer-based information systems. However, developing countries such as ours are in a very premature stage in the implementation of this kind of highly modern information systems.

Information systems commonly contain information in relation to the activities of the owner organizations. These activities may be departmental activities or general purpose information regarding the organization. Depending on the kind of sector the organizations are involved and depending on their need the content of the components in the information system might vary.

The schedule cycle is the period of time over which the schedule does not repeat itself. It may be as short as a day or as long as the entire term. The cycle often coincides with the five-day calendar week. Since the cycle repeats itself throughout the term, it is the cycle that is scheduled. The cycle is composed of a number of days and a number of periods (modules) each day.

Conflict is said to exist between time patterns when they overlap by one or more periods on any days. A conflict is said to exist between classes when the time patterns they are assigned conflict in any way. Conflicting classes pose a potential problem. No resource (instructor, room, nor student) can be assigned to conflicting classes without coming up against the overlapping hours. One of the central objectives of school scheduling is to cope with conflicting classes.

3.2. System Requirements

The purpose of System Requirements Analysis is to obtain thorough detailed understanding of the business need and is referred as a **Functional Requirement**. Requirement analysis deals with the description, analysis and prioritizing futures of the new system. Another type of requirement specifies something about the system itself, and how well it performs its functions. Such requirements are often called **Non-functional requirements** or performance requirements. Requirements definition is a careful assessment of the system needed, based on current and foreseen conditions

3.2.1. Functional Requirements of the System

Functional requirement is a description of activities and services that a system must provide. It is frequently identified in terms of inputs, outputs, processes and stored data that are needed to satisfy the system improvement objects.

We identified the following functional requirements to be embodied in the proposed new system.

- The system should determine the number of students of each section,
- The system should assign instructor for each section,
- The system should assign courses for each section,
- The system should display forms of classroom schedule and exam schedule, when after the appropriate user enters a password and user name,
- The system should display error warning message when the user makes mistakes.

3.2.2. Non-Functional Requirements of the System

Non-functional requirements are typically performance-level rather than processing requirements of a system. Non-functional requirements are imposing constraints on the design or implantation of the system.

- The system must be user friendly and it should be graphical user interface,
- The system should be able to accommodate users at different physical location,
- The process of classroom and exam scheduling must takes short time,
- The system must be compatible to satisfy users,
- The system must be secure.

3.3. Functional Modeling

3.3.1. Use Case Model

Use case model describes what a system does, without describing how the system does it. It reflects the view of the system from the perspective of a user outside of the system.

A use case model partitions system functionality into behaviors (use case) that are significant to the user of the system (actors).

3.3.2. Actors

An actor represents anything that needs to interact with the system to exchange information. Since actors can trigger system activity, it can be user or external system.

Our new automated system has three actors, named

- Administrator (Admin),
- Student,
- Instructor,

Table 2: Actors Description

NO.	Actor Name	Description
1.	Admin	# Full time employee of the school # Responsible for generating and controlling the system
2.	Student	# Customer of the school # Registered for the appropriate course and section. # Attend in the class according to the given schedule # To take final exam must attend 80% of the class # Full-time employee of the school
3	Instructor	# Full-time or par-time employee of the school # Prepare course outline, exams and according to the given schedule, begin and finished the class on time.

3.3.3. Use Cases

The new automated scheduling system of Tikur Anbessa Senior Secondary has two major system functionality which are classroom and exam schedule. To manage properly these two system functionality we create eight use cases.

The creation of use cases has proven to be excellent techniques in order to better understand and document system requirements. The name of each use case and the respective description are shown below.

Table 3: Case Name and Description

	Use Case Name	Description
1.	Generate classroom schedule	# assign instructor, course and section to the specified classroom
2.	Generate exam schedule	# assign instruction, classroom and sector to the specified exam course
3.	Access registration information	# Gather required information for the system from registration personal.
4.	Modification	# Works on required modifications
5.	View classroom schedule	# Access the classroom schedule program
6.	View exam schedule	# Access the exam schedule program

