AUTOMATIC TIME TABLE GENERATOR

Submitted in partial fulfillment of the requirements

for the award of the degree of

Bachelor of Computer Applications

To

Guru Gobind Singh Indraprastha University, Delhi

Guide:
Ms. Suman Singh
(Assistant Professor)

Submitted by:
Mostakim Mullick
(02213702016)

Institute of Information Technology & Management,

New Delhi – 110058

Batch (2016-2019)

Certificate

I, **Mostakim Mulllick** Roll No. **02213702016** certify that the Project Report/Dissertation (BCA-356) entitled **"Automatic Time Table Generator"** is done by me and it is an authentic work carried out by me at **Institute of Information Technology & Management.**

The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Signature of the Student
Date:

Certified that the Project Report/Dissertation (BCA-356) entitled **"Automatic Time Table Generator"** done by **Mostakim Mullck**, Roll No. **02213702016**, is completed under my guidance.

Signature of the Guide
Date:
Name of the Guide: Suman Singh
Designation: Assistant Professor
Address:
Institute of Information Technology & Management, Janakpuri
New Delhi

Countersigned
Program Coordinator

Countersigned
Director

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend our sincere thanks to all of them.

I am highly indebted to **Ms.Suman Singh** for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the project.

I would like to express our gratitude towards our parents & member of **Institute of Information Technology & Management at Janakpuri** , for their kind co-operation and encouragement which helped me in completion of this project. I would also like to express our special gratitude and thanks to industry persons for giving us such attention and time.

Signature of the Students
Date:

TABLE OF CONTENTS

S No	Topic	Page No
1	Certificate	-
2	Acknowledgements	-
3	Synopsis	-
4	List of Tables/Figures/Symbols	-
5	Chapter-1: Introduction	1-6
6	Chapter-2: System Requirements Analysis	7-9
7	Chapter-3: System Design	10-16
8	Chapter-4: System Development	17-41
9	Chapter-5: Summary and Conclusions	42-43
10	References/Bibliography	44
11	Appendices	

LIST OF TABLES

Table No	Title	Page No
Database Tables		
3.1	Admin Table	14
3.2	Classroom Table	14
3.3	First Year Table	14
3.4	Second Year Table	15
3.5	Third Year Table	15
3.6	Subject Table	16
3.7	Faculty Table	16
Testing Tables		
4.1	Functioning Testing	36
4.2	Interface Testing	39
4.3	Navigation Testing	39

LIST OF FIGURES

Figure No	Title	Page No
1.1	Waterfall Model	3
1.2	Gantt Chart	6
2.1	Block Diagram	8
2.2	Use-case Diagram	9
3.1	E-R Diagram	10
3.2	Class Diagram	11
3.3	Sequence Diagram	12
3.4	Activity Diagram	13

SYNOPSIS

1. Title of the Project: Automatic Time Table Generator.

2. Problem Definition:

- 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 timetable and modifying it but still it is a tedicios job to incoperate changes.
- To overcome all these problems, we propose to make an automated system. The system will take various inputs like details of subjects, class rooms and Faculty available, depending upon these inputs it will generate a possible time table, making optimal utilization of all resources in a way that will best suit any of constraints or college rules.

3. Objectives & Scope :.

The main objectives of our project are:

- The final system should able to generate time tables in completely automated way which will save a lot of time and effort of an institute administration.
- User defined constraints handling.
- Ease of use for user of system so that he/she can make automatic timetable.
- Focus on optimization of resources i.e teachers, labs and rooms etc.
- Provide a facility for everyone to view timetable.
- Generate multiple useful views from time table.

Scope:

Timetable Generation System generates timetable for each class and teacher, in keeping with the availability calendar of teachers, availability and capacity of physical resources (such as classrooms, laboratories and computer room) and rules applicable at different classes, semesters, teachers and subjects level. Best of all, this Timetable Generation System tremendously improves resource utilization and optimization.

4. Methodology:

1. Information Gathering

During information gathering we observed that Making a Daily Clash Free time table is a very Tedious Tasks. It is very difficult to create a clash free time table. For this, we read the Research paper on "Generating Timetable and Students schedule based on data mining techniques" by Safwan M. Shatnawi, Fawzi Albalooshi and Khaleel Rababa'h, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 2, Issue 4, July-August 2012, pp.1638-1644.

2. Tools and Techniques

The tools and techniques used are mentioned as:

- Tools used:
    1. Designing: XAMPP/WAMPP
    2. Coding: Notepad++
    3. Database: MySQL
    4. Creating UML Diagrams: ArgoUML
- Techniques used:
    1. Front-End: HTML, CSS
    2. Back-End: Java script, PHP
    3. Database: MySQL, PHPMyAdmin

Resources Needed

• Processor: Pentium(R) Dual-core CPU or Higher

• Hard Disk: Minimum 40 GB

• Ram: 512MB or Higher

CHAPTER-1

1.1Description of Organization

Institute of Information Technology and Management (IITM) was set up in 1999 under the aegis of Mata Leelawati Shikshan Sansthan (MLSS), a registered education society engaged in philanthropic activities, with the Late Shri T.N. Chaturvedi, the well known educationist, parliamentarian, ex-Governor of Karnataka and CAG of India and Padma Vibhushan Awardee, as founder President of both the society and the Institute.

The Institute takes pride in having developed the faculty support and infrastructure imperative to effectively implement the 'Outcome Based Education', a technology-based learner centric and result-oriented approach which enhances students' learning and performance capabilities. We are passionate about grooming the nation's youth to grow into excellent professionals and good human beings destined to become torch bearers of their respective domains.

IITM conducts a plethora of short duration skill enhancement and syllabus enrichment workshops related to areas of management specialisations and emerging IT technologies. We have a strong alumni network of over 4800 professionals working at various management levels in the leading corporate houses of the country.

1.2About the Proposed system

1.2.1 AIM

To overcome the previous problems of Time table scheduling by making a Web-Based Clash Free, Automatic Time Table Generator. The system will take various inputs like details of subjects, class rooms and Faculty available, depending upon these inputs it will generate a possible time table, making optimal utilization of all resources in a way that will best suit any of constraints or college rules.

1.2.2 System Objective

- The final system should able to generate time tables in completely automated way which will save a lot of time and effort of an institute administration.
- User defined constraints handling.
- Ease of use for user of system so that he/she can make automatic timetable.
- Focus on optimization of resources i.e teachers, labs and rooms etc.
- Provide a facility for everyone to view timetable.
- Generate multiple useful views from time table.

1.2.3 System Scope

Timetable Generation System generates timetable for each class and teacher, in keeping with the availability calendar of teachers, availability and capacity of physical resources (such as classrooms, laboratories and computer room) and rules applicable at different classes, semesters, teachers and subjects level. Best of all, this Timetable Generation System tremendously improves resource utilization and optimization.

1.2.4 Module Details

Different modules in Automatic Time Table Generator are:

1. **Login.**
    - **Admin Login:** Admin can login by using Username and Password.
    - **Faculty Login:** Faculty can login using his/her Tcode**.**
2. **Teacher Details-** Admin will provide the Teacher Details.
3. **Subject Details-** Admin will provide the Subject Details.
4. **Allocation Details-** Admin will allocate Faculty and subjects with respect to theories and practical lab accordingly.
5. **Room Details-** Admin will provide the room number so no clash may happen.
6. **Generate and View Timetable.** Admin can Generate and view the Time Table but Faculty and students can only download and view the time table**.**

1.3Methodology for Data Collection

1.3.1 Primary data collection.

Data has been collected for providing some details in the system. During information gathering we observed that Making a Daily Clash Free time table is a very Tedious Tasks. It is very difficult to create a clash free time table.

1.3.2 Secondary data collection

For this, we read the Research paper on "Generating Timetable and Students schedule based on data mining techniques" by Safwan M. Shatnawi, Fawzi Albalooshi and Khaleel Rababa'h, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 2, Issue 4, July-August 2012, pp.1638-1644.

1.4Methodology used for system design

Software Process Model

The Waterfall Model was first Process Model to be introduced. In a Waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. Waterfall model is the earliest SDLC approach that was used for software development.

In "The Waterfall" approach, the whole process of software development is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. This means that any phase in the development process begins only if the previous phase is complete.

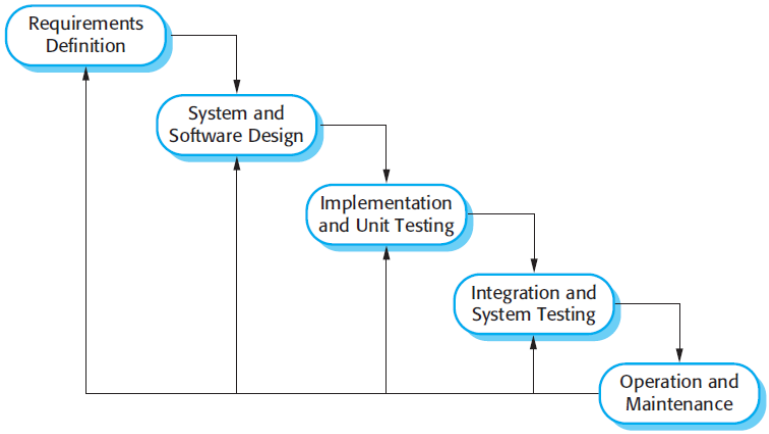


Fig. 1.1 Waterfall Model.

1.5 Methods for system Design (UML)

1.5.1 Use case diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system

The notation for a use case diagram is pretty straightforward and doesn't involve as many types of symbols as other UML diagrams. Here are all the shapes you will be able to find in Lucidchart:

- **Use cases:** Horizontally shaped ovals that represent the different uses that a user might have.

- **Actors:** Stick figures that represent the people actually employing the use cases.
- **Associations:** A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.
- **System boundary boxes:** A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system.

1.5.2 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

1.5.3 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

1.5.4 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

1.6System Tools Requirements

1.6.1 Hardware Specification

- Processor: Pentium(R) Dual-core CPU or Higher
- Hard Disk: Minimum 40 GB
- Ram: 512MB or Higher
- Monitor
- Keyboard, Mouse: optical.

1.6.2 Software Specification

The tools and techniques used are mentioned as:

- Tools used:
    1. Designing: XAMPP/WAMPP
    2. Coding: Notepad++
    3. Database: MySQL
    4. Creating UML Diagrams: ArgoUML
- Techniques used:
    1. Front-End: HTML, CSS
    2. Back-End: Java script, PHP
    3. Database: MySQL, PHPMyAdmin.

1.7Project Planning

1.7.1 Gantt Chart

Today, Gantt charts are most commonly used for tracking project schedules. For this it is useful to be able to show additional information about the various tasks or phases of the project.

Features of Gantt chart includes:

a. It creates a picture of complexity.

b. It organizes your thoughts

c. It demonstrates that you know what you're doing

d. It helps you to set realistic time frames

e. It can be highly visible.

