ACKNOWLEDGEMENT

First and foremost our sincere thanks to the Almighty for giving us the strength, skill and perseverance to complete this interim report successfully.

We express our heartfelt gratitude to Dr. J. Isaac, the principal of RSET and to Prof. Ajith Kumar CS, the Head of the Department of Computer Science Department of RSET for providing us with the means and facilities for the completion of the seminar.

We wish to extend our words of gratitude to Mr.Biju Abraham N, Asst. Professor, Department of Computer Science, RSET, who played the important role of guiding us throughout the entire seminar and enlightened us with the latest ideas applicable to the seminar.

We also express our wholehearted thanks to our project co-ordinator Mrs. Mintu Philip, Asst. Professor, Department of Computer Science, RSET without whose guidance we would not have completed the interim report.

We sincerely thank all other technical teaching and non-teaching staffs who extended their help at all points throughout the completion of the seminar.

Last but not the least, we take this opportunity to put across our deepest gratefulness to our friends and to our parents, whose encouragement and support helped them a lot for the successful completion of this seminar.

ABSTRACT

Timetable generation is an NP-complete problem .i.e. there is no specific algorithm which can be used for creating optimal solutions. As constraints for creating timetable vary from institute to institute, separate algorithms have to be defined for each institute considering the constraints imposed. Here, we design a timetable generator for our college Rajagiri School of Engineering and Technology for automatically generating timetable, considering all institution specifications. Teacher availability and class availability are the key issues that need to be solved.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

JDK-Java Development Kit

GUI-Graphical User Interface

JSP-Java Server Pages

NP Complete-Non Polynomial Type Complete

OOAD-Object Oriented Analysis and Design

OOD-Object-Oriented Design

RAM-Random Access Memory

UML-Unified Modeling Language

1. INTRODUCTION

1.1 PROBLEM STATEMENT

Project aims at automatic generation of time table for Rajagiri School of Engineering and Technology there by easing manual work which is involved in computation. Even though most college administrative works have been computerized, the timetable scheduling is still mostly done manually due to its inherent difficulties. The manual timetable scheduling demands considerable time and efforts. The lecture-timetable scheduling is a constraint satisfaction problem in which we find a solution that satisfies the given set of constraints. The college timetabling problem asks us to find some time slots and classrooms which satisfy the constraints imposed on offered courses, teachers, class rooms and so on.

Since the problem is a combinatorial optimization problem belonging to NP-hard class, the computation time for timetabling tends to grow exponentially as the number of variables increase. There have been a number of approaches made in the past decades to the problem of constructing timetables for colleges and schools.

1.2 PROJECT SCOPE/OBJECTIVE

Objective of the system is to generate time table schedule automatically, considering constraints like availability of teachers, number of subjects, number of hours per week required for each subject as specified in syllabus etc. Timetable Generation System generates timetable for each class, in keeping with the availability calendar of teachers, availability and capacity of physical resources and rules applicable at different classes, semesters, teachers and subject's level.

Overview of the Existing System

The current system of time table generation in Rajagiri School of Engineering and Technology is manual which involves lot of time and hand work for generating the B.Tech time table. Automation in this work will greatly reduce the time spent by college officials on

time table generation. This application reduces the manual work with automatic generation of time table taking required inputs from user.

Draw backs in existing system:

- Existing system was slow and man driven
- Efficient human effort and attention is required for good results
- Highly time consuming

Proposal for dissertation

We propose a system for efficiently computing timetable by checking the availability of teachers and avoiding clashes if any exists in any of the slots. The application randomly selects the subjects and allocates time slots. Then checks for clashes, eliminates it and finalizes the allocation. System also checks for constraints like number of time slots required for a subject, maximum number of lectures a teacher can handle, non allocation of continuous slots etc.

1.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

There are various constraints to be satisfied at the time to instantiate variables about time slots and classrooms. The constraints can be categorized into Hard and Soft constraints.

Hard Constraints

A timetable which breaks a hard constraint is not a feasible solution, and must be repaired or rejected by the timetabling algorithm. Hard constraints considered here are:

- No two teacher is allocated different slots for the same hour
- All subjects should be allocated number of time slots as specified in specified in the syllabus
- Teachers, classes, could not be assigned to periods at which they're not available.
- Teachers should not be allocated to continuous slots
- HODs should not be allocated to 1st and 3rd hours
- No subject should be always allocated first or last hours in a week for a class

- The teachers' busy slots read through input has to be considered (eg. If a teacher has an hour allocated to M.Tech he can make it a busy slot). Those slots or consecutive slots should not be allotted to them in our time table
- Pre assigned labs should not be reassigned to any other subject.

Soft Constraints

Soft constraints are less important than hard constraints, and it is usually impossible to avoid breaking at least some of them. Whichever timetabling method is applied, timetables are usually rated by a penalty function, which calculates the extent to which a timetable has violated its soft constraints. Some soft constraints are more important than others, and this is often specified with a priority value. Soft constraints are:

- Even distribution of subjects throughout the week
- There should not be more than 2 hours allocated for a single subject in a day
- Teachers' daily lecture hours should be restricted to be within the allowed maximum hours.

1.4 **DEVELOPMENT METHOD**

Development method used is Incremental model. The incremental model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping. The incremental model is an intuitive approach to the waterfall model. Multiple development cycles take place here, making the life cycle a "multi-waterfall" cycle. Cycles are divided up into smaller, more easily managed iterations. Each iteration passes through the requirements, design, implementation and testing phases.