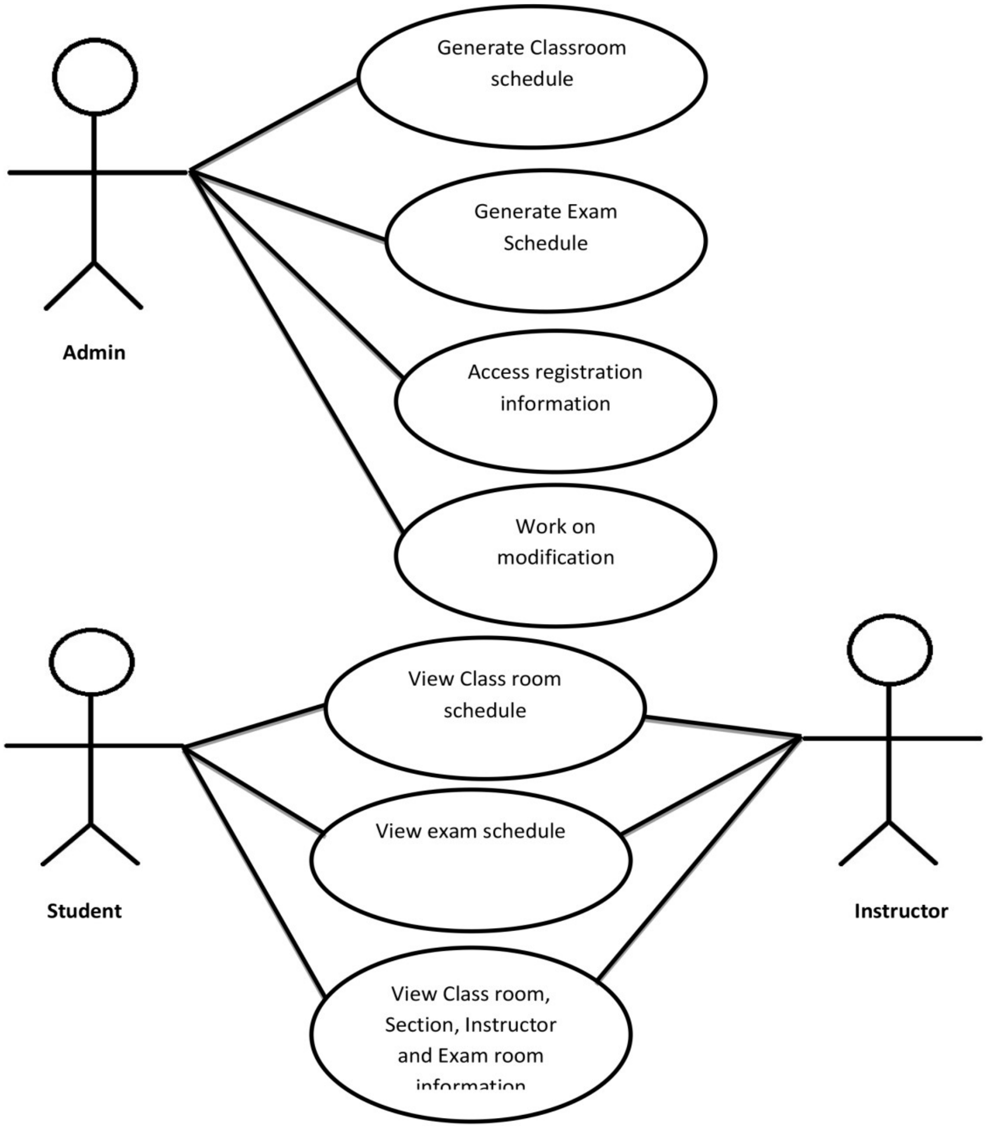

Figure 3.1: Use case diagram

3.3.3.1. Use Case Description

Use case describes sequence of events and actions. An action initiates an event happen and that triggers a use case; then the use case performs the action triggered by the event. Use case identified; expand it to include the use case typical courses of events and its alternative course. A use cases typical course of events in a step-by-step description starting with the action initiating the use case and counting until the end of the business event.

We describe only the basic course of action to show the sequence of action.

Table 4: Student Needs Information about Classroom Schedule

Use case Name	Classroom Schedule	
Actors	Student Admin	
Description	student interact with Admin to get a print copy of classroom schedule	
Triggering events	student enters inquires about classroom schedule	
Basic Course of action	**Steps**	**Action**
	1.	student initiates classroom schedule with admin
	2.	admin agent asks student id card
	3.	Admin asks student registration slip
	4.	student indicate section, name, Id and course title
	5.	admin check if the course is found on the schedule table
	6.	admin enters student information
	7.	admin print the classroom schedule information of the specified course.
	8.	admin provide the print copy to the student
	9.	student receives the printed paper of classroom schedule

Table 5: Instructor Needs Information about Exam Schedule

Use case Name	Exam Schedule	
Actors	Instructor admin	
Description	Instructor interact with admin to get a print copy of exam schedule	
Triggering events	Instructor enters inquires about exam schedule	
Basic Course of action	**Steps**	**Action**
	1.	Instructor initiates exam schedule with admin
	2.	Admin ask instructor id card
	3.	Instructor indicate course title
	4.	Admin checks if the course is found on the schedule table.
	5.	Admin enters instructor information
	6.	Admin prints the exam schedule information of the specified course.
	7.	Admin provide the print copy to the instructor
	8.	Instructor receives the printed paper of exam schedule information.

3.3.3.2. Use Case Scenarios

Use case consists of a standard flow of events within the system that describes a standard system behavior; use case scenarios are more detailed variations of the behavior. They are exceptions to the main behavior described by the primary use case.

- Instance actors -Instructor –Admin.
- Use case - Classroom Schedule.
- Use case Scenario
- Instructor needs information about classroom schedule.
- Instructor interacts with admin to get print copy of the classroom schedule.
- Admin ask ID card and verify.
- If it is not correct the interaction between instructor and admin will stop.
- If it is correct the interaction between instructor and admin will continue.
- Instructor indicate course title and instructor id.
- Admin check whether the course is found on the classroom schedule table or not.
- If it is not found the interaction between instructor and admin will stop.
- If it is found the interaction between instructor and admin will continue.
- Admin enter instruction information and give the print copy of classroom schedule of the specified course to the instructor.
- The interaction between instructor and admin on classroom scheduling is finished.

3.3.4. Sequence Diagrams

Sequence diagrams can illustrate a succession of interactions between object instances over time. Sequence diagrams are often used to illustrate the processing described in use case scenarios. In practice, sequence diagrams are derived from use case analysis and are used in system design to derive the interactions, relationships, and methods of the objects in our system.

Sequence diagrams graphically depict how objects interact with each other via message in the execution of a use case on operation.