Task Mode	Task Name	Duration	Start	Finish	% Work Complete
	Evaluate current system	2 days	Thu 1/17/19	Fri 1/18/19	100%
	Define Problem	2 days	Sat 1/19/19	Mon 1/21/19	100%
	Planning	3 days	Tue 1/22/19	Thu 1/24/19	75%
	Define Requirement	2 days	Fri 1/25/19	Sat 1/26/19	50%
	System Design	15 days	Sun 1/27/19	Thu 2/14/19	50%
	System Development	15 days	Fri 2/15/19	Thu 3/7/19	50%
	System Integration	4 days	Fri 3/8/19	Wed 3/13/19	25%
	System Testing	5 days	Thu 3/14/19	Wed 3/20/19	25%
	System Deployment	3 days	Thu 3/21/19	Sun 3/24/19	25%
	Prepare Documentation	47 days	Sun 1/20/19	Sun 3/24/19	75%

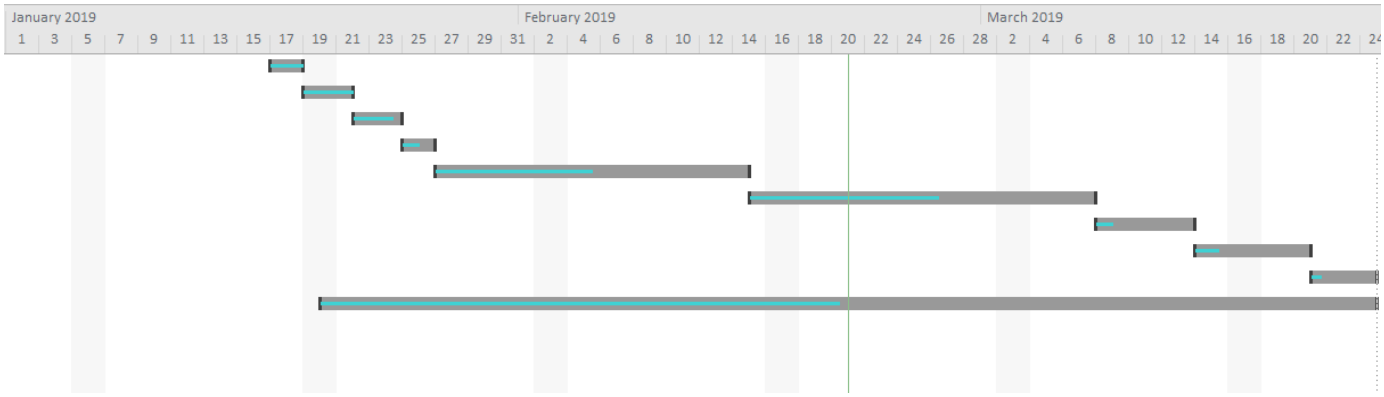


Fig. 1.2 Gantt Chart

CHAPTER-2

2.1 System Requirement Analysis

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirements analysis is critical to the success of a development project. Requirements must be actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

Conceptually, requirements analysis includes three types of activity:

- **Eliciting requirements:** the task of communicating with customers and users to determine what their requirements are. This is sometimes also called requirements gathering.
- **Analyzing requirements:** determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues.
- **Recording requirements:** Requirements might be documented in various forms, such as natural-language documents, use cases, user stories, or process specifications.

2.2 System Overview

A good project system overview provides enough information that the reader has a good sense of the capacity of the system, what it can do, what it can interact with as well as any special requirements.

We propose to make an **Automatic Time Table Generator** system. The system will take various inputs like details of subjects, class rooms and Faculty available, depending upon these inputs it will generate a possible time table, making optimal utilization of all resources in a way that will best suit any of constraints or college rules.

2.3 Block Diagram.

A **block diagram** is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks.[1] They are

heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams.

Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation. Contrast this with the schematic diagrams and layout diagrams used in electrical engineering, which show the implementation details of electrical components and physical construction.

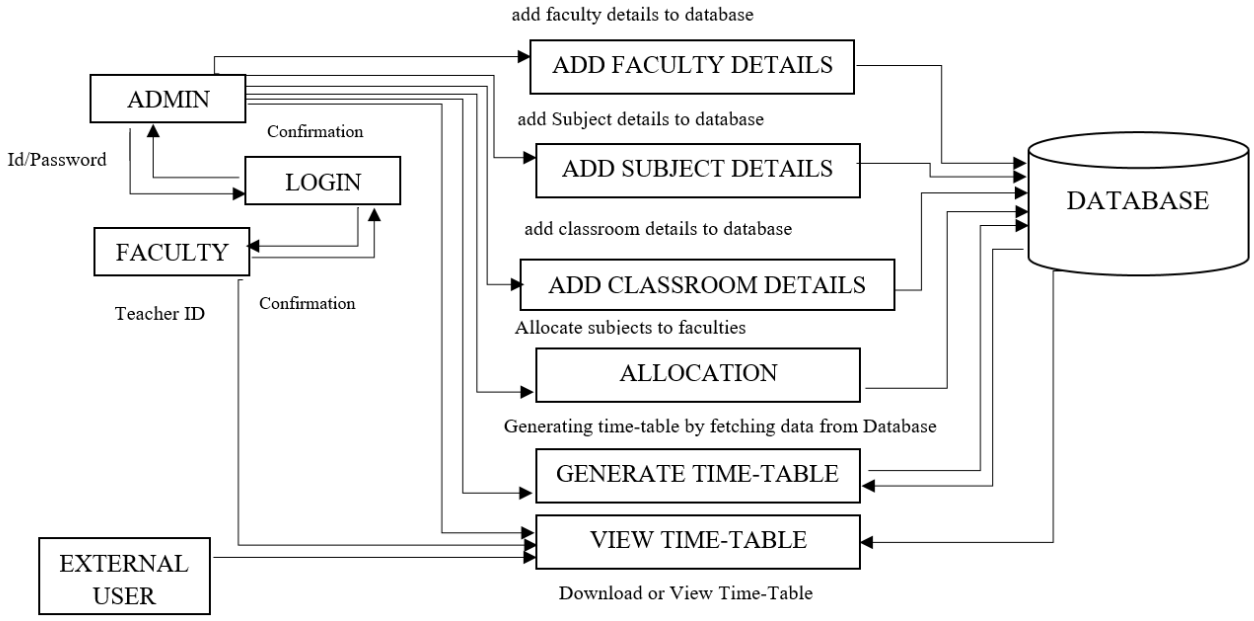


Fig 2.1 Block Diagram

2.4 System Process input and output Specification.

2.4.1. Process of your system

- **Login.**
  - **Admin Login:** Admin can login by using Username and Password.
  - **Faculty Login:** Faculty can login using his/her Teacher ID**.**
- **Teacher Details-** Admin will provide the Teacher Details.
- **Subject Details-** Admin will provide the Subject Details.
- **Allocation Details-** Admin will allocate Faculty and subjects with respect to theories and practical lab accordingly.
- **Room Details-** Admin will provide the room number so no clash may happen.
- **Generate and View Timetable.** Admin can Generate and view the Time Table but Faculty and students can only download and view the time table**.**

2.5 Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system

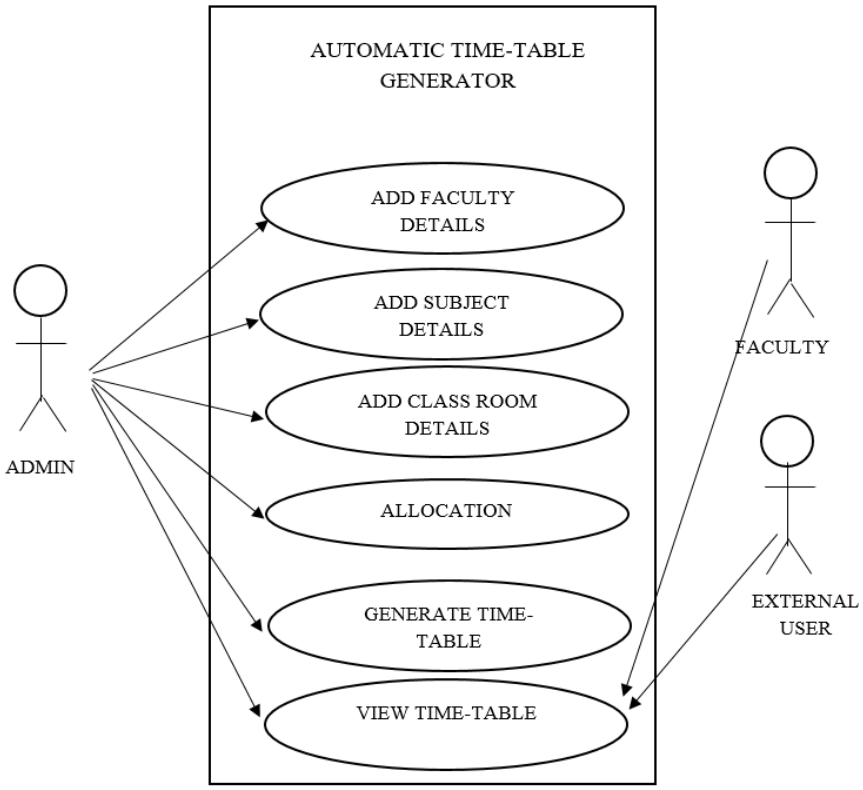


Fig 2.2 Use-Case Diagram

Common components include:

- **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.
- **System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.
- **Goals:** The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

CHAPTER-3

3.1 ER diagram

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (*entities*) that are connected by lines (*relationships*) which express the associations and dependencies between entities. An ER model can also be expressed in a verbal form, for example: *one building may be divided into zero or more apartments, but one apartment can only be located in one building.*

Entities may be characterized not only by relationships, but also by additional properties (*attributes*), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity–relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering.

Conceptual data model

This is the highest level ER model in that it contains the least granular detail but establishes the overall scope of what is to be included within the model set. The conceptual ER model normally defines master reference data entities that are commonly used by the organization. Developing an enterprise-wide conceptual ER model is useful to support documenting the data architecture for an organization.

A conceptual ER model may be used as the foundation for one or more *logical data models* (see below). The purpose of the conceptual ER model is then to establish structural metadata commonality for the master data entities between the set of logical ER models. The conceptual data model may be used to form commonality relationships between ER models as a basis for data model integration.

Logical data model

A logical ER model does not require a conceptual ER model, especially if the scope of the logical ER model includes only the development of a distinct information system. The logical ER model contains more detail than the conceptual ER model. In addition to master data entities, operational and transactional data entities are now defined. The details of each data entity are developed and the relationships between these data entities are established. The logical ER model is however developed independently of the specific database management system into which it can be implemented.

Physical data model

One or more physical ER models may be developed from each logical ER model. The physical ER model is normally developed to be instantiated as a database. Therefore, each physical ER model must contain enough detail to produce a database and each physical ER model is technology dependent since each database management system is somewhat different.

The physical model is normally instantiated in the structural metadata of a database management system as relational database objects such as database tables, database

indexes such as unique key indexes, and database constraints such as a foreign key constraint or a commonality constraint. The ER model is also normally used to design modifications to the relational database objects and to maintain the structural metadata of the database.

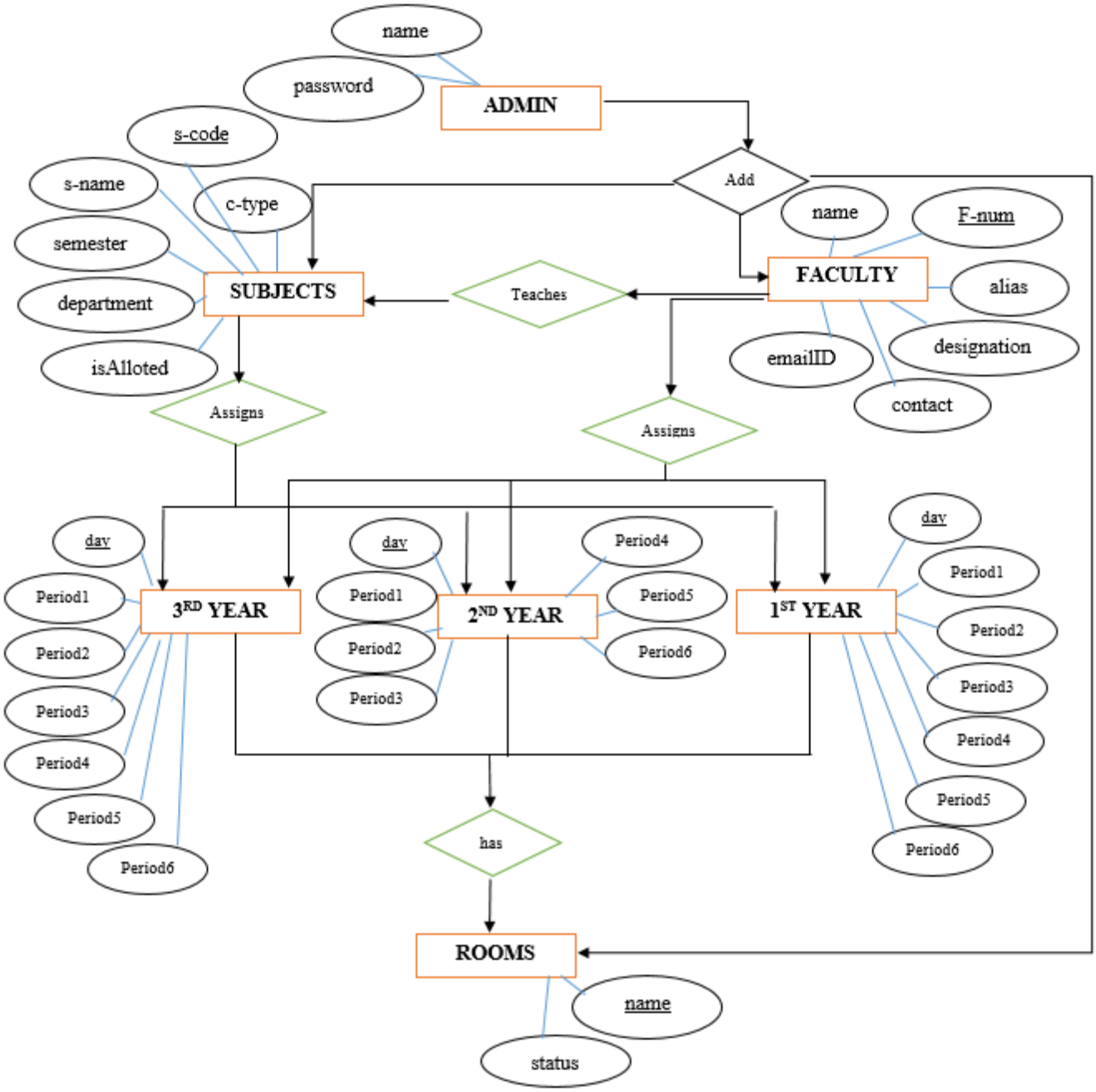


Fig 3.1 Entity- Relationship Diagram

3.2 Class Diagram

In software engineering, a **class diagram** in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