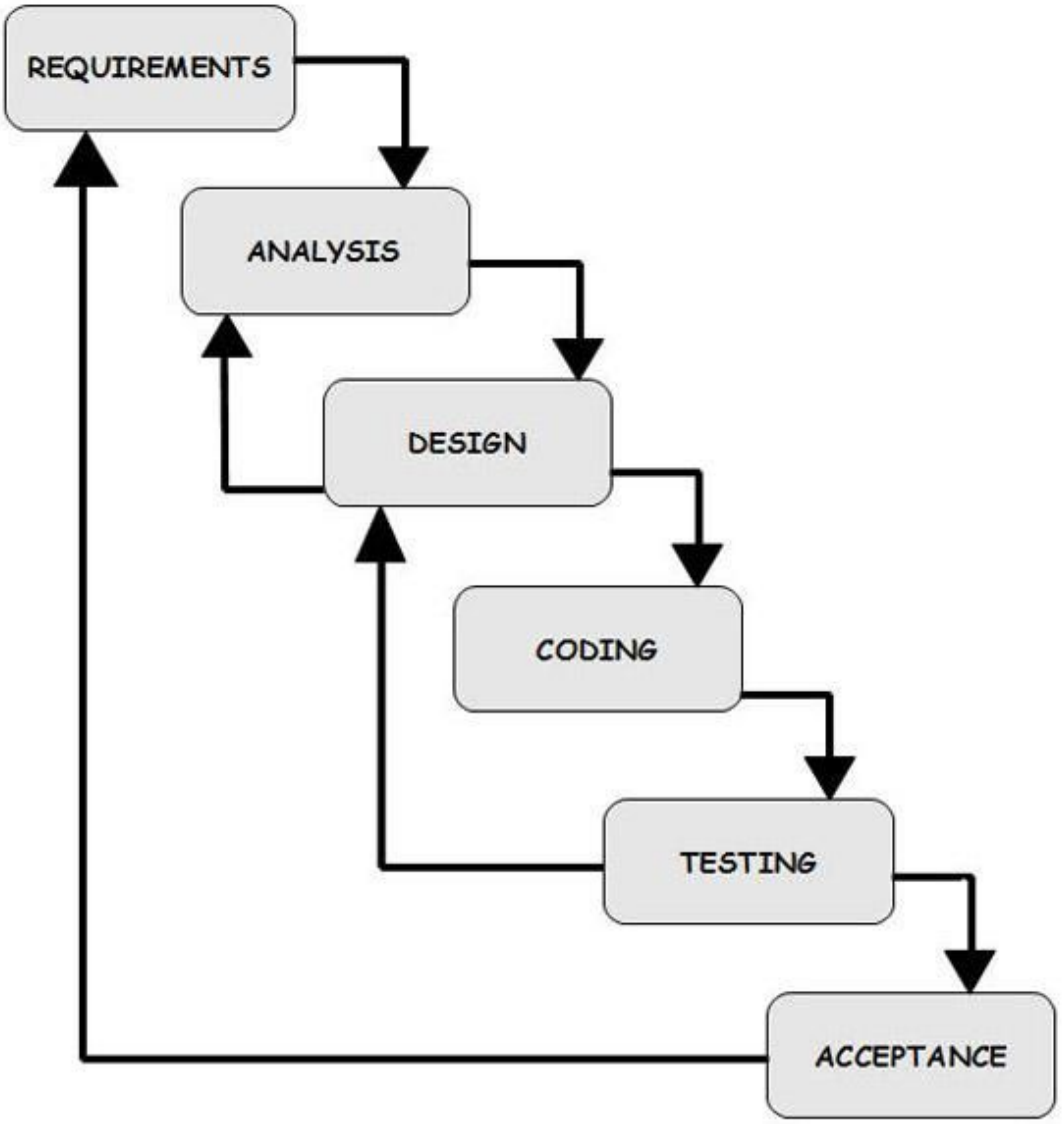

Fig1.5.1: Incremental model

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.

- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Object-Oriented Analysis and Design(OOAD) is a software development approach to design and implement software system as a collection of interacting stateful objects with specified structure and behavior. Here we use OOAD approach. **Object-oriented analysis (OOA)** is the process of analyzing a task (also known as a problem domain) to develop a conceptual model that can then be used to complete the task.

A typical OOA model would describe computer software that could be used to satisfy a set of customer-defined requirements. During the analysis phase of problem-solving, the analyst might consider a written requirements statement, a formal vision document, or interviews with stakeholders or other interested parties. The task to be addressed might be divided into several subtasks (or domains), each representing a different business, technological, or other areas of interest. Each subtask would be analyzed separately. Implementation constraints, (e.g., concurrency, distribution, persistence, or how the system is

to be built) are not considered during the analysis phase; rather, they are addressed during object-oriented design (OOD).

The conceptual model that results from OOA will typically consist of a set of use cases, one or more UML class diagrams, and a number of interaction diagrams. It may also include some kind of user interface mock-up.

2. SYSTEM SPECIFICATION

2.1 SYSTEM FUNCTIONS OR USE CASES

The project consist of three use cases:-

- Administrator view
- Student view
- Teacher view

Administrator will be responsible for formulating and allotting slots for different subjects. Student and Teacher view can be accessed by all the students and teachers for viewing their respective time table. Teachers and students are allowed just to view, not to update.

Administrator activities:

- Authenticated login
- Update the timetable.
- View timetable.

Teacher activities:

- View teachers timetable also class wise timetable.
- Can't update its Schedule, its responsibility only to Administrator.

Student activities:

- Only view class time table.
- Can't update its Schedule, its responsibility only to Administrator.

2.2 NON-FUNCTIONAL REQUIREMENTS

Security Requirements:

This project requires a database which stores user authentication details to avoid any unauthorized person to use the application. We don't enforce any other security steps. We expect user to ensure the security by giving passwords that are difficult to guess and change the passwords frequently.

3. SYSTEM DESIGN

3.1 ARCHITECTURAL OR STRUCTURAL DESIGN

SYSTEM ARCHITECTURE:

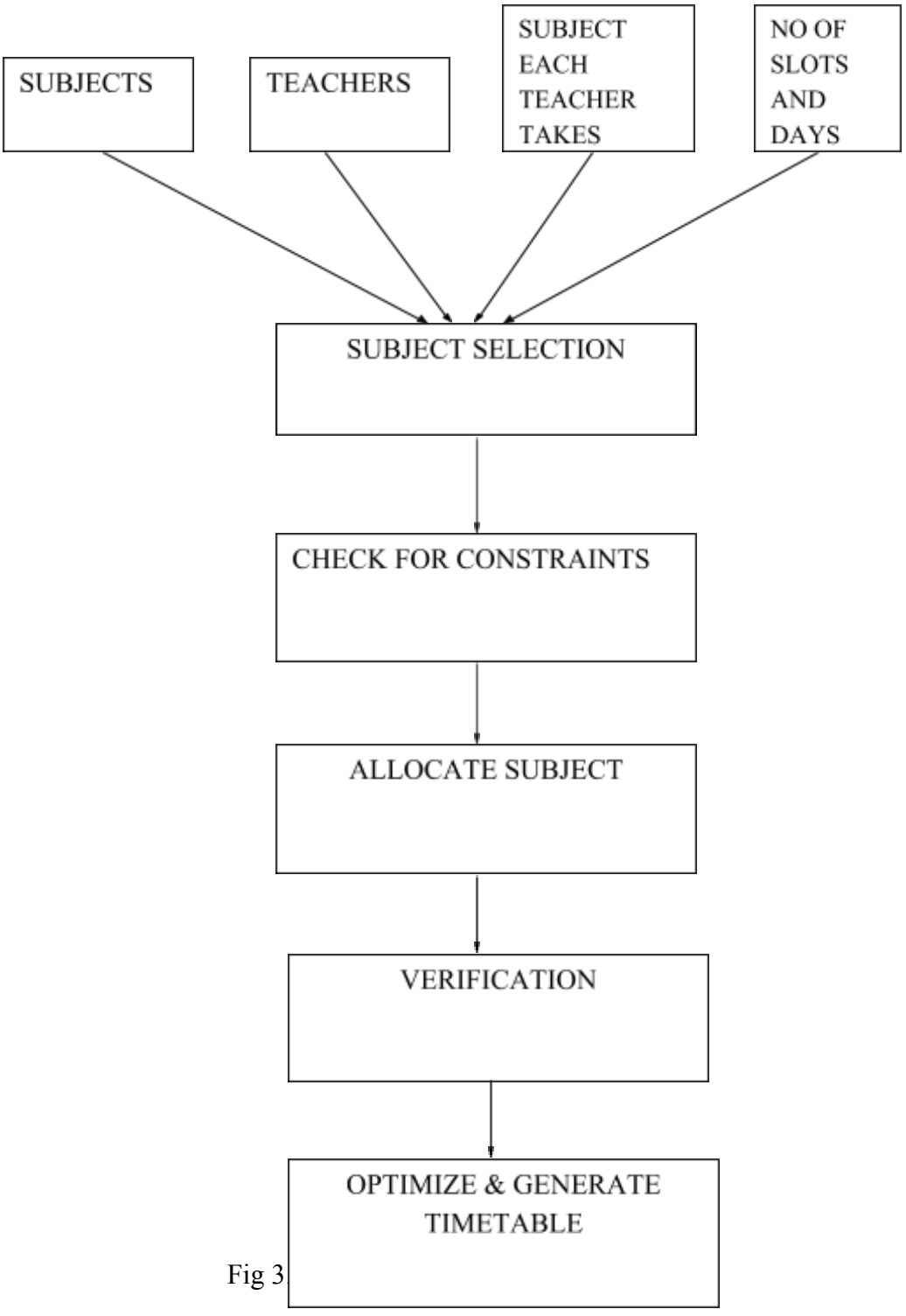

Fig 3

USE CASE DIAGRAM:

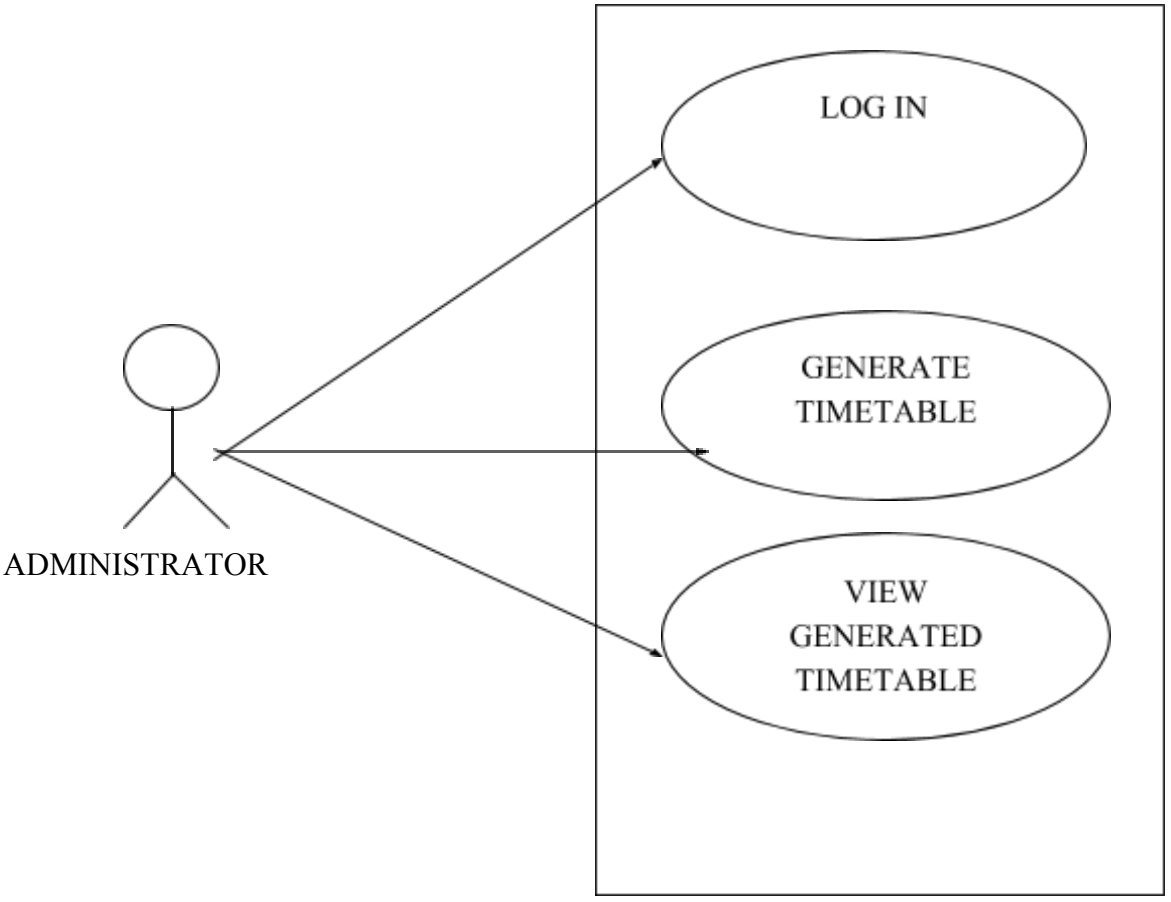

Fig:3.1.2 Use Case Diagram 1

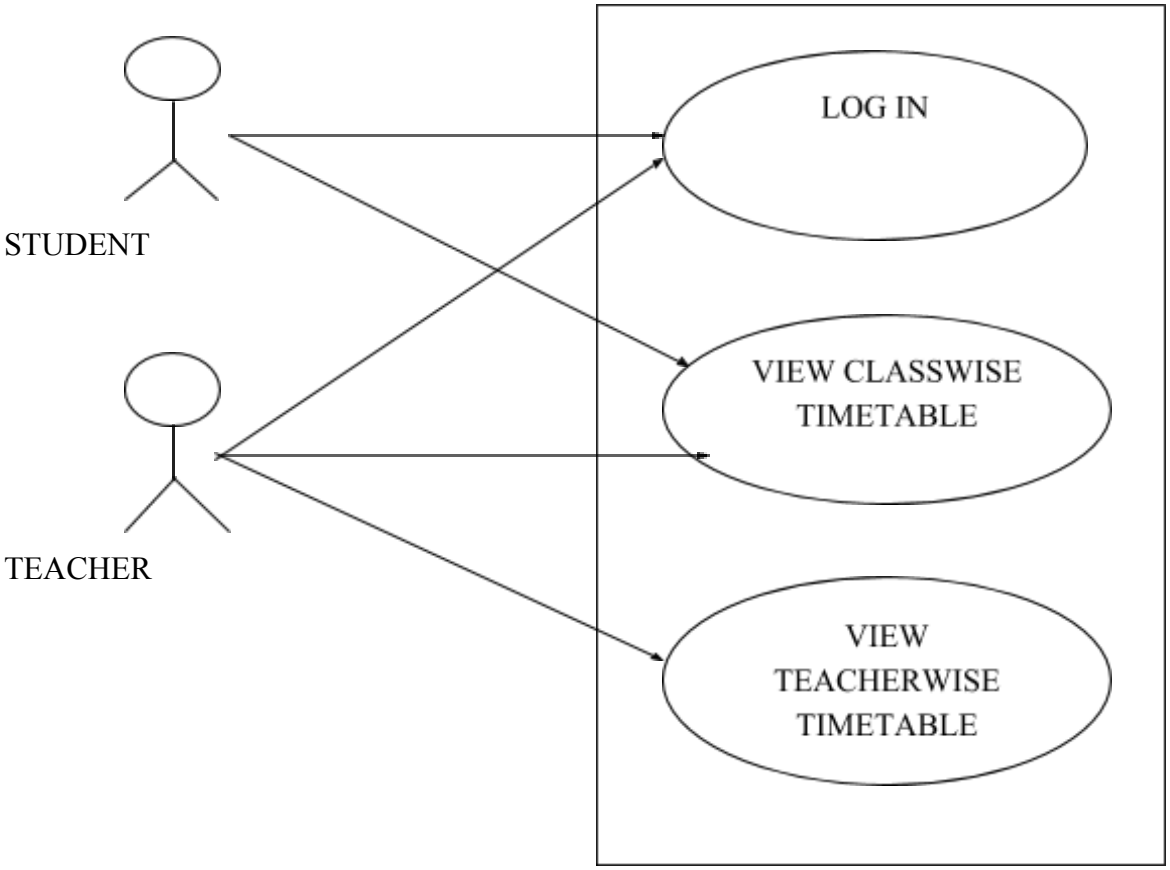

Fig:3.1.3 Use Case Diagram 2

ACTIVITY DIAGRAM:

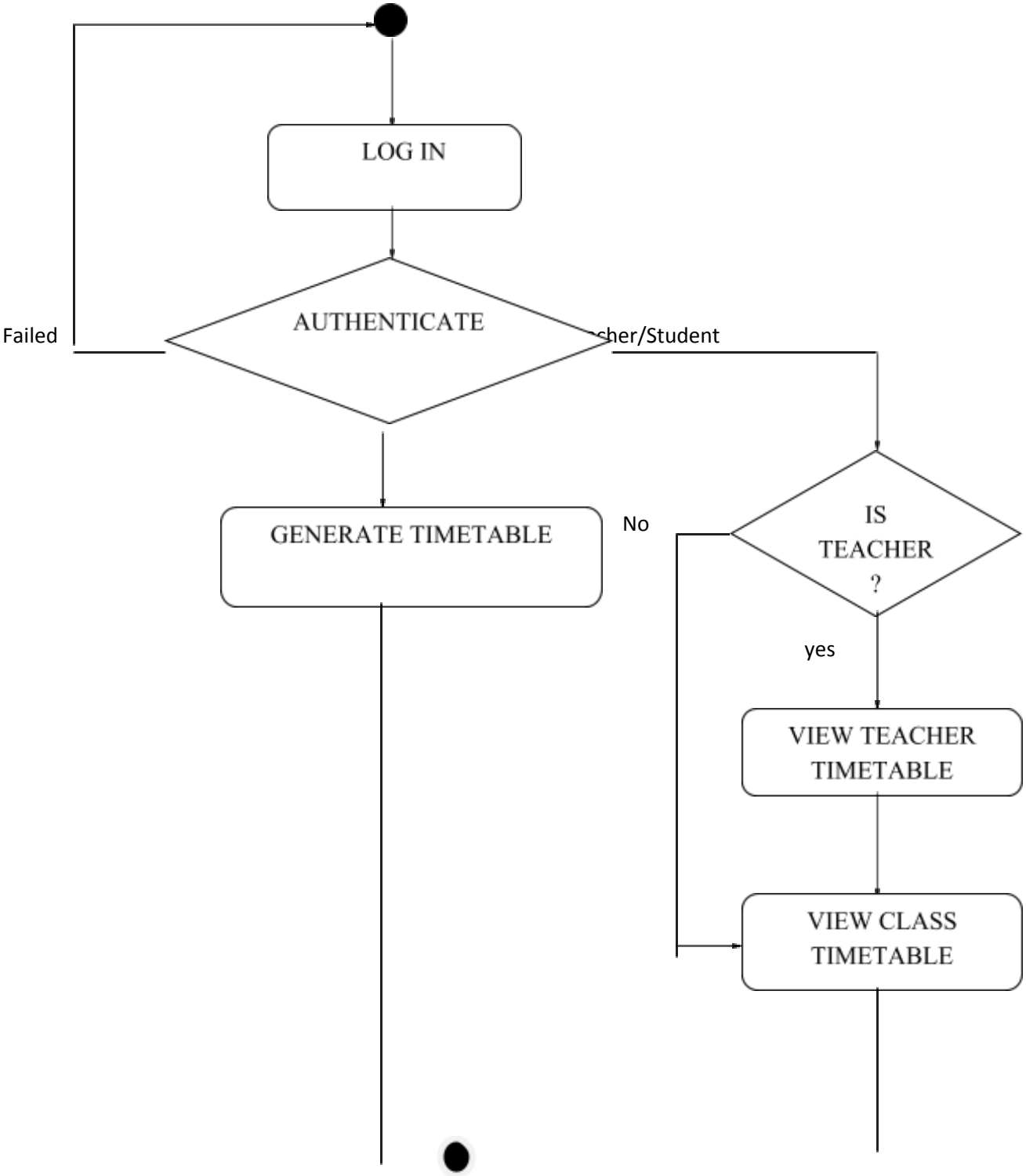

Fig:3.1.4Activity Diagram

CLASS DIAGRAM:

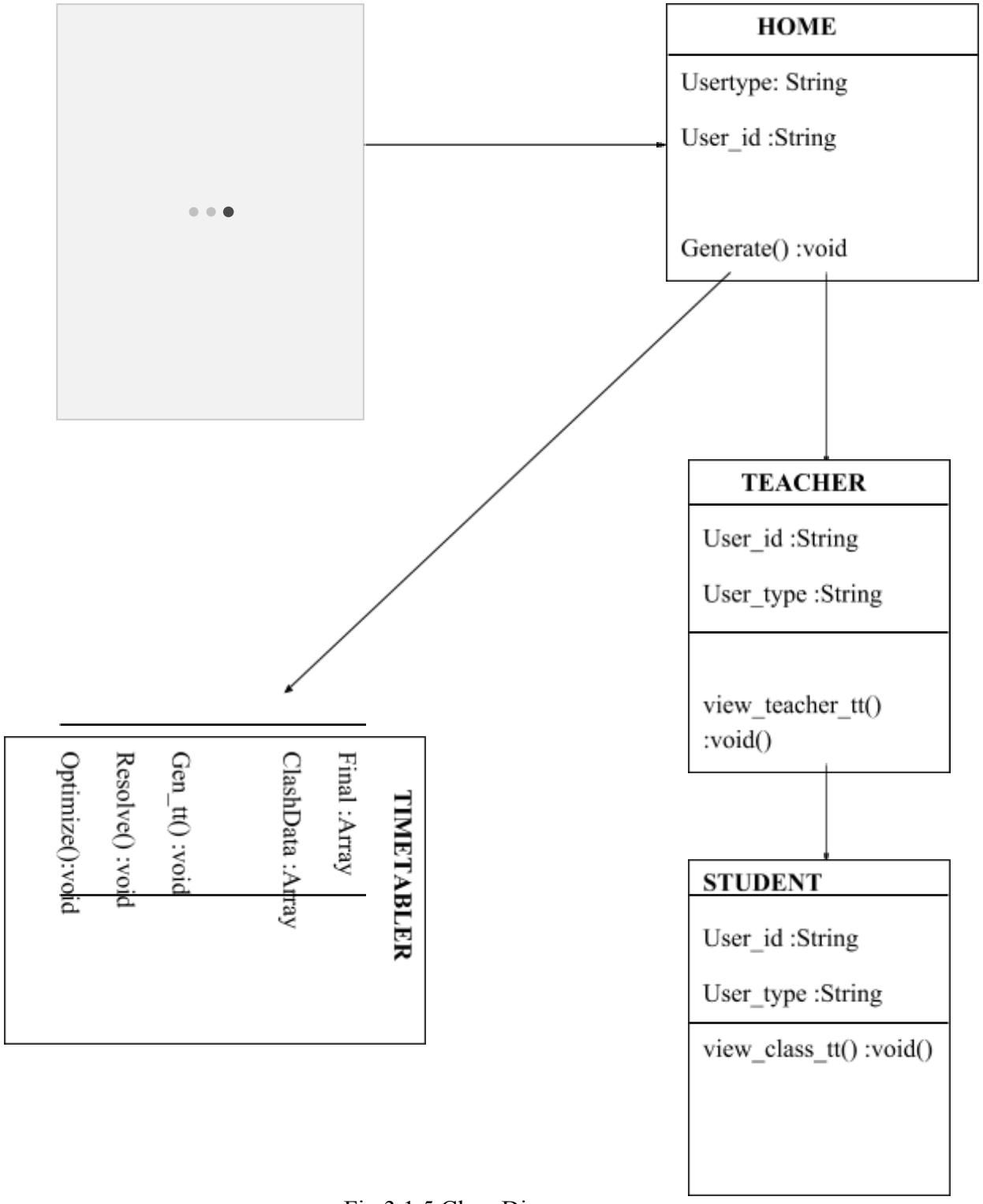

Fig 3.1.5 Class Diagram

SEQUENCE DIAGRAM:

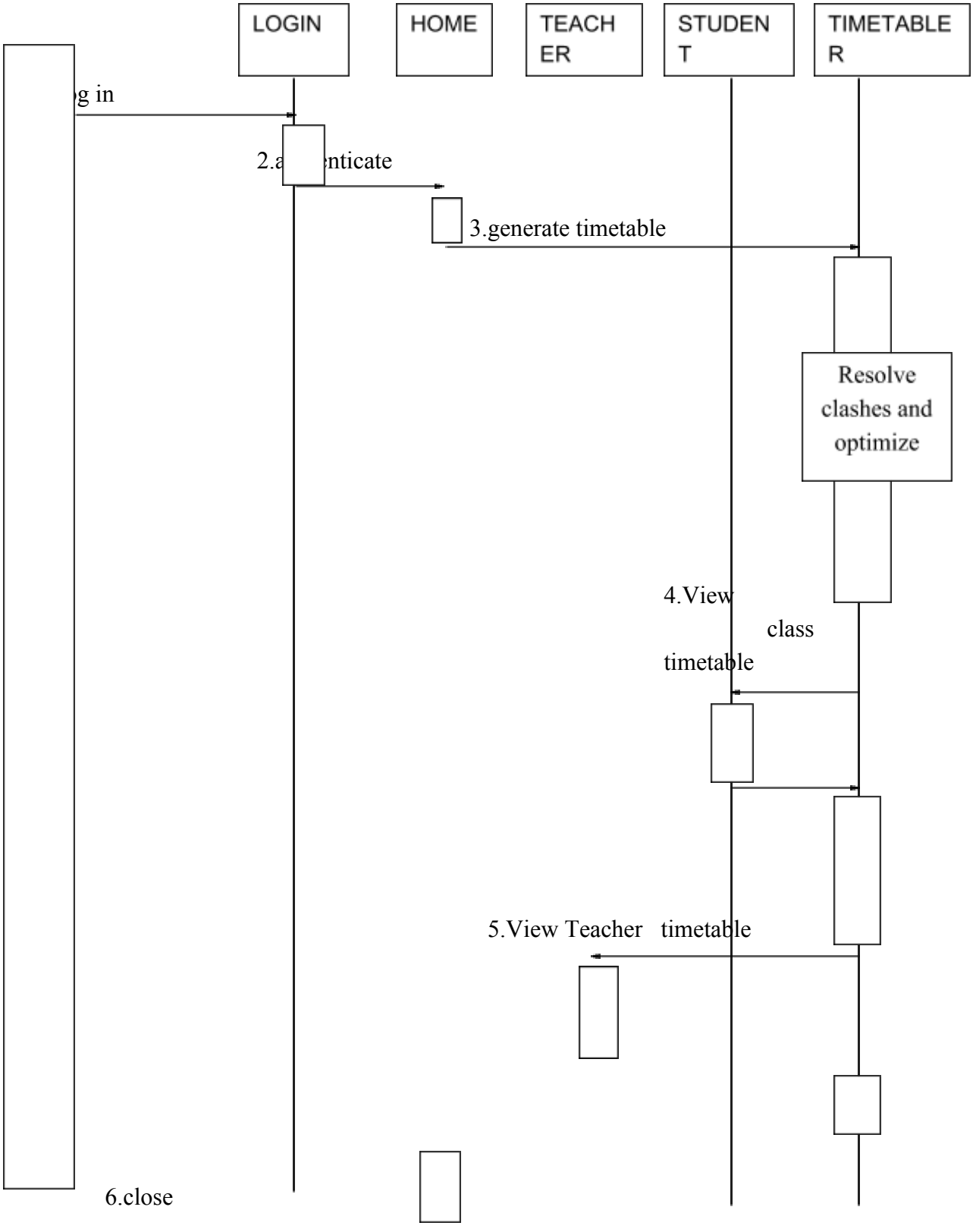

Fig 3.1.6 Sequence Diagram

COLLABORATION DIAGRAM

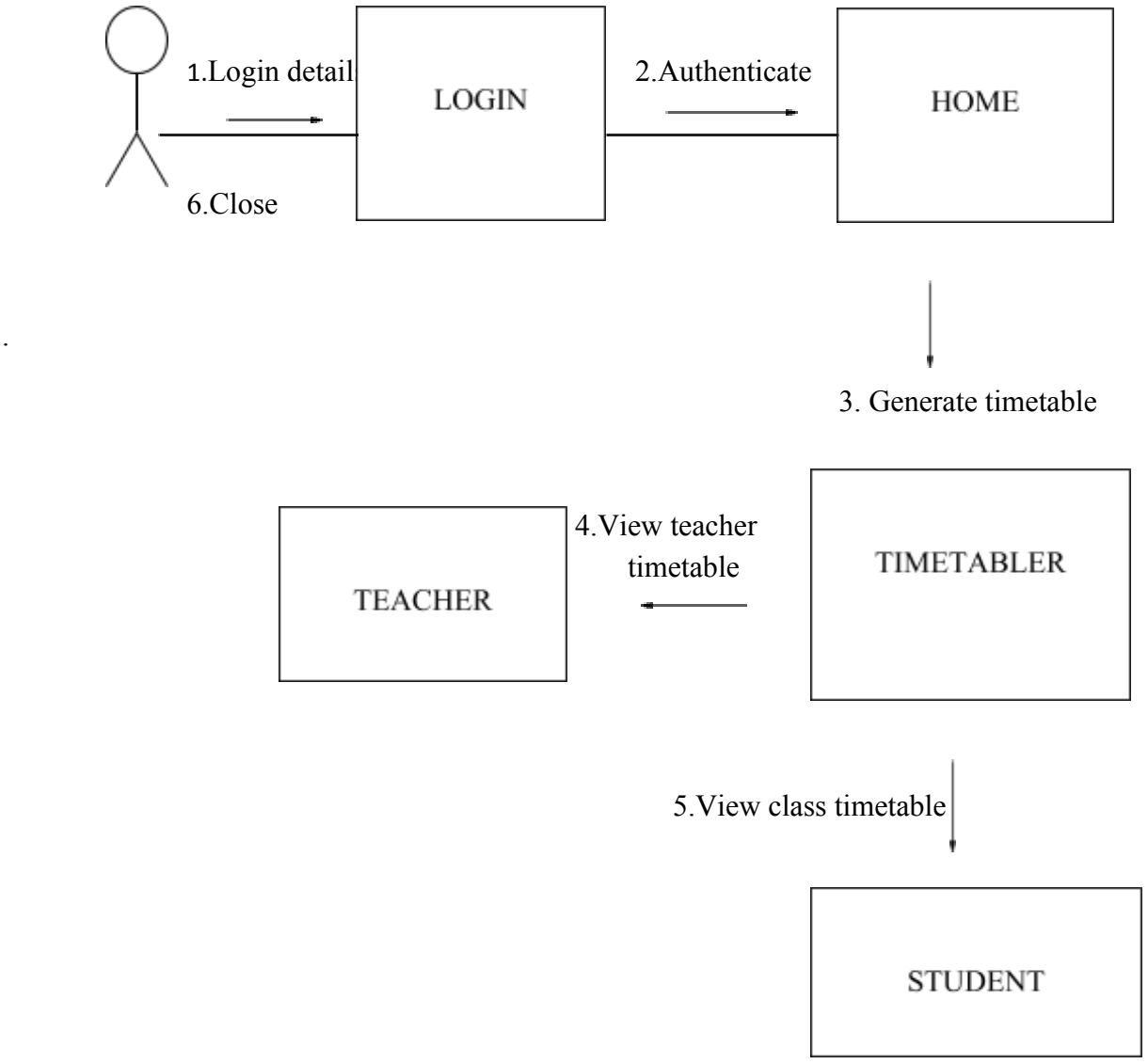

Fig3.1.7: Collaboration Diagram

3.2 DATA DESIGN

ENTITY RELATIONSHIP DIAGRAM:

Fig 3.2.1: Entity Relationship Diagram

3.3 EXTERNAL INTERFACE DESIGN

GUI and User interface:

Our system reads

- the inputs like number of teachers, number of classes through web page
- subject each teacher handle through an excel sheet with teacher name and subject
- subjects for each semester, department pair through second excel sheet