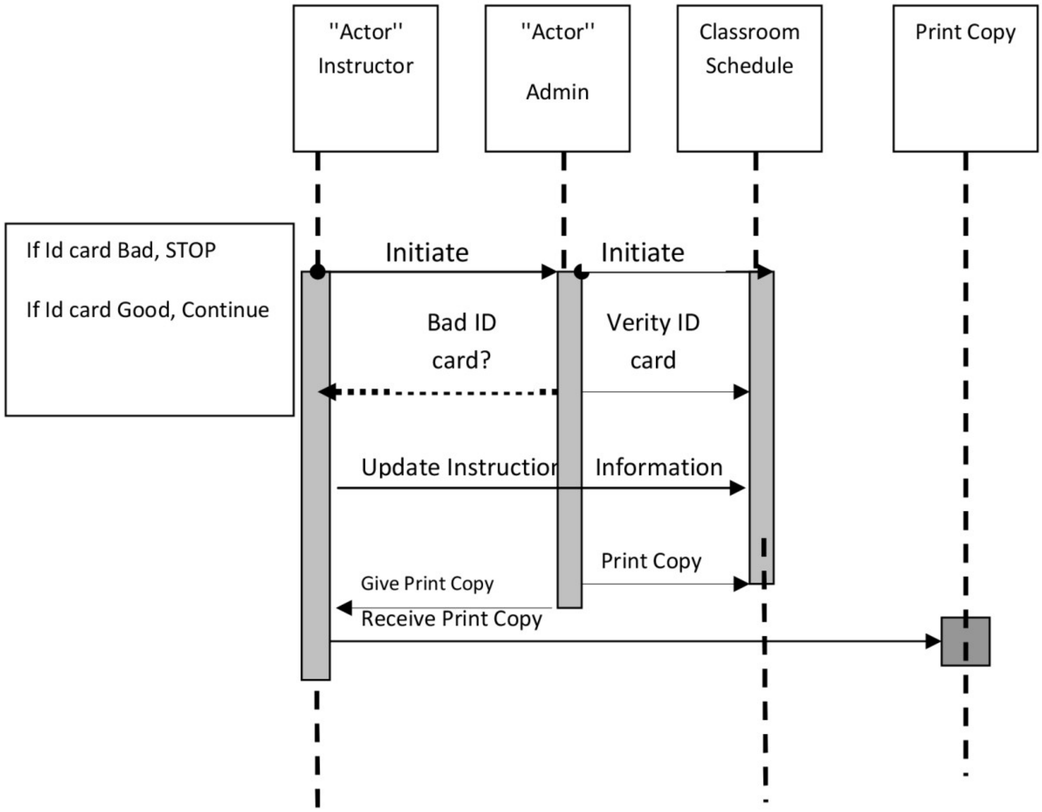

Figure 3.2 Sequence diagram of instructor and admin interact on classroom schedule