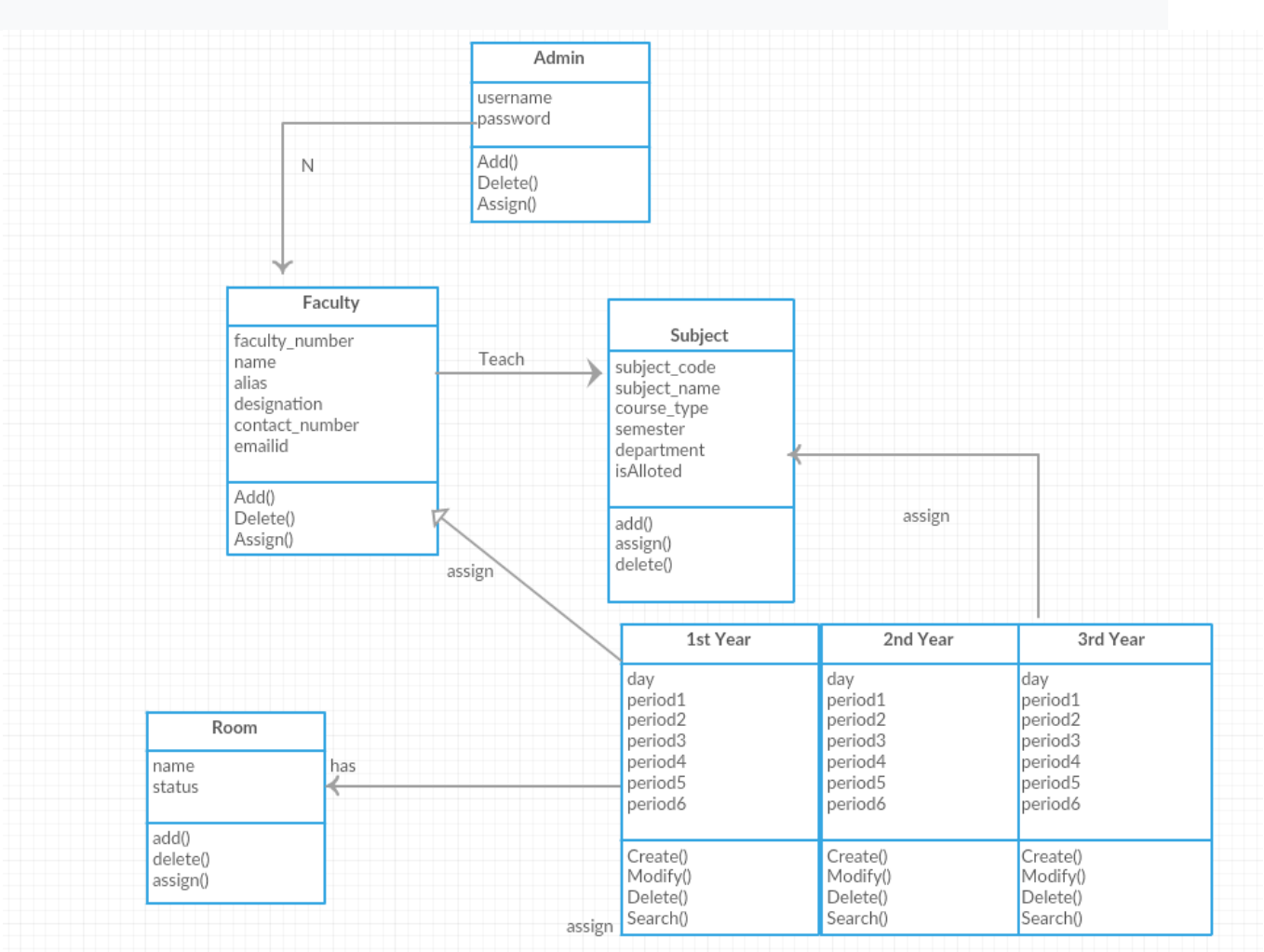


Fig 3.2 Class Diagram

3.3 Sequence Diagram.

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the

order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

1. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.
   We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.
   For example – Here the user in seat reservation system is shown as an actor where it exists outside the system and is not a part of the system.
2. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram. The standard in UML for naming a lifeline follows the following format – Instance Name : Class Name
   We display a lifeline in a rectangle called head with its name and type. The head is located on top of a vertical dashed line (referred to as the stem) as shown above. If we want to model an unnamed instance, we follow the same pattern except now the portion of lifeline's name is left blank.
   Difference between a lifeline and an actor – A lifeline always portrays an object internal to the system whereas actors are used to depict objects external to the system. The following is an example of a sequence diagram:
3. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.
   Messages can be broadly classified into the following **categories** :

   - **Synchronous messages –** A synchronous message waits for a reply before the interaction can move forward. The sender waits until the receiver has completed the processing of the message. The caller continues only when it knows that the receiver has processed the previous message i.e. it receives a reply message. A large number of calls in object oriented programming are synchronous. We use a solid arrow head to represent a synchronous message.

   - **Asynchronous Messages –** An asynchronous message does not wait for a reply from the receiver. The interaction moves forward irrespective of the receiver processing the previous message or not. We use a lined arrow head to represent an asynchronous message.

   - **Create message –** We use a Create message to instantiate a new object in the sequence diagram. There are situations when a particular message call requires the creation of an object. It is represented with a dotted arrow and create word labelled on it to specify that it is the create Message symbol.
     For example – The creation of a new order on a e-commerce website would require a new object of Order class to be created.

- **Delete Message –** We use a Delete Message to delete an object. When an object is deallocated memory or is destroyed within the system we use the Delete Message symbol. It destroys the occurrence of the object in the system.It is represented by an arrow terminating with a x.
  For example – In the scenario below when the order is received by the user, the object of order class can be destroyed.

- **Self Message –** Certain scenarios might arise where the object needs to send a message to itself. Such messages are called Self Messages and are represented with a U shaped arrow.
  For example – Consider a scenario where the device wants to access its webcam. Such a scenario is represented using a self message.
  Reply Message – Reply messages are used to show the message being sent from the receiver to the sender. We represent a return/reply message using an open arrowhead with a dotted line. The interaction moves forward only when a reply message is sent by the receiver.

  For example – Consider the scenario where the device requests a photo from the user. Here the message which shows the photo being sent is a reply message.

- **Found Message –** A Found message is used to represent a scenario where an unknown source sends the message. It is represented using an arrow directed towards a lifeline from an end point. For example: Consider the scenario of a hardware failure. It can be due to multiple reasons and we are not certain as to what caused the hardware failure.
  Lost Message – A Lost message is used to represent a scenario where the recipient is not known to the system. It is represented using an arrow directed towards an end point from a lifeline. For example: Consider a scenario where a warning is generated.

  The warning might be generated for the user or other software/object that the lifeline is interracting with. Since the destination is not known before hand, we use the Lost Message symbol.

4. **Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.
   For example: In order to be able to withdraw cash, having a balance greater than zero is a condition that must be met as shown below.

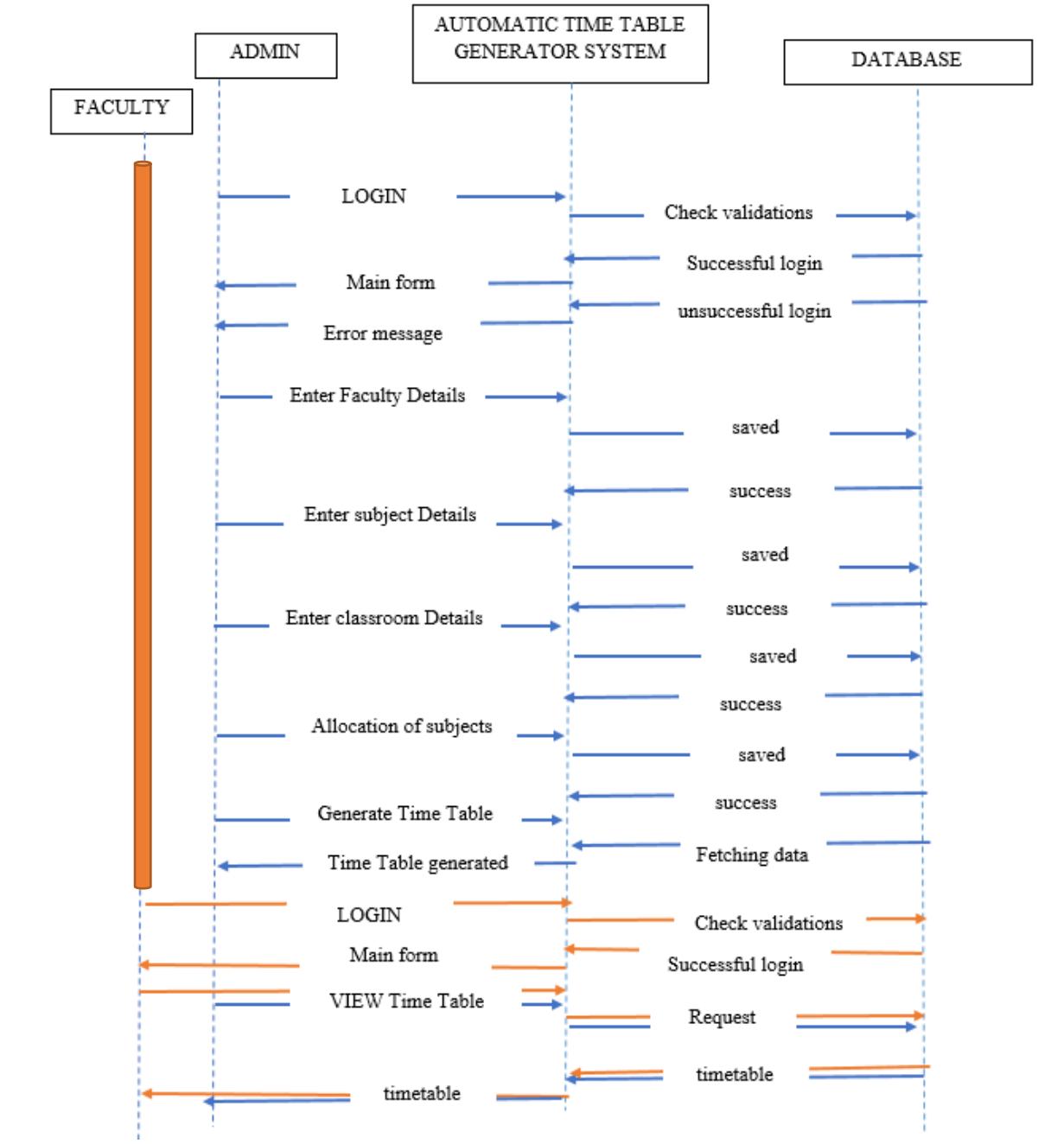


Fig 3.3 Sequence Diagram

3.4 Activity Diagram.

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system.

An activity diagram is very **similar to a flowchart**. So let us understand if an activity diagrams or a flowcharts are any different :

Difference between an Activity diagram and a Flowchart –

Flowcharts were typically invented earlier than activity diagrams. Non programmers use Flow charts to model workflows. For example: A manufacturer uses a flow chart to explain and illustrate how a particular product is manufactured. We can call a flowchart a primitive version of an activity diagram. Business processes where decision making is involved is expressed using a flow chart.