- Number of classes under each semester, department pair through web page
- Hours that teachers are engaged in any other work through web page
- Number of timeslots in a day and working days in a week through web page.
- Pre allotted lab time slots through an excel sheet
- Output for class wise and teacher wise timetable on web page
- Complete timetable in an excel sheet.

Hardware Interface:

- Database for user authentication

3.4 COMPONENT LEVEL DESIGN

COMPONENT DIAGRAM:

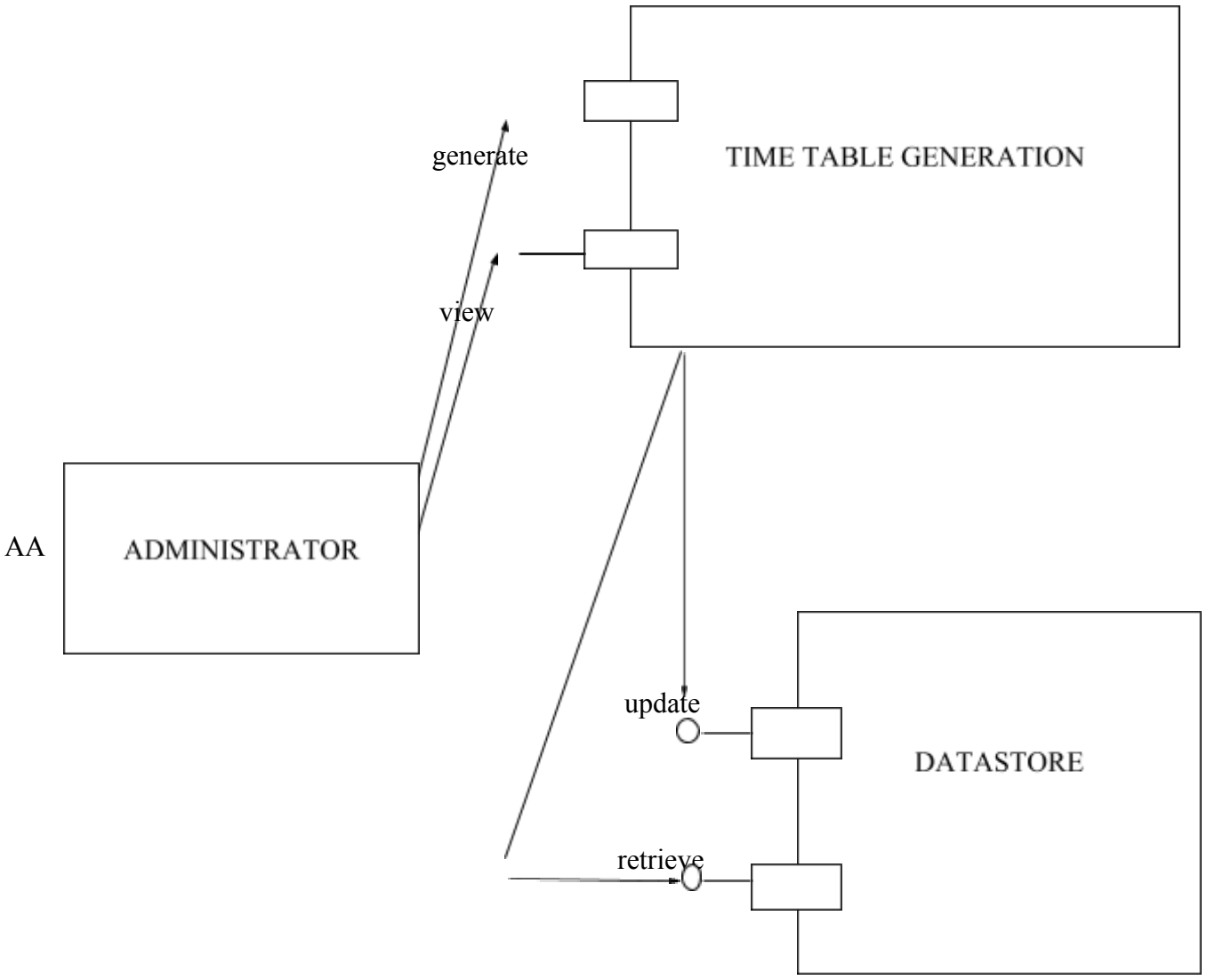

Fig: 3.4.1Component Diagram

3.4.1 HARDWARE AND SOFTWARE REQUIREMENTS

Software requirements:

Operating System	: Windows 7
Programming Language	: Java
Development Kit	: JDK 1.7
Technology	:JSP

Hardware requirements:

Processor	: Pentium P4 Processors or better
Memory	: 1 GB of RAM [Required]
Hard Disk	: 40 GB

4. PROJECT SCHEDULE AND MAJOR MILESTONES

MILESTONES AND TARGET DATES

s	**ng time**	**g time**	**ion**
rement analysis	o-2013	o-2013	ks
ng	o-2013	t-2013	ks
n	t-2013	v-2013	ks
nentation	v-2013	-2014	eks
g	r-2014	r-2014	ks

CURRENT STATUS OF MILESTONES

Phases	**Name**	**Status**
rement analysis	A	Done
ng	B	Done
n	C	Done
nentation	D	To be completed
g	E	To be completed

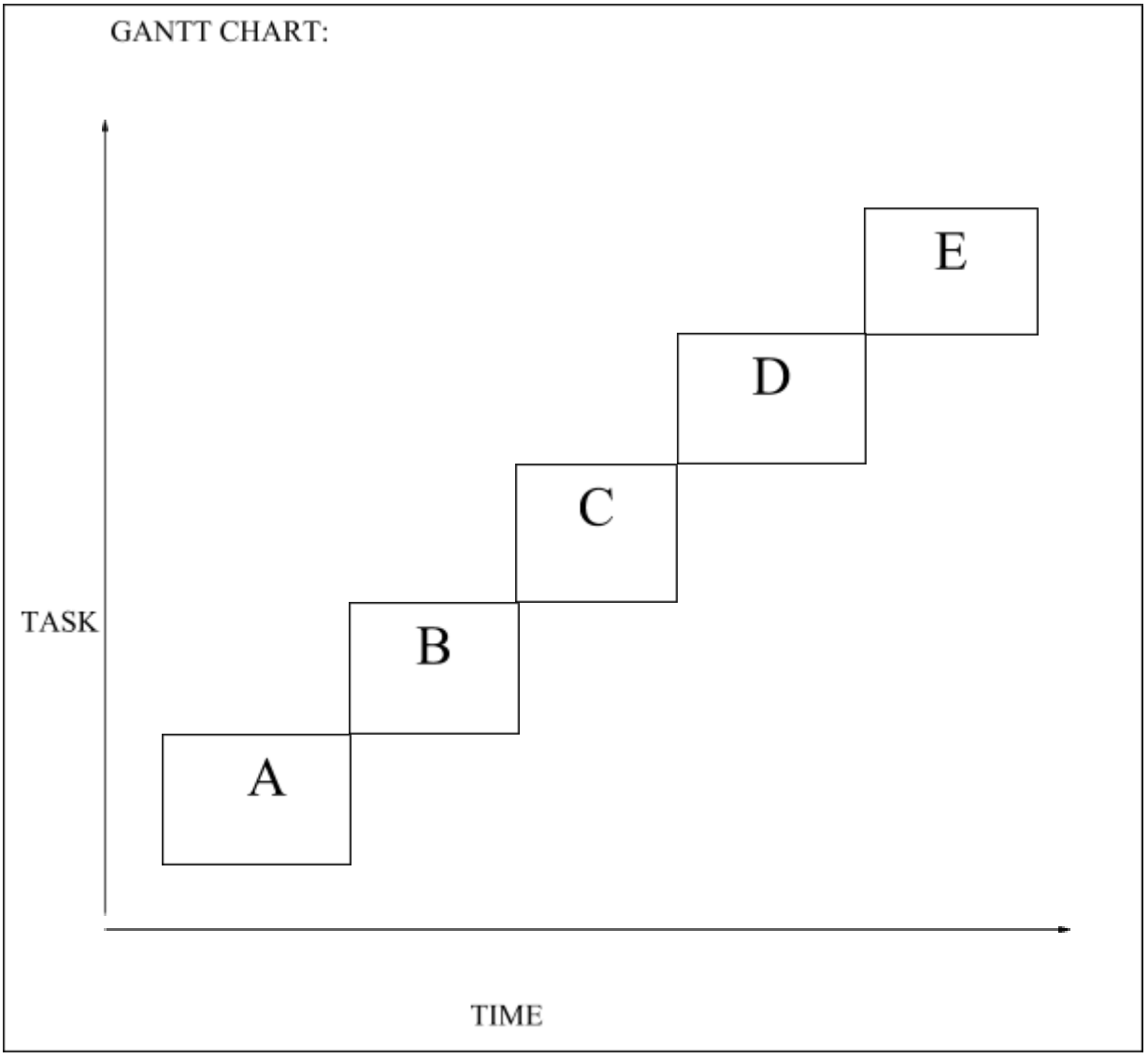

Fig 4.1:Gantt Chart

5. RISKS AND CHALLENGES

For any change in inputs say teachers list, subject list etc new time table should be generated, i.e. program should be executed again. Also for a time table generation problem no best solution is obtained, therefore we go for a set of good solutions and choose the better one from these solutions.

- Optimal solution cannot be generated in polynomial time.
- Better solution from a generated set is chosen
- Inputs must be fixed initially

6. GLOSSARY

Activity diagram

A diagram which shows flow of control through various stages

Class diagram

A diagram which shows classes and relation among them

ER diagram

A diagram which shows relations and entities

Incremental Model:

A method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished.

Object-Oriented Analysis and Design(OOAD):

A software development approach to design and implement software system as a collection of interacting stateful objects with specified structure and behavior

Sequence digram

A diagram which shows flow of control through various classes

Use case diagram

A diagram which shows relation between actors and use cases

7. BIBLIOGRAPHY

1. An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach, Anirydha Nanda, Manisha P. Pai and Abhijeet Gole, International Journal Of Machine Learning and Computing Vol.2,No.4, August 2012.
2. A New Hybrid GA-Bees Algorithm for a Real-world University Timetabling Problem, Nguyen Tan Tran Minh Khang, Tran Thi Hue Nuong 2011 International Conference on Intelligent Computation and Bio-Medical Instrumentation.
3. www.researchgate.net/genetic_algorithm
4. www.wikipedia.org/wiki/heuristic