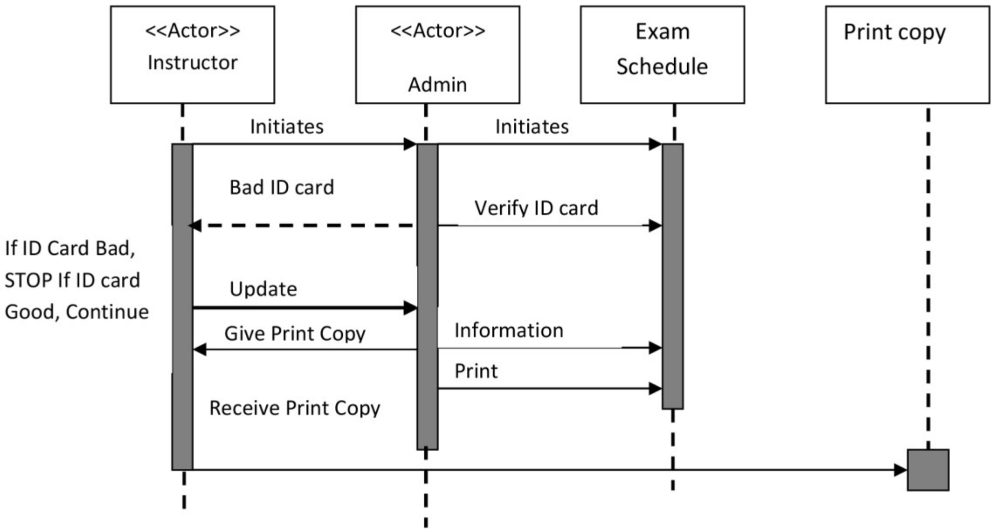

Figure 3.3 Sequence Diagram of Instructor and Admin Interact on Exam Schedule

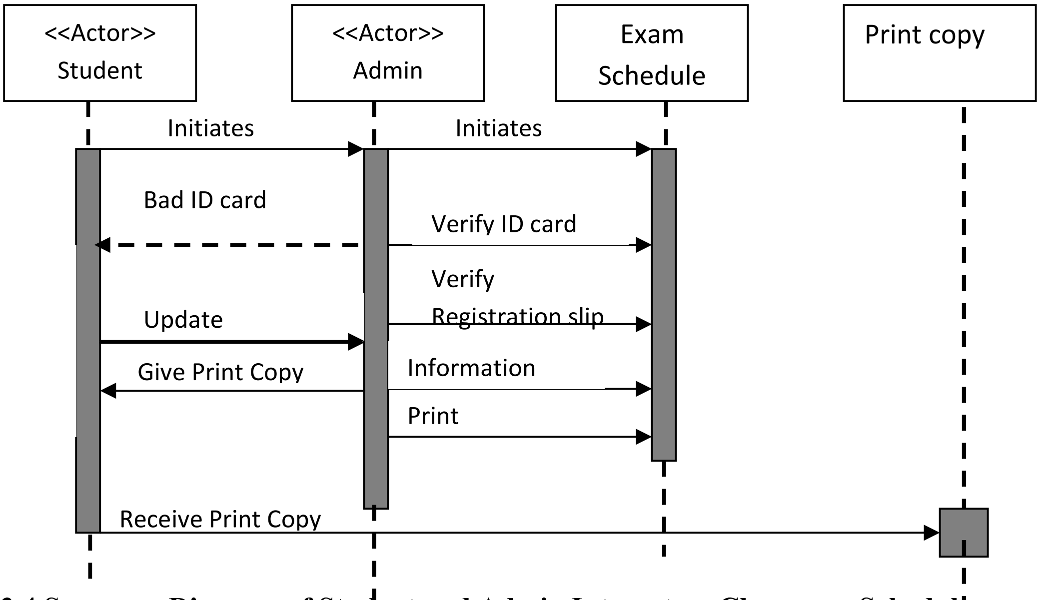

Figure 3.4 Sequence Diagram of Student and Admin Interact on Classroom Schedule

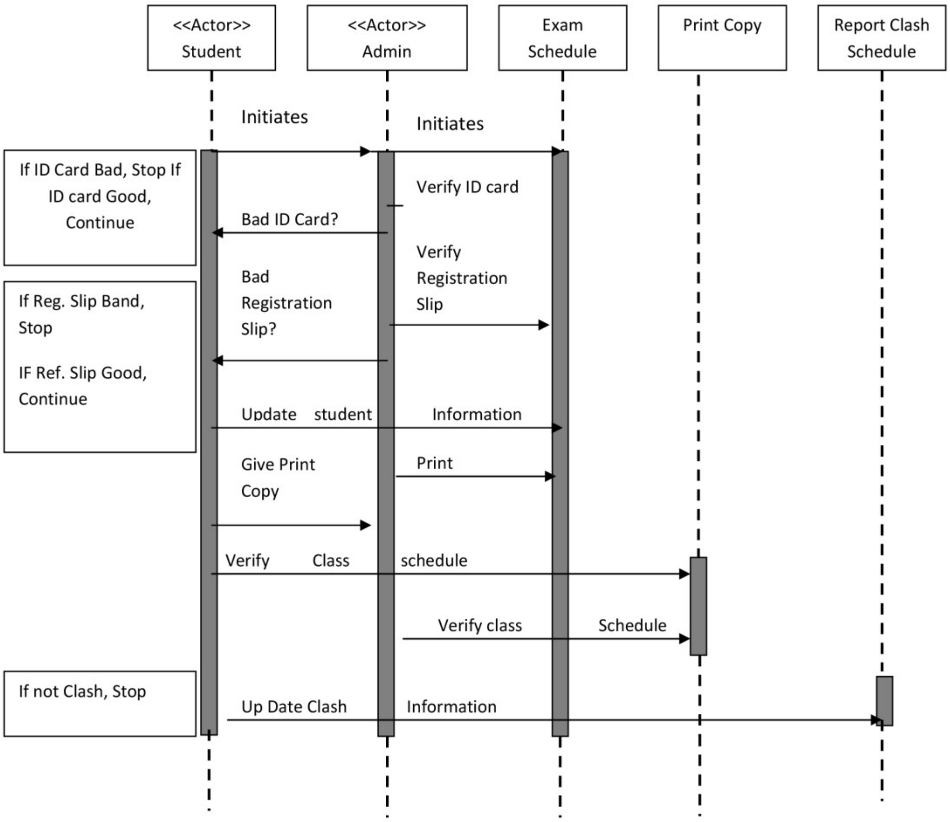

Figure 3.5 Sequence Diagrams of Student and Admin Interaction Exam Schedule

3.4. Project Requirements

Project requirement make an important role to develop the scheduling system. The project will include software requirement and hardware requirement that will be used to develop the system.

3.4.1 Software Requirements: Table 6

Requirement	Description
Microsoft word	Documentation
Net beans 8.0	code writing
Adobe reader	Documentation
C#	code writing
Visual Studio 2012	code writing
Microsoft SQL Server 2008	code writing

3.4.2. Hardware Requirements: Table 7

Requirement	Minimum system requirement
Laptop(optional)	Processor speed: 2.10GHz, 4GB RAM, 10GB free space
Desktop computer (optional)	Processor speed: 2.10GHz, 4GB RAM, 10GB free space
Requirement	**Description**
Power source	Generating power
printer	Printing
A4-paper	Printing

3.5. Behavioral Modeling

Computer memory, sometimes called storages, is one of the most valuable resources requiring allocation in a computer system. This particular resource is highly interchangeable, is required in differing configurations by literally every job in the system, and is the object of some contention. Beyond the questions of when to allocate memory, and how much, a major issue is where to allocate it. Location is important because, once allocated, that commitment affects all subsequent allocations. The potential problem is the danger of obstructive fragmentation, whereby subsequent requests for contiguous memory are frustrated by the particular location of an earlier commitment which has fragmented available memory into discontinuous blocks too small to support the new requests. This is particularly unfortunate when the total space is adequate, but the individual pieces are too highly fragmented. In modern computer systems, there is an eventual process of deal location attending release of the memory resource, and in such a dynamic storage environment, the avoidance of obstructive fragmentation is even more important than in the permanent allocation environment.

The important observation for us regarding computer memory allocation, either permanent or dynamic, is to note what must be managed: the remaining space after each new allocation is committed. The objective is to maximize the utility of this remaining space. It is good to retain as much flexibility as possible for future use, and bad to so fragment the space as to limit its subsequent use. Incidentally, the profile of anticipated demands on remaining space is an important factor; if certain kinds of demand are the most likely, they may direct the way in which we make our decisions.

The analogy to school scheduling is very important. Every time a portion of the cycle is allocated to a class and its resources, it eliminates and preempts that portion of the cycle from subsequent use by those resources. The remaining portion of the cycle must be considered; subsequent allocations must occur in that remaining space. Particularly because of the variety of demands represented by the different resources involved, it is critical that the remaining space retains as much flexibility as possible for future use.

Allocation of cycle space is complicated by the non-contiguous and diverse time pattern shapes usually encountered in an academic environment. The more complex and disperse a time pattern is, the greater the danger of its contribution to obstructive fragmentation. As classes and their time patterns are assigned to a resource, the individual schedule of that resource is transformed from free space covering the entire cycle to the eventual schedule with most (if not all) of the time reserved. Throughout this cumulative process, each time a new class and its time pattern are assigned, the original free space is further and further diminished. Often, this remaining space is fragmented into holes, some of which may be permanently unusable for any further assignment. It is therefore very important to try to retain flexibility wherever we can, lest we implicitly preempt more cycle space than a class explicitly requires.