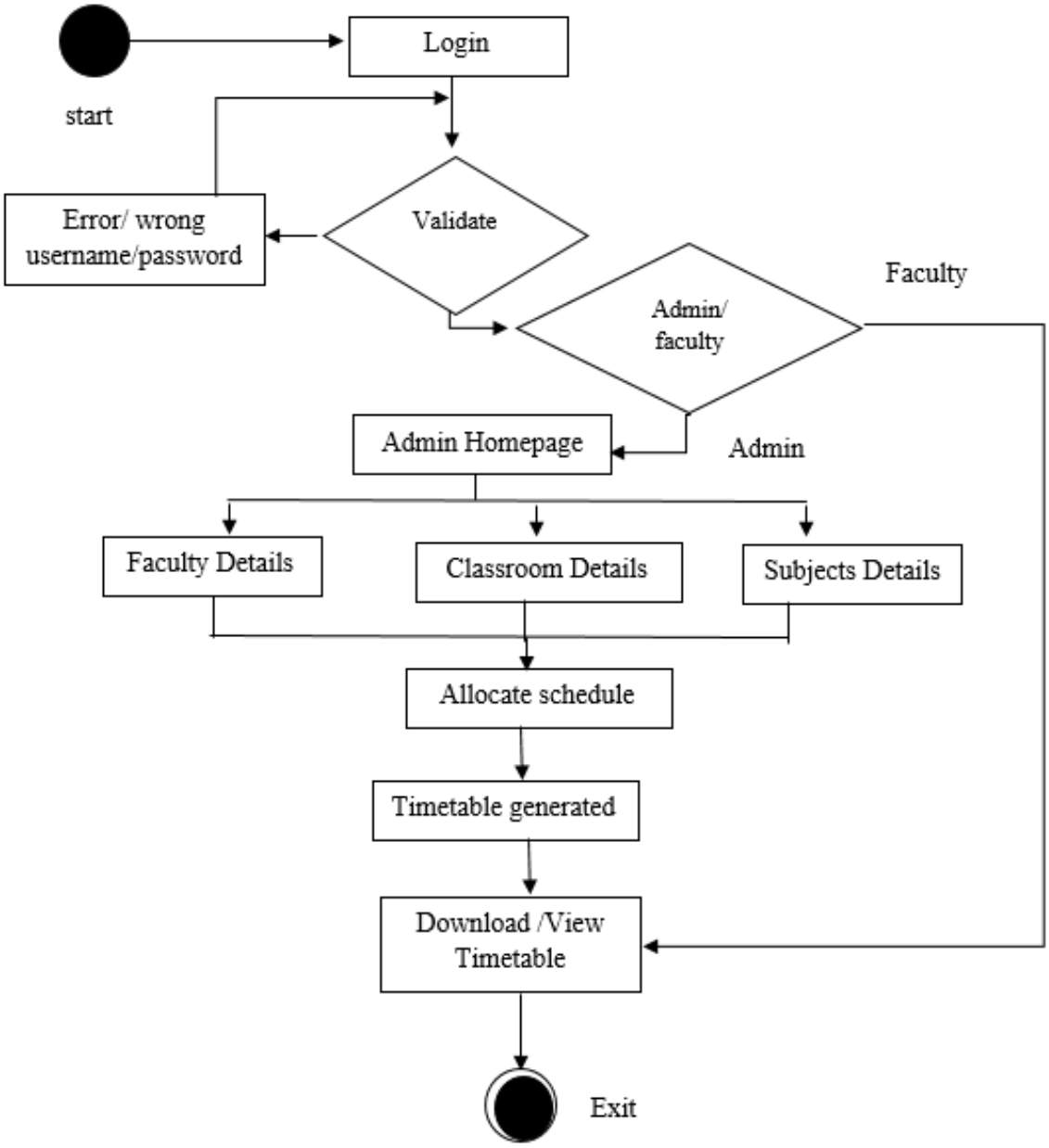


Fig 3.3 Activity Diagram

3.5 Data Base Table Structure.

ADMIN TABLE.

Field name	Field code	Field Type	Size(no. of characters)	Constraint	Description
Username	name	Alphanumeric	30	NOT NULL	Username of admin
Password	password	Alphanumeric	10	NOT NULL	Password to login

Tab 3.1 Admin Table

CLASSROOM TABLE.					
Field name	**Field code**	**Field Type**	**Size(no. of characters)**	**Constraint**	**Description**
Classroom number	name	Alphanumeric	30	PRIMARY KEY	Classroom number
Status	status	Numeric	1	NOT NULL	Status of the Classroom.

Tab 3.2 Classroom Table

FIRST YEAR TABLE.					
Field name	**Field code**	**Field Type**	**Size(no. of characters)**	**Constraint**	**Description**
DAY	day	Alphanumeric	10	PRIMARY KEY	Week Day
1st Period	period1	Alphanumeric	30	NOT NULL	First period of Class.
2nd Period	period2	Alphanumeric	30	NOT NULL	Second period of Class.
3rd Period	period3	Alphanumeric	30	NOT NULL	Third period of Class.
4th Period	period4	Alphanumeric	30	NOT NULL	Fourth period of Class.
5th Period	period5	Alphanumeric	30	NOT NULL	Fifth period of Class.
6th Period	period6	Alphanumeric	30	NOT NULL	Sixth period of Class.

Tab 3.3 First Year Table

SECOND YEAR TABLE.					
Field name	**Field code**	**Field Type**	**Size(no. of characters)**	**Constraint**	**Description**
DAY	day	Alphanumeric	10	PRIMARY KEY	Week Day
1st Period	period1	Alphanumeric	30	NOT NULL	First period of Class.
2nd Period	period2	Alphanumeric	30	NOT NULL	Second period of Class.
3rd Period	period3	Alphanumeric	30	NOT NULL	Third period of Class.
4th Period	period4	Alphanumeric	30	NOT NULL	Fourth period of Class.
5th Period	period5	Alphanumeric	30	NOT NULL	Fifth period of Class.
6th Period	period6	Alphanumeric	30	NOT NULL	Sixth period of Class.

Tab 3.4 Second Year Table

THIRD YEAR TABLE.					
Field name	**Field code**	**Field Type**	**Size(no. of characters)**	**Constraint**	**Description**
DAY	day	Alphanumeric	10	PRIMARY KEY	Week Day
1st Period	period1	Alphanumeric	30	NOT NULL	First period of Class.
2nd Period	period2	Alphanumeric	30	NOT NULL	Second period of Class.
3rd Period	period3	Alphanumeric	30	NOT NULL	Third period of Class.
4th Period	period4	Alphanumeric	30	NOT NULL	Fourth period of Class.

5th Period	period5	Alphanumeric	30	NOT NULL	Fifth period of Class.
6th Period	period6	Alphanumeric	30	NOT NULL	Sixth period of Class.

Tab 3.5 Third Year Table

SUBJECT TABLE.					
Field name	**Field code**	**Field Type**	**Size(no. of characters)**	**Constraint**	**Description**
Subject Code	subject_code	Alphanumeric	10	PRIMARY KEY	Code of subject
Subject Name	subject_name	Alphanumeric	30	NOT NULL	Name of subject.
Course Type	course_type	Alphanumeric	15	NOT NULL	Name of course
Semester	semester	Numeric	1	NOT NULL	Semester number.
Department	department	Alphanumeric	50	NOT NULL	Type of department.
Is Alloted	isAlloted	Numeric	1	NOT NULL	Subject is allocated to whom.

Tab 3.6 Subject Table

FACULTY TABLE.					
Field name	**Field code**	**Field Type**	**Size(no. of characters)**	**Constraint**	**Description**
Faculty Code	faculty_number	Alphanumeric	10	PRIMARY KEY	Code of Fcaulty.
Faculty Name	name	Alphabetic	30	NOT NULL	Name of Faculty.
Alias	alias	Alphanumeric	10	NOT NULL	Short name.
Designation	designation	Alphanumeric	30	NOT NULL	Designation of faculty
Contact no.	contact_number	Numeric	10	NOT NULL	Contact number
Email ID	emailid	Alphanumeric	50	NOT NULL	Email ID.

Tab 3.7 Faculty Table

CHAPTER- 4

4.1 Program Development

4.1.1 Home page Code and Output

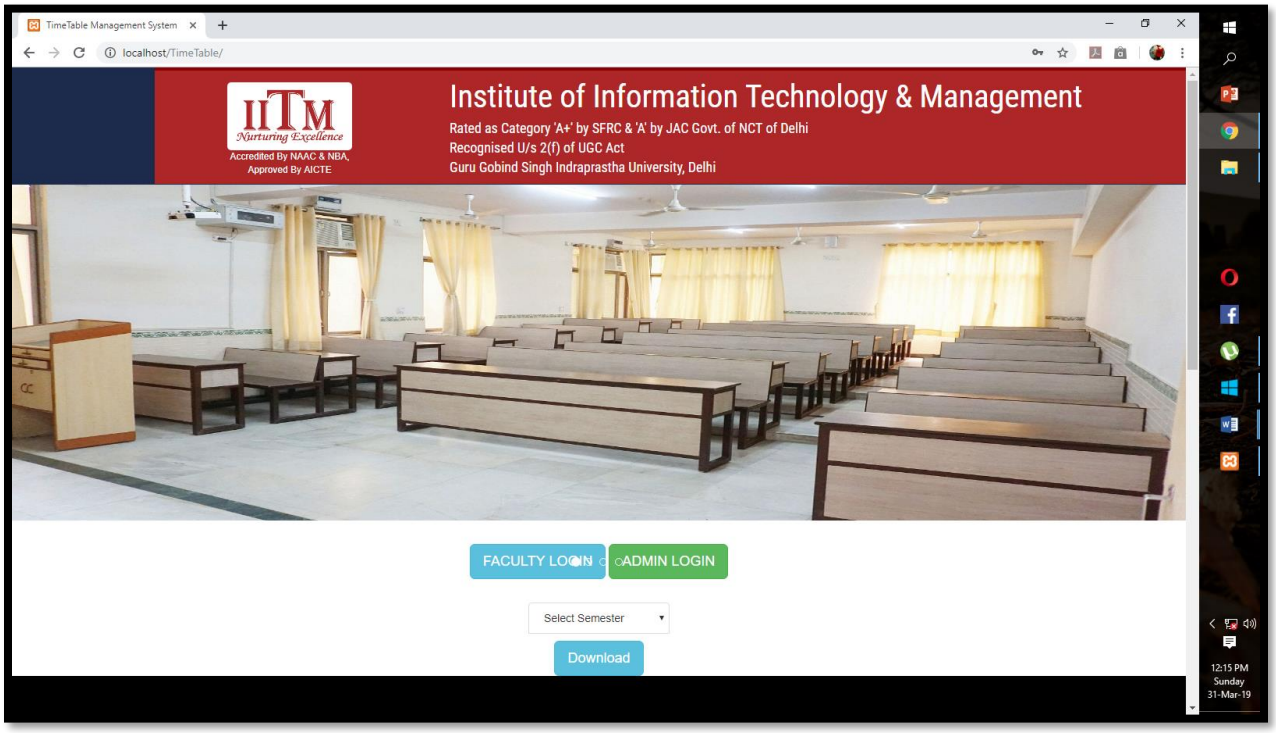
```
<?php
if (isset($_GET['generated']) && $_GET['generated'] == "false") {
   unset($_GET['generated']);
   echo '<script>alert("Timetable not generated yet!!");</script>';
}
?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <meta charset="utf-8"/>
   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1"/>
   <meta name="description" content=""/>
   <meta name="author" content=""/>
   <title>TimeTable Management System</title>
   <!-- BOOTSTRAP CORE STYLE CSS -->
   <link href="assets/css/bootstrap.css" rel="stylesheet"/>
   <!-- FONT AWESOME CSS -->
   <link href="assets/css/font-awesome.min.css" rel="stylesheet"/>
   <!-- FLEXSLIDER CSS -->
   <link href="assets/css/flexslider.css" rel="stylesheet"/>
   <!-- CUSTOM STYLE CSS -->
   <link href="assets/css/style.css" rel="stylesheet"/>
   <!-- Google     Fonts -->
   <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,700,300' rel='stylesheet'
type='text/css'/>

</head>
<body>
<div class="navbar navbar-inverse navbar-fixed-top " id="menu">
   <div class="container">
      <div align="right">
         <img src="assets/img/header.png" alt="Flower">
      </div>
   </div>
</div>

<div id="myCarousel" class="carousel slide" data-ride="carousel">
   <!-- Indicators -->
   <ol class="carousel-indicators" style="margin-bottom: 160px">
      <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
      <li data-target="#myCarousel" data-slide-to="1"></li>
      <li data-target="#myCarousel" data-slide-to="2"></li>
      <li data-target="#myCarousel" data-slide-to="3"></li>
   </ol>

   <!-- Wrapper for slides -->
   <div class="carousel-inner" role="listbox">
      <div class="item active">
         <img src="assets/img/lab1.jpg" alt="Chania">
      </div>

      <div class="item">
         <img src="assets/img/lab2.jpg" alt="Chania">
      </div>

      <div class="item">
         <img src="assets/img/building.jpg" alt="Flower">
      </div>
```

```
                <div class="item">
                    <img src="assets/img/lab1.jpg" alt="Flower">
                </div>
            </div>
        </div>
        <script type="text/javascript">
            function genpdf() {
                var doc = new jsPDF();

                doc.addHTML(document.getElementById('TT'), function () {
                    doc.save('demo timetable.pdf');
                });
                window.alert("Downloaded!");
            }
        </script>
        <div align="center" STYLE="margin-top: 30px">
            <button data-scroll-reveal="enter from the bottom after 0.2s"
                    id="teacherLoginBtn" class="btn btn-info btn-lg">FACULTY LOGIN
            </button>
            <button data-scroll-reveal="enter from the bottom after 0.2s"
                    id="adminLoginBtn" class="btn btn-success btn-lg">ADMIN LOGIN
            </button>
        </div>
        <br>
        <div align="center">
            <form data-scroll-reveal="enter from the bottom after 0.2s" action="studentvalidation.php"
method="post">
                <select id="select_semester" name="select_semester" class="list-group-item">
                    <option selected disabled>Select Semester</option>
                    <option value="3"> BCA  ( Semester III )</option>
                    <option value="4"> BCA  ( Semester IV )</option>
                    <option value="5"> BCA  ( Semester V )</option>
                    <option value="6"> BCA  ( Semester VI )</option>
                    <option value="7"> BCA  ( Semester I )</option>
                    <option value="8"> BCA  ( Semester II )</option>
                </select>
                <button type="submit" class="btn btn-info btn-lg" style="margin-top: 10px">Download</button>
            </form>
        </div>
        <!-- The Modal -->
        <div id="myModal" class="modal">

            <!-- Modal content -->
            <div class="modal-content">
                <div class="modal-header">
                    <span class="close">&times</span>
                    <h2 id="popupHead">Modal Header</h2>
                </div>
                <div class="modal-body" id="LoginType">
                    <!--Admin Login Form-->
                    <div style="display:none" id="adminForm">
                        <form action="adminFormvalidation.php" method="POST">
                            <div class="form-group">
                                <label for="adminname">Username</label>
                                <input type="text" class="form-control" id="adminname" name="UN"
placeholder="Username ...">
                            </div>
                            <div class="form-group">
                                <label for="password">Password</label>
                                <input type="password" class="form-control" id="password" name="PASS"
                                       placeholder="Password ...">
```

```
                    </div>
                    <div align="right">
                        <input type="submit" class="btn btn-default" name="LOGIN" value="LOGIN">
                    </div>
                </form>
            </div>
        </div>
        <!--Faculty Login Form-->
        <div style="display:none" id="facultyForm">
            <form action="facultyformvalidation.php" method="POST" style="overflow: hidden">
                <div class="form-group">
                    <label for="facultyno">Faculty No.</label>
                    <input type="text" class="form-control" id="facultyno" name="FN" placeholder="Faculty No.
..."> 
                </div>
                <div align="right">
                    <button type="submit" class="btn btn-default" name="LOGIN">LOGIN</button>
                </div>
            </form>
        </div>
    </div>
</div>


<script>
    // Get the modal
    var modal = document.getElementById('myModal');

    // Get the button that opens the modal
    var teacherLoginBtn = document.getElementById("teacherLoginBtn");
    var adminLoginBtn = document.getElementById("adminLoginBtn");
    var heading = document.getElementById("popupHead");
    var facultyForm = document.getElementById("facultyForm");
    var adminForm = document.getElementById("adminForm");
    // Get the <span> element that closes the modal
    var span = document.getElementsByClassName("close")[0];

    // When the user clicks the button, open the modal
    adminLoginBtn.onclick = function () {
        modal.style.display = "block";
        heading.innerHTML = "Admin Login";
        adminForm.style.display = "block";
        facultyForm.style.display = "none";

    }
    teacherLoginBtn.onclick = function () {
        modal.style.display = "block";
        heading.innerHTML = "Faculty Login";
        facultyForm.style.display = "block";
        adminForm.style.display = "none";


    }

    // When the user clicks on <span> (x), close the modal
    span.onclick = function () {
        modal.style.display = "none";
        adminForm.style.display = "none";
        facultyForm.style.display = "none";

    }
```

```
            // When the user clicks anywhere outside of the modal, close it
            window.onclick = function (event) {
                if (event.target == modal) {
                    modal.style.display = "none";
                }
            }
        </script>
        <!--HOME SECTION END-->
        <!--HOME SECTION TAG LINE END-->

        <div id="faculty-sec">
            <div class="container set-pad">
                <div class="row text-center">
                    <div class="col-lg-8 col-lg-offset-2 col-md-8 col-sm-8 col-md-offset-2 col-sm-offset-2">
                        <h1 data-scroll-reveal="enter from the bottom after 0.1s" class="header-line">OUR FACULTY
</h1>

                    </div>

                </div>
                <!--/.HEADER LINE END-->

                <div class="row">


                    <div class="col-lg-4  col-md-4 col-sm-4" data-scroll-reveal="enter from the bottom after 0.4s">
                        <div class="faculty-div">
                            <img src="assets/img/faculty/dr_jyoti_batra.jpg" class="img-rounded"/>
                            <h3 align="center">Prof. (Dr.) Prerna Mahajan</br>
        </h3>
                            <hr/>
                            <h4 align="center"> Head of Department (Computer Science)<br/>MCA, M.Tech, M.Phil,
UGC-NET, Ph.D.</h4>

                        </div>
                    </div>
                    <div class="col-lg-4  col-md-4 col-sm-4" data-scroll-reveal="enter from the bottom after 0.5s">
                        <div class="faculty-div">
                            <img src="assets/img/faculty/dr_prerna_mahajan.jpg" class="img-rounded"/>
                            <h3 align="center">Dr. Ramandeep Kaur (Assosiate Professor)</h3>
                            <hr/>
                            <h4 align="center">MCA, PGDBA, Ph.D.<br/> Computer Science</h4>

                        </div>
                    </div>
                    <div class="col-lg-4  col-md-4 col-sm-4" data-scroll-reveal="enter from the bottom after 0.6s">
                        <div class="faculty-div">
                            <img src="assets/img/faculty/neha_jain.jpg" class="img-rounded"/>
                            <h3 align="center">Prof. (Dr.) Sudhir Kumar Sharma</br>Professor</h3>
                            <hr/>
                            <h4 align="center">M.Tech(CSE), Ph.D.(IT)<br/> Department of Computer Science</h4>

                        </div>
                    </div>

                </div>
            </div>
        </div>

        <div class="container">
            <div class="row set-row-pad">
                <div class="col-lg-4 col-md-4 col-sm-4   col-lg-offset-1 col-md-offset-1 col-sm-offset-1 "
```

```
                data-scroll-reveal="enter from the bottom after 0.4s">

                <h2><strong>Our Location </strong></h2>
                <hr/>
                <div>
                  <h4><b>Institute of Information Technology & Management</b></br>
        D-29, Institutional Area, Janakpuri, New Delhi-110058</br>
         <b>Contact</b></br>
        +91/011-28525882, 28520239, 28525051</br>
         <b>Email</b></br>
        director@iitmipu.ac.in
                  </h4>

                </div>


              </div>
              <div class="col-lg-4 col-md-4 col-sm-4   col-lg-offset-1 col-md-offset-1 col-sm-offset-1"
                data-scroll-reveal="enter from the bottom after 0.4s">

                <h2><strong>Social Conectivity </strong></h2>
                <hr/>
                <div>
                  <a href="https://www.facebook.com/iiTmJanakpuri/"> <img
src="assets/img/Social/facebook.png" alt=""/> </a>
                  <a href="http://www.iitmjanakpuri.com/"> <img src="assets/img/Social/google-plus.png"
alt=""/></a>
                  <a href="https://twitter.com/iitm_ipu"> <img src="assets/img/Social/twitter.png" alt=""/></a>
                </div>
              </div>


            </div>
          </div>
          <!-- CONTACT SECTION END-->
          <div id="footer">
            <!--  &copy 2014 yourdomain.com | All Rights Reserved |  <a href="http://binarytheme.com"
style="color: #fff" target="_blank">Design by : binarytheme.com</a>
          --></div>
          <!-- FOOTER SECTION END-->

          <!--  Jquery Core Script -->
          <script src="assets/js/jquery-1.10.2.js"></script>
          <!--  Core Bootstrap Script -->
          <script src="assets/js/bootstrap.js"></script>
          <!--  Flexslider Scripts -->
          <script src="assets/js/jquery.flexslider.js"></script>
          <!--  Scrolling Reveal Script -->
          <script src="assets/js/scrollReveal.js"></script>
          <!--  Scroll Scripts -->
          <script src="assets/js/jquery.easing.min.js"></script>
          <!--  Custom Scripts -->
          <script src="assets/js/custom.js"></script>
          </div>
          </body>
          </html>
```



4.1.2 Add Teacher Code and Output

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <meta charset="utf-8"/>
   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1"/>
   <meta name="description" content=""/>
   <meta name="author" content=""/>
   <title>TimeTable Management System</title>
   <!-- BOOTSTRAP CORE STYLE CSS -->
   <link href="assets/css/bootstrap.css" rel="stylesheet"/>
   <!-- FONT AWESOME CSS -->
   <link href="assets/css/font-awesome.min.css" rel="stylesheet"/>
   <!-- FLEXSLIDER CSS -->
   <link href="assets/css/flexslider.css" rel="stylesheet"/>
   <!-- CUSTOM STYLE CSS -->
   <link href="assets/css/style.css" rel="stylesheet"/>
   <!-- Google     Fonts -->
   <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,700,300' rel='stylesheet'
type='text/css'/>
</head>
<body>

<div class="navbar navbar-inverse navbar-fixed-top " id="menu">
   <div class="container">
      <div class="navbar-header">
         <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-
collapse">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
         </button>