The shape and orientation of time patterns, the dimensions of the cycle itself, and the ways in which time patterns interact with each other are clearly critical to the cumulative cycle space allocation process, and therefore, to the overall school scheduling process.

Before concentrating on the time variables involved in school scheduling, it is useful to understand a few of the approaches that can be taken to the problem as a whole. The word "approach" is a useful one, since a schedule is often the net result of a series of successive approximations. Trial and error methods are normally employed over a number of iterations perhaps over a number of years with the dual purposes of seeking improvement and of gaining insight into the systems nature of the school being scheduled.

There are two opposite directions from which the eventual schedule for a school can be approached. A school can try to accomplish all its goals at once. It can specify a schedule with all the "bells and whistles", reflecting all the resource preferences as well as the constraints. In other words, a school can take an optimistic ambitious approach. The problem usually is that such a highly desirable schedule often lacks feasibility. A school cannot have all of its instructors teaching between 11:00A.M. and 2:00 P.M., with Friday afternoons off, and still expect to come up with decent room utilization!. The opposite direction of approach is to start with modest expectations, requiring only the most serious constraints. In other words, a school can take a conservative step by step approach. The problem here is that such a schedule, if feasible, is often less than desirable.

Despite difficulties with either direction of approach, the conservative approach is usually the best, because it deals from a fallback position of feasibility. When an additional goal is superimposed on the system and it succeeds, it can be incorporated. But should it fail, the school still has the previous feasible schedule. Often when too much is asked of a schedule, and it shows to be infeasible, there is little information as to which of the myriad niceties should be compromised. There are psychological advantages to having a tangible schedule in hand

Of course, any schedule should possess some degree of realism. It may be wishful thinking to temporarily ignore room assignments in the hope that "they can be added in later", or to blindly extrapolate an entire schedule from one that serves a single student year. Then again, there is a certain value to negative information. If a school cannot be feasibly scheduled even without rooms, the problem won't go away when the rooms are added.

4. System Design

4.1. Introduction

This section outlines the system design and the object design, specification of the automated scheduling system of Tikur Anbessa Senior Secondary School.

The purpose of the system design is that this document shows, first, which would briefly outline the reason the system is developed, and design goals, second, which would specify the qualities of the software that developers should optimize, the current system status regarding to software architecture and why we need to change it, third, the document show the architecture of the newly proposed system. Other main components of the system such as hardware/ software mapping, persistent data managements, access control polices, and boundary conditions would be addressed.

4.2. Design Goals

Design goals describe the qualities of the system that developers should optimize. Such goals are normally derived from the non-functional analysis document. Recall that, the non-functional requirements set in the analysis document are to have consistent interface formats and button sets of all forms in the application, to have a form based interface for all data entry and viewing formats, and to generate reports that are formatted in a table and that should look like the existing manual report formats.

In addition to what is outlined in the analysis document, which outlines the end users criteria, the design goals are grouped into four additional categories.

Performance: - The system should have a fast response time with maximum through put. Furthermore, the system should not be taking up too much space in memory.

Dependability: - The organization need the system to be highly dependable as it is expected to be used by any user. The system should be fault tolerant. As the system handling sensitive data of the organization, high emphasis should be given with regards to security.

Cost: - The system should be developed with as minimum cost as possible. However, then it comes to compromising between initial cost and maintenance costs, the organization prefer to invest more initially to develop bug free software and reduce maintenance and support costs at a later stage.

Maintenance: - The system should be easily extensible to add new functionalities at a later stage. It should also be easily modifiable to make changes to the features and functionalities.

4.3. Proposed Software Architecture

The proposal system is expected to replace the existing manual system into an automated system. It would be designed using object oriented software engineering techniques.

The proposed system mainly deals with sub system decomposition dividing the system into manageable components. Another major task in system design deals with hardware and software mapping that deals with which components would be part in which hardware. Access control and security policy issues are also addressed in this part.

4.3.1. Decomposition of the System

A complex system is usually too difficult to fully understand when view as a whole. Therefore, in systems analysis we separate a system into component subsystems, which are decomposed into smaller subsystems, until we have identified manageable subsets of the overall system.

Decomposition is the act of breaking a system into its component subsystems, process and sub processes. Each level of abstractor reveals more or less detail about the overall system or a subset of that system.

Break a system into small, manageable, and understandable subsystems will help us to:

- focus attention on one area or subsystems at a time, without interference from other area;
- concentrate on the part of the system pertinent to a particular group of users, without confusing users with unnecessary details; and
- build different parts of the system at independent times and have the help of different analysis.