      </div>
      <div class="navbar-collapse collapse move-me">
         <ul class="nav navbar-nav navbar-left">
            <li><a href="addteachers.php">ADD TEACHERS</a></li>
```

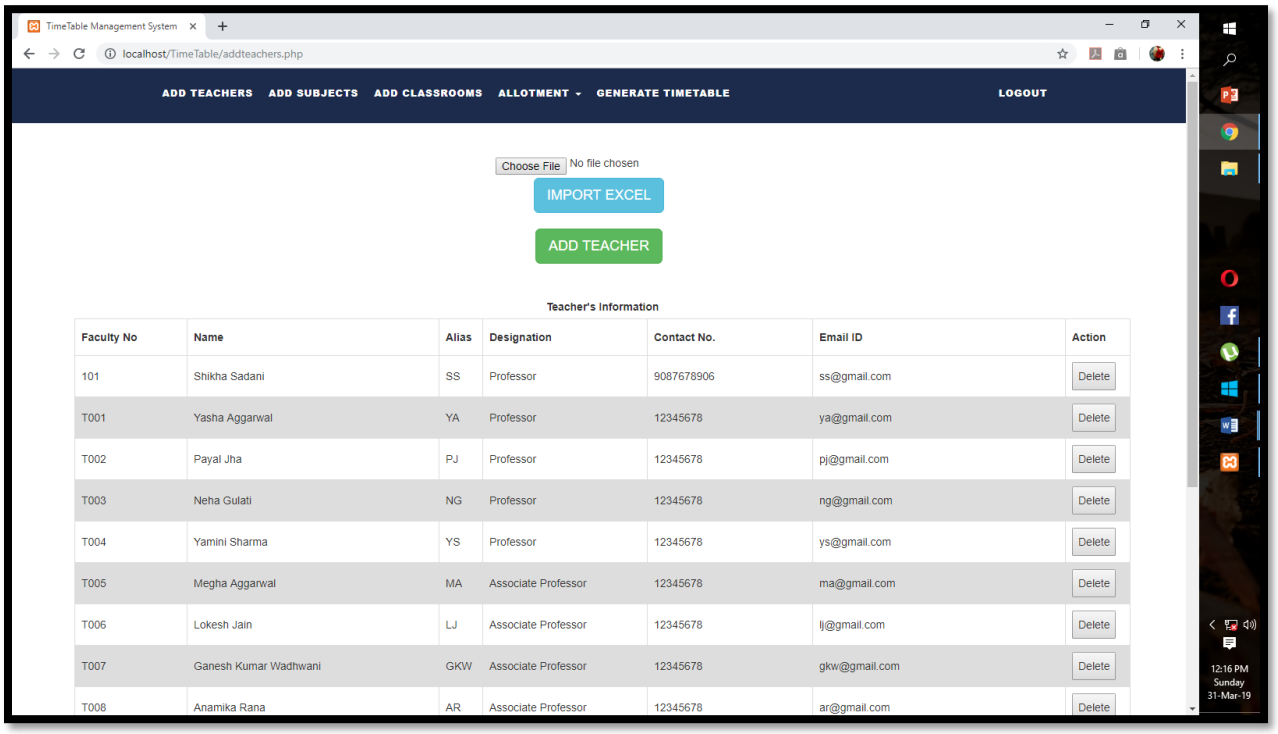
```
                        <li><a href="addsubjects.php">ADD SUBJECTS</a></li>
                        <li><a href="addclassrooms.php">ADD CLASSROOMS</a></li>
                        <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" aria-
expanded="false">ALLOTMENT
                                <span class="caret"></span></a>
                            <ul class="dropdown-menu">
                                <li>
                                    <a href=allotsubjects.php>THEORY COURSES</a>
                                </li>
                                <li>
                                    <a href=allotpracticals.php>PRACTICAL COURSES</a>
                                </li>
                                <li>
                                    <a href=allotclasses.php>CLASSROOMS</a>
                                </li>
                            </ul>
                        </li>
                        <li><a href="generatetimetable.php">GENERATE TIMETABLE</a></li>

                    </ul>
                    <ul class="nav navbar-nav navbar-right">
                        <li><a href="index.php">LOGOUT</a></li>
                    </ul>

                </div>
            </div>
        </div>
        <!--NAVBAR SECTION END-->
        <br>

        <div align="center" style="margin-top:80px">
            <form name="import" method="post" enctype="multipart/form-data">
                <input type="file" name="file"/>
                <input type="submit" name="teacherexcel" id="teacherexcel" class="btn btn-info btn-lg"
value="IMPORT EXCEL"/>
            </form>
            <?php
            if (isset($_POST['teacherexcel'])) {
                if (empty($_FILES['file']['tmp_name'])) {
                    echo '<script>alert("Select a file first! ");</script>';
                } else {
                    $file = $_FILES['file']['tmp_name'];
                    $handle = fopen($file, 'r');
                    $headings = true;
                    while (!feof($handle)) {
                        $filesop = fgetcsv($handle, 1000);

                        $facno = $filesop[0];
                        $name = $filesop[1];
                        $alias = $filesop[2];
                        $designation = $filesop[3];
                        $contact = $filesop[4];
                        $email = $filesop[5];
                        if ($facno == "" || $facno == "Faculty No.") {
                            continue;
                        }
                        $q = mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
                            "INSERT INTO teachers VALUES
('$facno','$name','$alias','$designation','$contact','$email')");
                        if ($q) {
                            $sql = "CREATE TABLE " . $facno . " (
                        day VARCHAR(10) PRIMARY KEY,
```

```
                    period1 VARCHAR(30),
                    period2 VARCHAR(30),
                    period3 VARCHAR(30),
                    period4 VARCHAR(30),
                    period5 VARCHAR(30),
                    period6 VARCHAR(30)
                    )";
                        mysqli_query(mysqli_connect("localhost", "root", "", "ttms"), $sql);
                        $days = array('monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday');
                        for ($i = 0; $i < 6; $i++) {
                            $day = $days[$i];
                            $sql = "INSERT into " . $facno . " VALUES('$day','','','','','')";
                            mysqli_query(mysqli_connect("localhost", "root", "", "ttms"), $sql);
                        }
                    }
                }
            }
        }
        ?>
    </div>
    <div align="center" style="margin-top:20px">
        <button id="teachermanual" class="btn btn-success btn-lg">ADD TEACHER</button>
    </div>

    <div id="myModal" class="modal">

        <!-- Modal content -->
        <div class="modal-content" style="margin-top: -60px">
            <div class="modal-header">
                <span class="close">&times</span>
                <h2 id="popupHead">Add Teacher</h2>
            </div>
            <div class="modal-body" id="EnterTeacher">
                <!--Admin Login Form-->
                <div style="display:none" id="addTeacherForm">
                    <form action="addteacherFormValidation.php" method="POST">
                        <div class="form-group">
                            <label for="teachername">Teacher's Name</label>
                            <input type="text" class="form-control" id="teachername" name="TN"
                                placeholder="Teacher's Name ...">
                        </div>
                        <div class="form-group">
                            <label for="TF">Faculty No</label>
                            <input type="text" class="form-control" id="facultyno" name="TF" placeholder="Faculty
No ...">
                        </div>
                        <div class="form-group">
                            <label for="TF">Alias</label>
                            <input type="text" class="form-control" id="alias_name" name="AL"
placeholder="Alias..">
                        </div>
                        <div class="form-group">
                            <label for="designation">Designation</label>

                            <select class="form-control" id="designation" name="TD">
                                <option selected disabled>Select</option>
                                <option value="Professor">Professor</option>
                                <option value="Assistant Professor">Assistant Professor</option>
                                <option value="Associate Professor">Associate Professor</option>
                                <option value="Guest Faculty">Guest Faculty</option>
                            </select>
                        </div>
```

```
                <div class="form-group">
                    <label for="teachercontactnumber">Contact No.</label>
                    <input type="text" class="form-control" id="teachercontactnumber" name="TP"
                        placeholder="+91 ...">
                </div>

                <div class="form-group">
                    <label for="teacheremailid">Email-ID</label>
                    <input type="text" class="form-control" id="teacheremailid" name="TE"
                        placeholder="abc@xyz.com ...">
                </div>
                <div align="right">
                    <input type="submit" class="btn btn-default" name="ADD" value="ADD">
                </div>
            </form>
        </div>
    </div>
    <div class="modal-footer">
    </div>
  </div>
</div>

<script>
  // Get the modal
  var modal = document.getElementById('myModal');

  // Get the button that opens the modal
  var addteacherBtn = document.getElementById("teachermanual");
  var heading = document.getElementById("popupHead");
  var facultyForm = document.getElementById("addTeacherForm");
  // Get the <span> element that closes the modal
  var span = document.getElementsByClassName("close")[0];

  // When the user clicks the button, open the modal

  addteacherBtn.onclick = function () {
    modal.style.display = "block";
    //heading.innerHTML = "Faculty Login";
    facultyForm.style.display = "block";
    //adminForm.style.display = "none";


  }

  // When the user clicks on <span> (x), close the modal
  span.onclick = function () {
    modal.style.display = "none";
    //adminForm.style.display = "none";
    facultyForm.style.display = "none";

  }

  // When the user clicks anywhere outside of the modal, close it
  window.onclick = function (event) {
    if (event.target == modal) {
      modal.style.display = "none";
    }
  }
</script>


<div>
```

```
<br>
<style>
   table {
      margin-top: 10px;
      font-family: arial, sans-serif;
      border-collapse: collapse;
      margin-left: 30px;
      width: 90%;
   }

   td, th {
      border: 1px solid #dddddd;
      text-align: left;
      padding: 8px;
   }

   tr:nth-child(even) {
      background-color: #dddddd;
   }
</style>

<script>
   function deleteHandlers() {
      var table = document.getElementById("teacherstable");
      var rows = table.getElementsByTagName("tr");
      for (i = 0; i < rows.length; i++) {
         var currentRow = table.rows[i];
         //var b = currentRow.getElementsByTagName("td")[0];
         var createDeleteHandler =
            function (row) {
               return function () {
                  var cell = row.getElementsByTagName("td")[0];
                  var id = cell.innerHTML;
                  var x;
                  if (confirm("Are You Sure?") == true) {
                     window.location.href = "deleteteacher.php?name=" + id;

                  }

               };
            };
         currentRow.cells[6].onclick = createDeleteHandler(currentRow);
      }
   }
</script>

<table id=teacherstable style="margin-left: 80px">
   <caption><strong>Teacher's Information </strong></caption>
   <tr>
      <th width="130">Faculty No</th>
      <th width=290>Name</th>
      <th width=50>Alias</th>
      <th width="190">Designation</th>
      <th width="190">Contact No.</th>
      <th width="290">Email ID</th>
      <th width="40">Action</th>
   </tr>
   <tbody>
   <?php
   include 'connection.php';
   $q = mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
      "SELECT * FROM teachers ORDER BY faculty_number ASC");
```

```
                while ($row = mysqli_fetch_assoc($q)) {
                   echo "<tr><td>{$row['faculty_number']}</td>
                         <td>{$row['name']}</td>
                         <td>{$row['alias']}</td>
                         <td>{$row['designation']}</td>
                         <td>{$row['contact_number']}</td>
                         <td>{$row['emailid']}</td>
                        <td>
                         <button>Delete</button></td>
                         </tr>\n";
                }
              <!--  &copy 2014 yourdomain.com | All Rights Reserved |  <a href="http://binarytheme.com"
style="color: #fff" target="_blank">Design by : binarytheme.com</a>
            -->
            <!-- FOOTER SECTION END-->

            <!--  Jquery Core Script -->
            <script src="assets/js/jquery-1.10.2.js"></script>
            <!--  Core Bootstrap Script -->
            <script src="assets/js/bootstrap.js"></script>
            <!--  Flexslider Scripts -->
            <script src="assets/js/jquery.flexslider.js"></script>
            <!--  Scrolling Reveal Script -->
            <script src="assets/js/scrollReveal.js"></script>
            <!--  Scroll Scripts -->
            <script src="assets/js/jquery.easing.min.js"></script>
            <!--  Custom Scripts -->
            <script src="assets/js/custom.js"></script>
            </body>
            </html>
```



4.1.3 Generate TimeTable Code and Output

```
<?php
// Start the session
session_start();
if (isset($_GET['success'])) {
   echo "<script type='text/javascript'>alert('Time Table Generated');</script>";
}
?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <meta charset="utf-8"/>
   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1"/>
   <meta name="description" content=""/>
   <meta name="author" content=""/>
   <title>TimeTable Management System</title>
   <script type="text/javascript" src="assets/jsPDF/dist/jspdf.min.js"></script>
   <script type="text/javascript" src="assets/js/html2canvas.js"></script>
   <!-- BOOTSTRAP CORE STYLE CSS -->
   <link href="assets/css/bootstrap.css" rel="stylesheet"/>
   <!-- FONT AWESOME CSS -->
   <link href="assets/css/font-awesome.min.css" rel="stylesheet"/>
   <!-- FLEXSLIDER CSS -->
   <link href="assets/css/flexslider.css" rel="stylesheet"/>
   <!-- CUSTOM STYLE CSS -->
   <link href="assets/css/style.css" rel="stylesheet"/>
   <!-- Google     Fonts -->
   <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,700,300' rel='stylesheet'
type='text/css'/>
</head>
<body>

<div class="navbar navbar-inverse navbar-fixed-top " id="menu">
   <div class="container">
      <div class="navbar-header">
         <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-
collapse">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
         </button>

      </div>
      <div class="navbar-collapse collapse move-me">
         <ul class="nav navbar-nav navbar-left">
            <li><a href="addteachers.php">ADD TEACHERS</a></li>
            <li><a href="addsubjects.php">ADD SUBJECTS</a></li>
            <li><a href="addclassrooms.php">ADD CLASSROOMS</a></li>
            <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" aria-
expanded="false">ALLOTMENT
               <span class="caret"></span></a>
            <ul class="dropdown-menu">
               <li>
                  <a href=allotsubjects.php>THEORY COURSES</a>
               </li>
               <li>
                  <a href=allotpracticals.php>PRACTICAL COURSES</a>
               </li>
               <li>
                  <a href=allotclasses.php>CLASSROOMS</a>
               </li>
            </ul>
```