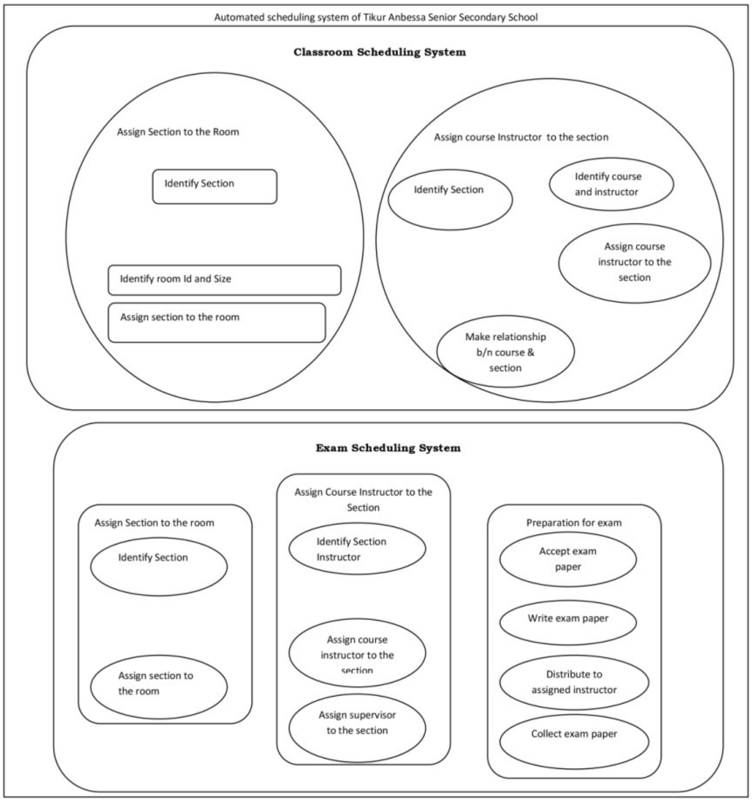

Figure 3.6. The automated scheduling system of Tikur Anbessa Senior Secondary School consists of the above subsystems and processes

A decomposition diagram is essentially a planning tool for more detailed process models.

- Each process in a decomposition diagram is either a parent process, a child process or both.
- A parent must have two or more children; a single child does not make sense because that would not reveal any additional details about the system.
- In most decomposition diagramming standards, a child may have only one parent.
- Finally, a child of one parent may be the parent of its own children.

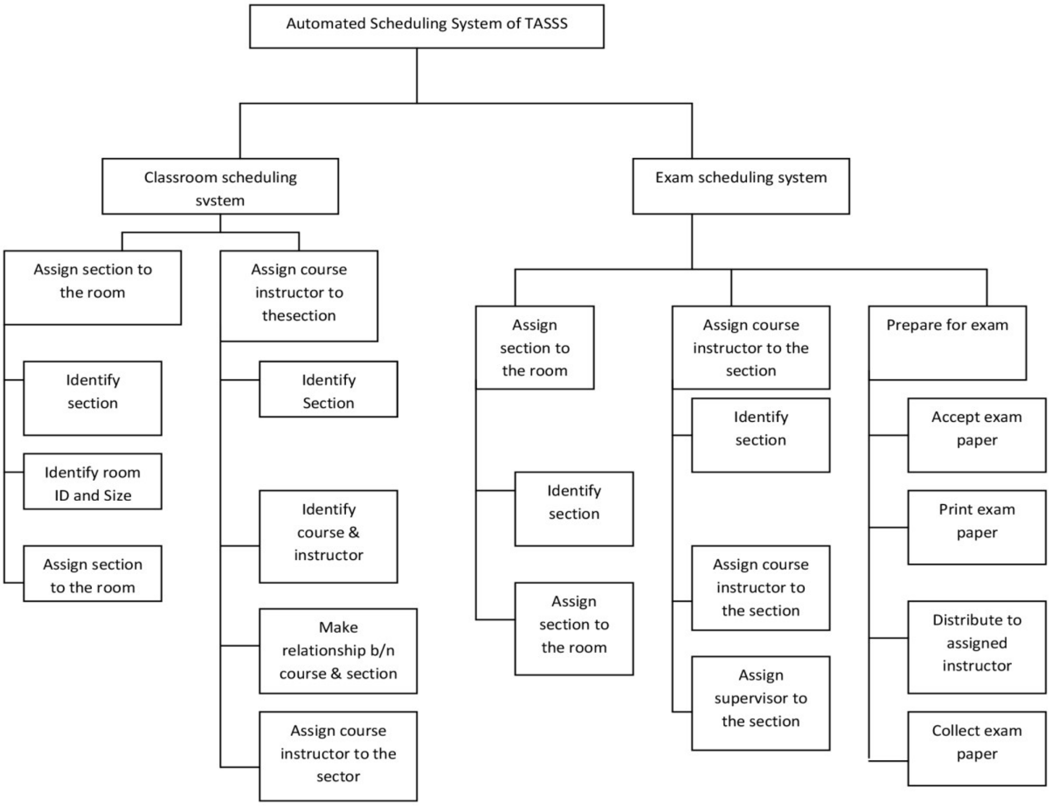

Figure 3.7. Decomposition diagram

4.4. Attribute and Method Description

4.4.1Attribute Description

Table 8: Attribute Description for Instructor

Attribute Name	Description
Instructor ID	Its Uniquely identify the instructor
First Name	Indicate the first Name of the Instructor
Last Name	Indicate the last name of the Instructor
Gender	Indicate sex of the instructor
Home phone	Indicate home phone number of the instructor
Mobile Phone 1	Indicate mobile phone number of instructor
E-mail	Indicate e-mail address of the instructor

Table 9: Attribute Description of the Classroom schedule

Attribute Name	Description
Room Id	Uniquely identify the Room
Section Id	Uniquely identify the section
Course Id	Uniquely identify the course
Building No	Uniquely identify the Building
Instructor Id	Uniquely identify the instructor

Table 10: Attribute Description of exam schedule

Attribute Name	Description
Room Id	Uniquely identify the room
Section ID	Uniquely identify the section
Course Id	Uniquely identify the course
Building No.	Uniquely identify the building
Supervision ID	Uniquely identify the supervisor
Instructor Id	Uniquely identify the instructor

Table 11: Attribute Description for Classroom

Attribute Name	Description
Room ID	Uniquely Identify the room
Building No	Uniquely identify the building
Room Capacity	Indicate number of student that the room can hold

Table 12: Attribute Description for Section

Attribute Name	Description
Section Id	Uniquely identify the section
Section Name	Identify the section
No of Student	Indicate number of student that the section hold

4.4.2 Method Description

Table 13: Method Description for register agent

Method Name	Description
Initialize (2)	Initialize the system
Update	update information
print copy	print exam and classroom schedule paper
Exam schedule	Prepare exam Schedule
Classroom schedule	Prepare classroom schedule
Attend class	control attendance of the class

4.5. Constraints of attribute and method

Table 14: Attribute Constraints for instructor

Attribute Name	Constraint
Instructor ID	Not Null
First Name	Not Null
Last Name	Not Null
Gender	Not Null
Home Phone	Null
Mobile Phone 1	Not Null

Table 15: Attribute Constraints for Student

Attribute Name	Constraint
Student Id	Not Null
First Name	Not Null
Last Name	Not Null
Gender	Not Null

Table 16: Attribute Constraints for Classroom

Attribute Name	Constraint
Room Id	Not Null
Building ID	Not Null
Room Capacity	Not Null

Table 17: Attribute Constraints for Section

Attribute Name	Constraint
Section ID	Not Null
Section Name	Not Null
No of Student	Not Null

Table 18: Attributes Constraints for classroom Schedule

Attribute Name	Constraint
Room Id	Not Null
Section ID	Not Null
Course ID	Not Null
Instructor Id	Not Null

Table 19: Attribute constraints for exam Schedule

Attribute Name	Constraint
Room Id	Not Null
Section ID	Not Null
Course ID	Not Null
Supervisor ID	Not Null

4.6. Database Design

Data becomes the central resource in a database environment. Information systems are built around this central resource to give both computer programmers and end users flexible access to data. Database design is concerned with the data focus from the perspective of the system designer. The end result is called database schema, a technical blue print of the database. Database is a set of interrelated files. This means that records of one file are related with the recorded of the different file. Database management system is specialized computer software available from computer venders that is used to create access, control and manage the database. The core of the database management system is often called a database engine. The engine responds to specific commands to create database structures and then to create, road, update and delete records in the database. The purpose of normalization is to eliminate insertion, deletion and update problem in the table.

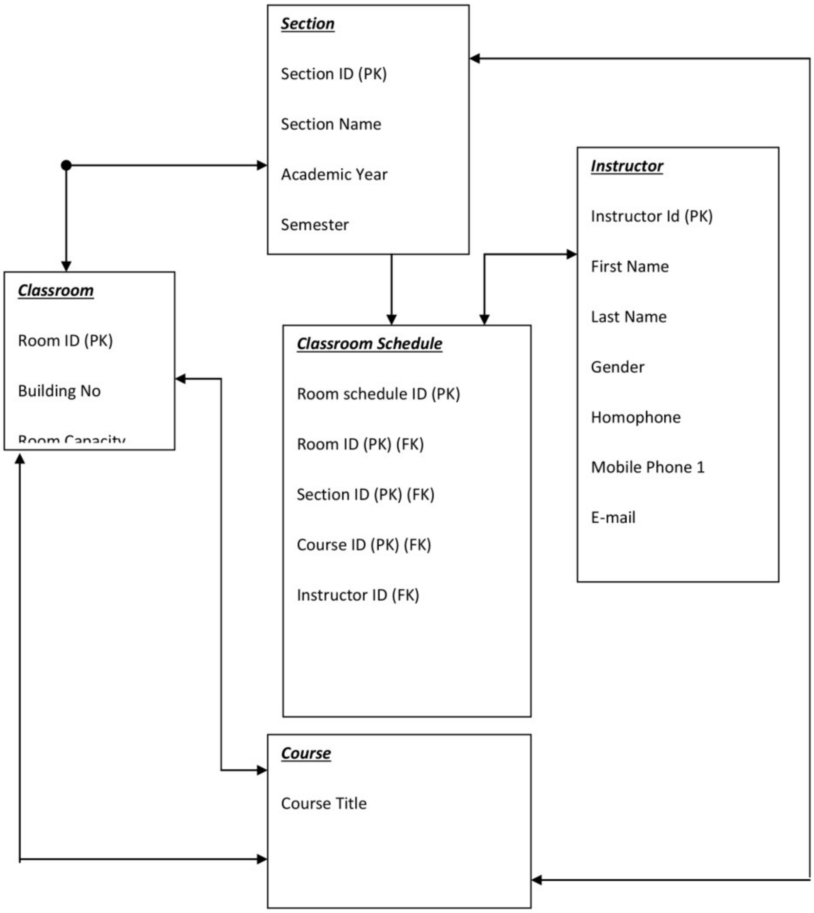

Figure 3.8: The Normalized Database Design

Table 20: Instructor Table

Field Name	Data Type	Field size	Required
Instructor Id	Integer	5	Not Null
First Name	text	50	Not Null
Last Name	Text	50	Not Null
Gender	text	1	Not Null
Home Phone No	text	20	null
Mobile Phone 1	text	20	Not Null
E-mail	text	20	Null

Table 21: Course Table

Field Name	Data Type	Field Size	required
Course title	text	20	Not Null

Table 22: Classroom Table

Field Name	Data type	field size	Required
Room ID	Text	10	Not Null
Room Capacity	Integer	2	Not Null

Table 23: Section Table

Field Name	Data type	field size	Required
Section ID	text	10	Not Null
Section Name	text	20	Not Null
No of Student	Integer	2	Not Null

Table 24: Classroom Schedule Table

Field Name	Data type	field size	Required
Room ID	text	10	Not Null
Section ID	text	10	Not Null
Course ID	text	10	Not Null
Instructor ID	text	10	Not Null

Table 25: Exam Schedule Table

Field Name	Data type	field size	Required
Room ID	text	10	Not Null
Section ID	text	10	Not Null
Course ID	text	10	Not Null
Supervisor ID	text	10	Not Null