```
            </li>
            <li><a href="generatetimetable.php">GENERATE TIMETABLE</a></li>

          </ul>
          <ul class="nav navbar-nav navbar-right">
            <li><a href="index.php">LOGOUT</a></li>
          </ul>

        </div>
      </div>
</div>
<!--NAVBAR SECTION END-->
<br>


<!--Algorithm Implementation-->
<div id="myModal" class="modal">

    <!-- Modal content -->
    <div class="modal-content">
        <div class="modal-header">
            <span class="close">&times</span>
            <h2 id="popupHead">Assign Substitute</h2>
        </div>
        <div class="modal-body" id="AssignSubstitute">
            <!--Admin Login Form-->

            <div style="display:block" id="assignSubstituteForm">
                <form method="post" action="assignSubstituteFormValidation.php">
                    <div class="form-group">
                        <label for="substitute">Substitute</label>
                        <select class="form-control" id="substitute" name="SB">

                        </select>
                        <input type="hidden" id="cell_number" class="btn btn-default" name="CN">

                    </div>
                    <div align="right" class="form-group">

                        <input type="submit" id="submit" class="btn btn-default" name="ADD" value="CHECK">
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>
<script>
    var assignsubstitueForm = document.getElementById("assignSubstitueForm");
    // Get the <span> element that closes the modal
    var modal = document.getElementById('myModal');
    var span = document.getElementsByClassName("close")[0];
    span.onclick = function () {
        modal.style.display = "none";
        assignsubstitueForm.style.display = "none";
    }

    // When the user clicks anywhere outside of the modal, close it
    window.onclick = function (event) {
        if (event.target == modal) {
            modal.style.display = "none";
            assignsubstitueForm.style.display = "none";
        }
```

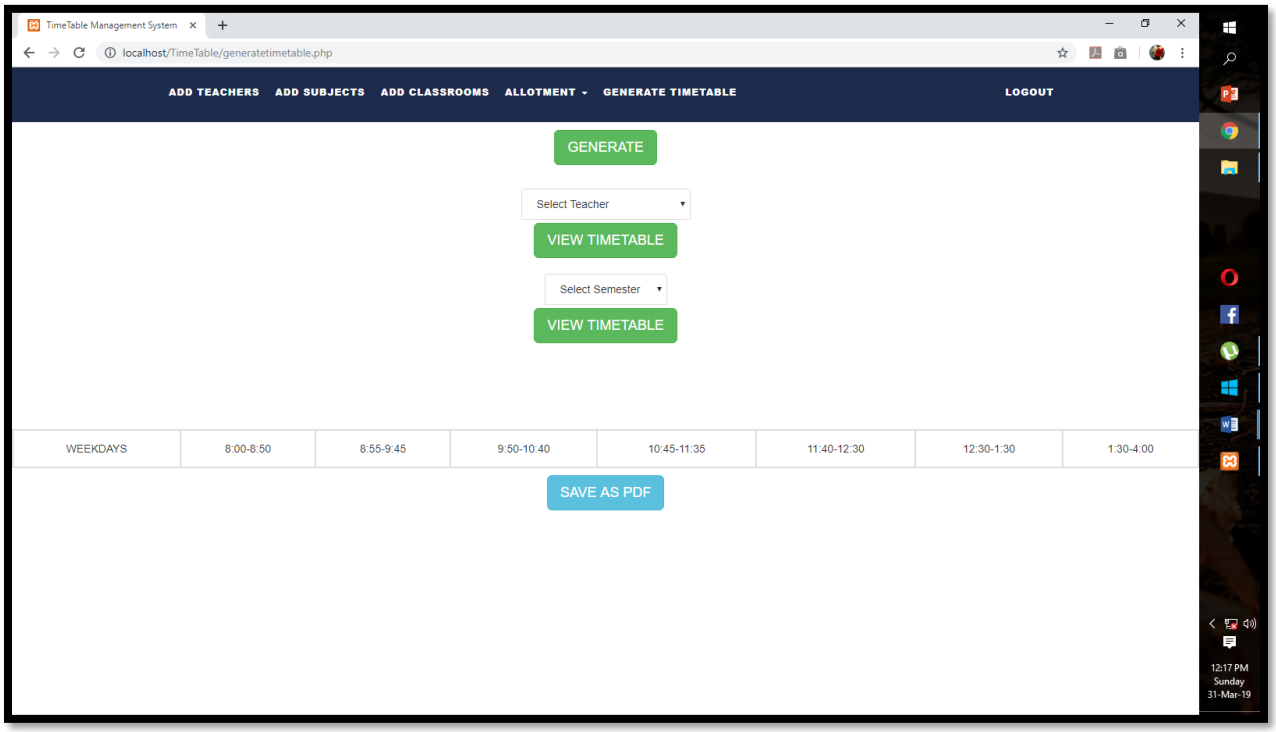
```
        }
    </script>
    <form action="algo.php" method="post">
        <div align="center" style="margin-top: 50px">
            <button type="submit"
                id="generatebutton" class="btn btn-success btn-lg">GENERATE
            </button>
        </div>
    </form>
    <form action="generatetimetable.php" method="post">
        <div align="center" style="margin-top: 30px">
            <select name="select_teacher" class="list-group-item">
                <option selected disabled>Select Teacher</option>
                <?php
                $q = mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
                    "SELECT * FROM teachers ");
                while ($row = mysqli_fetch_assoc($q)) {
                    echo " \"<option value=\"{$row['faculty_number']}\">{$row['name']}</option>\"";
                }
                ?>

            </select>
            <button type="submit" id="viewteacher" class="btn btn-success btn-lg" style="margin-top:
5px">VIEW TIMETABLE
            </button>
        </div>
    </form>
    <form action="generatetimetable.php" method="post">
        <div align="center" style="margin-top: 20px">
            <select name="select_semester" class="list-group-item">
                <option selected disabled>Select Semester</option>
                <option value="3">1</option>
                <option value="4">2</option>
                <option value="5">3</option>
                <option value="6">4</option>
                <option value="7">5</option>
                <option value="8">6</option>
            </select>
            <button type="submit" id="viewsemester" class="btn btn-success btn-lg" style="margin-top:
5px">VIEW TIMETABLE
            </button>
        </div>
    </form>
    <script>
        var index = -1;
        function Substitute() {
            var table = document.getElementById("timetable");
            var cells = table.getElementsByTagName("td");
            // window.alert(cells[3].innerHTML.toString());
            for (i = 0; i < cells.length; i++) {
                if (i % 8 == 6 || i % 8 == 7 || parseInt(i / 8) == 0 || i % 8 == 0) {
                    continue;
                }
                var currentCell = cells[i];
                //var b = currentRow.getElementsByTagName("td")[0];
                var createSubstituteHandler =
                    function (cell, i) {
                        return function () {

                            document.getElementById('cell_number').value = i;
                            index = i;
                            var xmlhttp = new XMLHttpRequest();
```

```
                        xmlhttp.onreadystatechange = function () {
                            if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
                                var modal = document.getElementById('myModal');
                                modal.style.display = "block";
                                document.getElementById("substitute").innerHTML = this.responseText;

                            }
                        };
                        xmlhttp.open("GET", "getcellindex.php?i=" + i, false);
                        xmlhttp.send();
                    };
                };
                currentCell.onclick = createSubstituteHandler(currentCell, i);
            }
        }
    </script>

    <div>
        <br>
        <style>
            table {
                margin-top: 20px;
                font-family: arial, sans-serif;
                border-collapse: collapse;
                width: 100%;
            }

            td, th {
                border: 2px solid #dddddd;
                text-align: left;
                padding: 8px;
            }

            tr:nth-child(even) {
                background-color: #ffffff;
            }

            tr:nth-child(odd) {
                background-color: #ffffff;
            }
        </style>
        <div id="TT" style="background-color: #FFFFFF">
            <table border="2" cellspacing="3" align="center" id="timetable">
                <caption><strong><br><br>
                    <?php
                    if (isset($_POST['select_semester'])) {
                        echo "COMPUTER ENGINEERING DEPARTMENT SEMESTER " .
$_POST['select_semester'] . " ";
                        $year = (int)($_POST['select_semester'] / 2) + $_POST['select_semester'] % 2;
                        $r = mysqli_fetch_assoc(mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
"SELECT * from classrooms
                        WHERE status = '$year'"));
                        echo " ( " . $r['name'], " ) ";
                    } else if (isset($_POST['select_teacher'])) {
                        $id = $_POST['select_teacher'];
                        $r = mysqli_fetch_assoc(mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
"SELECT * from teachers
                        WHERE faculty_number = '$id'"));
                        echo $r['name'];
                    } else if (isset($_GET['display'])) {
                        $id = $_GET['display'];
                        $r = mysqli_fetch_assoc(mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
```

```
"SELECT * from teachers
                        WHERE faculty_number = '$id'"));
                        echo $r['name'];

                    }
                    ?>
                </strong></caption>
            <tr>
                <td style="text-align:center">WEEKDAYS</td>
                <td style="text-align:center">8:00-8:50</td>
                <td style="text-align:center">8:55-9:45</td>
                <td style="text-align:center">9:50-10:40</td>
                <td style="text-align:center">10:45-11:35</td>
                <td style="text-align:center">11:40-12:30</td>
                <td style="text-align:center">12:30-1:30</td>
                <td style="text-align:center">1:30-4:00</td>
            </tr>
            <tr>
                <?php
                $table = null;
                if (isset($_POST['select_semester'])) {
                    $table = " semester" . $_POST['select_semester'] . " ";
                } else if (isset($_POST['select_teacher'])) {
                    $table = " " . $_POST['select_teacher'] . " ";
                } else if (isset($_GET['display'])) {
                    $table = " " . $_GET['display'] . " ";
                } else
                    echo '</table>';
                if (isset($_POST['select_semester']) || isset($_POST['select_teacher']) || isset($_GET['display'])) {
                    $q = mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
                        "SELECT * FROM" . $table);
                    $qq = mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
                        "SELECT * FROM subjects");
                    $days = array('MONDAY', 'TUESDAY', 'WEDNESDAY', 'THURSDAY', 'FRIDAY',
'SATURDAY');
                    $i = -1;
                    $str = "<br>";
                    $tid = "";
                    if (isset($_POST['select_semester'])) {
                        while ($r = mysqli_fetch_assoc($qq)) {
                            if ($r['isAlloted'] == 1 && $r['semester'] == $_POST['select_semester']) {
                                $str .= $r['subject_code'] . ": " . $r['subject_name'] . ", ";
                                if (isset($r['allotedto'])) {
                                    $id = $r['allotedto'];
                                    $qqq = mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
                                        "SELECT * FROM teachers WHERE faculty_number = '$id'");
                                    $rr = mysqli_fetch_assoc($qqq);
                                    $str .= " " . $rr['alias'] . ": " . $rr['name'] . " ";
                                }
                                if ($r['course_type'] !== "LAB") {
                                    $str .= "<br>";
                                    continue;
                                } else {
                                    $str .= ", ";
                                }
                                if (isset($r['allotedto2'])) {
                                    $id = $r['allotedto2'];
                                    $qqq = mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
                                        "SELECT * FROM teachers WHERE faculty_number = '$id'");
                                    $rr = mysqli_fetch_assoc($qqq);
                                    $str .= " " . $rr['alias'] . ": " . $rr['name'] . ", ";
                                }
```