4.8. Flow Charts

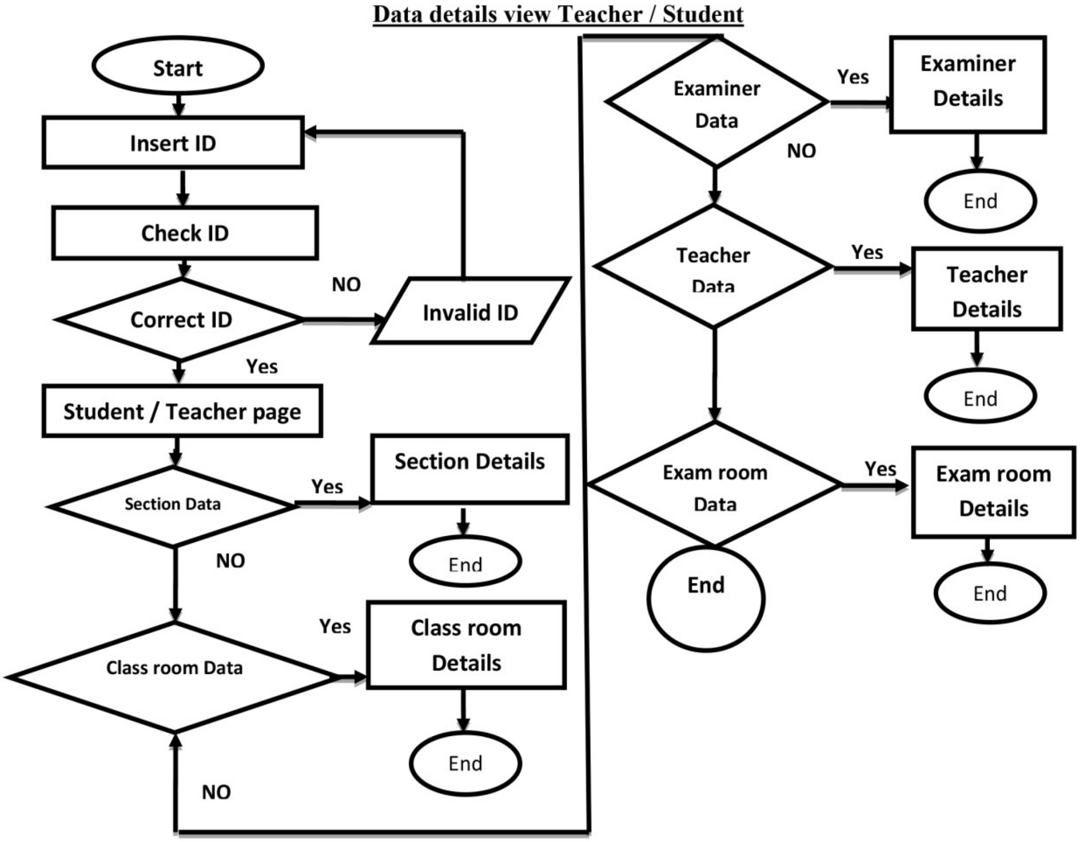

Chart 1 Data details view Teacher/student

5. Conclusion and Recommendation

5.1. Conclusion

In the introductory chapter, it was stated that the purpose of the documentation is "to draw the attention of school authorities to the importance of careful time pattern analysis, and to assist these authorities in performing a proper time pattern analysis." In order to accomplish this purpose, the thesis concentrates on the definition and justification of time pattern analysis, involving both theoretical and practical considerations.

It is hoped that the thesis will have a beneficial impact in the academic environment in terms of affording:

(1) insight into the systems nature of school scheduling, in particular into the role and importance of variables involving time, and

(2) opportunities for, and justification of, improvements to schedules, through time pattern analysis.

In conclusion, the world is heading toward automated information systems and a complex information resource usage. We believe that it is time to realize the need for modern information system and work towards making the most of it. The decision to have one as soon as possible will determine the schools future success.

5.2. Recommendation

In the present world of business and information, failure to have a right kind of Information System can be fatal to businesses and institutions. Information in our day is a very sensitive and decisive issue in our businesses, homes and communities. To survive in a very dynamic business world or in a very tense competition, the role of information system is undoubtedly important and necessary.

That is why we believed that Tikur Anbessa Senior Secondary School can benefit from an automated and modern information system. This kind of information system is highly recommended for better performance, efficiency, effectiveness and competitiveness. This project has been developed using modern techniques up-to-date technologies and experiences. This kind of information system is not only a support for the institution's work but can also be considered as a highly valuable asset for the staff and students.

Finally we would like to address other professionals who are interested in this particular system that wants' to upgrade it, to focus their main attention on reducing or completely removing the need of human contact in the process of preparing the schedule. However, the system we developed needs some degree of human support to make the system fully functional.

Reference

Adaptive Software Engineering G22.3033-007 Session 8–Sub-Topic 3. Presentation Object-Oriented Design Guidelines Dr. Jean-Claude Franchitti

Alexis Leon, Mathews Leon. Fundamental of Database Management System, L and L Consultancy service pvt. Ltd, India (1976)

Course module for Networking Technology.

Institute Timetable B.Tech. Project Report 1 Saeed Mirza (92228) Final Year B.Tech. Department Of Computer Science and Engineering I.I.T. Kanpur Project Guide: Dr. Dheeraj Sanghi 8th April 1996

Irwin. Database Management System, Designing and Building Business Applications, 1st ed, McGraw-Hill companies, USA (1976)

James F. Blakesley, Keith S. Murray, Frederick H. Wolf, and Dagmar Murray. Space management and Academic Scheduling Purdue University West Lafayette, IN 47907-1128 {jfb, dmurray}@purdue.edu

Jefery A.Hoffer, Modern Systems Analysis and Design, Forth ed., ISBN Bangladesh, India, published by Dorling Kindersley, 2005

Patrick G. Mekeown. Living with computers, The Dryden press Harcourt Brace collage publisher, printed in the USA (1977)

Tata McGraw-Hill, 7West Patel Nagar. System Analysis Design Methods 5thed, New Delhi 110 008 Division of the McGraw-Hill Companies