```
                        if (isset($r['allotedto3'])) {
                            $id = $r['allotedto3'];
                            $qqq = mysqli_query(mysqli_connect("localhost", "root", "", "ttms"),
                                "SELECT * FROM teachers WHERE faculty_number = '$id'");
                            $rr = mysqli_fetch_assoc($qqq);
                            $str .= " " . $rr['alias'] . ": " . $rr['name'] . "<br>";
                        }
                    }
                }
            } else if (isset($_POST['select_teacher']) || isset($_GET['display'])) {
                if (isset($_POST['select_teacher'])) {
                    $tid = $_POST['select_teacher'];
                } else if (isset($_GET['display'])) {
                    $tid = $_GET['display'];
                    $tid = strtoupper($tid);
                }
                while ($r = mysqli_fetch_assoc($qq)) {
                    if ($r['isAlloted'] == 1 && $r['allotedto'] == $tid) {
                        $str .= $r['subject_code'] . ": " . $r['subject_name'] . " <br>";
                    } else if ($r['isAlloted'] == 1 && isset($r['allotedto2']) && $r['allotedto2'] == $tid) {
                        $str .= $r['subject_code'] . ": " . $r['subject_name'] . " <br>";
                    } else if ($r['isAlloted'] == 1 && isset($r['allotedto3']) && $r['allotedto3'] == $tid) {
                        $str .= $r['subject_code'] . ": " . $r['subject_name'] . " <br>";
                    }
                }
            }
            while ($row = mysqli_fetch_assoc($q)
            echo '</table>';
            $sign = "GENERATED VIA TIMETABLE MANAGEMENT SYSTEM, DEPARTMENT OF
COMPUTER SCIENCE, IITM Janakpuri.";
            echo "<div style='margin-left: 10px' align='center'>" . "<br>" . $str . "<br></div>" .
                "<div style='margin-left: 10px' align='center'>" . "<strong>" . $sign .
"<br></strong></div>";
        }
        if (isset($_POST['select_teacher'])) {
            echo "<script>Substitute();</script>";
            $_SESSION['shown_id'] = $_POST['select_teacher'];
        }
        if (isset($_GET['display'])) {
            echo "<script>Substitute();</script>";
            $_SESSION['shown_id'] = $_GET['display'];
        }
        ?>
    </div>
</div>
<script type="text/javascript">
    function gendf() {
        var doc = new jsPDF();

        doc.addHTML(document.getElementById('TT'), function () {
            doc.save('<?php
                if (isset($_POST["select_semester"])) {
                    echo "ttms semester " . $_POST["select_semester"];
                } else if (isset($_POST["select_teacher"])) {
                    echo "ttms " . $_POST["select_teacher"];
                } else if (isset($_GET["display"])) {
                    echo "ttms " . $_GET["display"];
                }
                ?>' + '.pdf');
            alert("Downloaded!");

        });
```

```
    }

</script>
<div align="center" style="margin-top: 10px">
    <button id="saveaspdf" class="btn btn-info btn-lg" onclick="gendf()">SAVE AS PDF</button>
</div>

<!--HOME SECTION END-->

<!--<div id="footer">
    <!--  &copy 2014 yourdomain.com | All Rights Reserved |  <a href="http://binarytheme.com"
style="color: #fff" target="_blank">Design by : binarytheme.com</a>
-->
<!-- FOOTER SECTION END-->

<!--  Jquery Core Script -->
<script src="assets/js/jquery-1.10.2.js"></script>
<!--  Core Bootstrap Script -->
<script src="assets/js/bootstrap.js"></script>
<!--  Flexslider Scripts -->
<script src="assets/js/jquery.flexslider.js"></script>
<!--  Scrolling Reveal Script -->
<script src="assets/js/scrollReveal.js"></script>
<!--  Scroll Scripts -->
<script src="assets/js/jquery.easing.min.js"></script>
<!--  Custom Scripts -->
<script src="assets/js/custom.js"></script>
</body>
</html>
```



4.2 System Testing & Debugging:

4.2.1 FUNCTIONAL TESTING

Test Case ID	Description	Input	Expected Output
Tc1	1. LogIn on the website as an admin.	Provide ID and Password.	Time Table Scheduling Screen.
	2. LogIn on the website as a Faculty.	Provide Teacher ID.	Timetable Screen.
Tc2	Add Teacher	Provide Teacher Details	Teacher Added to Database.
Tc3	Add Subject	Provide Subject Details.	Subject Information added to database.
Tc4	Add Room	Provide Room number	Room Number added to database
Tc5	Allotment of subjects to teacher according to theory and lab subjects.	Select Faculty and allot him/her a subject.	Subject Alloted.
Tc6	Generate Timetable	Click on Generate Button.	Timetable generated.

Table 4.1: Functional Testing

4.2.2 User Interface Testing

User interface testing checklist may provide the opportunities for testers to ensure the proper functioning of user interface features such as hyperlinks, tables, frames, forms, buttons, menus, dialog boxes and error messages. A checklist may also discipline and organize testing activities and a sample checklist is given below. However organizations and website owners may modify the checklist according to requirements of their web application.

S. no	Description	Yes/ no/ NA	Remarks
Hyperlinks			
1.	Are the links meaningful?	Yes	
2.	Are there any broken links?	No	
3.	Do all internal links work correctly?	Yes	
4.	Do all external links work correctly?	Yes	
5.	Are all links to external sites in the website tested?	Yes	
6.	Are images correctly hyperlinked?	Yes	
7.	Does every hyperlink exist on the site map?	Yes	
8.	Are the hyperlink's colours standard	Yes	
9.	Does the link bring the user to the correct web page?	Yes	
10.	Can the user navigate using text only?	Yes	
Tables			
11.	Are the columns wide enough or the text wraps around the rows?	Yes	
12.	Are the row and columns headings of tables appropriate?	Yes	
13.	Are the complex tables broken down into simpler ones, wherever required?	Yes	
14.	Does the user have to scroll right constantly in order to see the contents in a table?	Yes	
15.	Are the captions meaningful?	Yes	
Frames			
16.	Is every frame associated with a title?	Yes	
17.	Can the user resize the frame?	No	

18.	Is the frame size appropriate?		
19.	Does the horizontal and vertical scrollbar appear wherever required?	No	
20.	Does any frame handling mechanism exist for browsers that do not support frames?	No	
Forms			
21.	Are keyboard shortcuts provided for movement between different fields of forms?		
22.	Are the mandatory fields marked clearly?	NA	
23.	Are descriptive labels for all fields provided?	Yes	
24.	Is information formatted , wherever required?	Yes	
25.	Are error messages meaningful and appropriate?	Yes	
26.	Does the size of text fields give enough room for the user to type?	Yes	
27.	Are fields used appopraitely?	Yes	
28.	Is any information asked more than once in the form? Is the user prevented from entering the same data multiple times?	Yes	
29.	Does the form include 'reset' button to clear its contents?		
Text fields, buttons, list boxes, check boxes			
30.	Does the text field accept invalid characters and special characters?	No	
31.	Can text be selected using shift + arrow key?		
32.	Is the user able to select any combination of options in check boxes?		
33.	Can the user select more than 1 option in radio buttons?	No	
34.	Does the button click trigger the required action?		
35.	Can the user add text in the combo boxes?		
36.	Can the user add text in the list boxes?		
37.	Do the required commands and options exist in each menu?		
38.	Are abbreviations used in list boxes/ buttons?		
39.	Are the label names meaningful?		
40.	Are mouse actions consistent across web pages?		
41.	Is red colour used to highlight active items?		
42.	Is all the data inside the list/ combo box listed in chronological order?		
43.	Are validation checks for text fields present?		
44.	Do fields with numeric values handle upper and lower range of values appropriately?(BVA)		
45.	Does the back navigation button work as		

	required?		
46.	Do the text fields accept maximum permissible data?	Yes	
47.	Can an alphanumeric character be entered in numeric fields?	No	
48.	Are the command buttons disabled when they are not in use?	No	
49.	Is there any spelling or grammatical mistakes in captions or labels?	No	

Table 4.2: Checklist for testing user interfaces

4.2.3 Navigation Testing

Test case id	Description	Inputs	Expected output
Tc1	Check all links on each web page.	Link 1= home Link 2= about us Link 3= contact us Link 4=Google+ Link 5= Twitter Link 6= Facebook	Appropriate web page is opened with respect
Tc 2	Generate Time Table	Faculty, Subject and Room are provided	Time table is Generated.
Tc3	Click on 'back' link present on each page.		The appropriate page is displayed.

Table 4.3 : Test cases for Time Table Generation

4.2.4 Database Testing

Web applications, many applications are database driven. It is important for these applications to provide security to the user's sensitive data such as personal details and credit card information.

For example, consider the example for purchasing items from an online store. The user performs a search based on some keywords and price preferences; the database server initiates a database query. Suppose due to some programming fault in the query, the query does not consider the price preferences given by the customer, this will produce erroneous results. These types of faults must be tested and removed during database testing.

Important issues in database testing may include:

i. Data validation
ii. Data consistency
iii. Data integrity
iv. Concurrency control and recovery
v. Data manipulation operations such as addition, deletion, updating and retrieval of data.
vi. Database security

A database must be tested for administrative level operations such as adding, deleting and updating an item in the database, and user operations such as searching an item from the database or providing personal details. In the example of the online shopping website, the most common administrative operations and user operations include:

a. Administrative operations
    a. Inserting a new item into the database
    b. Deleting an existing item from the database
    c. updating an existing item from the database
    d. viewing an existing item from the database
b. user operations
    a. searching items from the database
    b. registering into the website involves storing the user's personal details
    c. placing an order involves storing user preferences and purchase details into the database
    d. providing feedback involves storing information in the database
    e. tracking status of the order placed

In database testing, the following aspects of correctness must be ensured:

i. Are the database operations performed correctly?
ii. Is concurrent user's access to the database handled correctly?
iii. Is the database fault tolerant?
iv. Are the performance requirements such as throughput and response time met?
v. Are backup and recovery procedures designed and ensure uninterrupted services to the user?
vi. Does the database restore to a consistent state after crash recovery?

Database testing may include generation of new records, monitoring of system performance and verification of performance of the database processor.

CHAPTER-5

Scope of Improvement

Timetable Generation System generates timetable for each class and teacher, in keeping with the availability calendar of teachers, availability and capacity of physical resources (such as classrooms, laboratories and computer room) and rules applicable at different classes, semesters, teachers and subjects level. Best of all, this Timetable Generation System tremendously improves resource utilization and optimization.

In future, we will connect the Knowledge Portal of College with this website, so students and faculties can have fully access to all the information related to their courses and subjects respectively.

Summary

Timetable Generation System generates timetable for each class and teacher, in keeping with the availability calendar of teachers, availability and capacity of physical resources (such as classrooms, laboratories and computer room) and rules applicable at different classes, semesters, teachers and subjects level. Best of all, this Timetable Generation System tremendously improves resource utilization and optimization.

Our Timetabling Algorithm is main component of our project which produces he HTML based timetable even / odd semester sheet as the output. Our project takes various inputs from the user such as Teacher List, Course List, Semester List, Room List, Day List and Timeslot as well as various rules, facts and constraints using web based forms, which are stored in XML based knowledge base. This knowledge base serves as input to our Timetable Generator Algorithm residing on server machine. Our knowledgebase is in the middle, because it is between our timetabling algorithm and GUI front end which is designed in the last.

Conclusion

Timetabling concerns all activities with regard to producing a schedule that must be subjective to different constraints. Timetable can be defined as the optimization of given activities, actions or events to a set of objects in space-time matrix to satisfy a set of desirable constraints.

A college timetable is a temporal arrangement of a set of lectures and classrooms in which all given constraints are satisfied. Creating such timetables manually is complex and time-consuming process.

By automating this process with computer assisted timetable generator can save a lot of precious time of administrators who are involved in creating and managing course timetables. Since every college has its own timetabling problem, the commercially available software packages may not suit the need of every college.

Hence we have developed practical approach for building lecture course timetabling system, which can be customized to fit to any colleges timetabling problem. This project introduces a practical timetabling algorithm capable of taking care of both strong and weak constraints effectively, used in an automated timetabling system. So that each teacher and student can view their timetable once they are finalized for a given semester but they can't edit them.

Time Table Generator is a convenient time table managing website .Time table management may be aided by a range of skills, tools, and techniques used to manage time table when accomplishing specific subjects, semesters, and students. Initially, time table management referred to just work activities, but eventually the term broadened to include personal activities as well. A time table management system is a designed combination of processes, tools, techniques, and methods.

REFERENCES.

1. D. G. Maere, (2010). “How Working Group Automated Timetabling was founded”, retrieved from http://www.asap.ac.nott.ac.uk/, 2010, Last accessed date 9th December 2011.
2. Bhadhuri a “University timetable scheduling”. Advance in recent technologies in communication and computing. ARTCom ’09.International conference.
3. https://www.google.com/search?client=firefox-b-d&q=PHP+PROJECTS+LINK (24TH MARCH 2019)
4. https://www.google.com/search?safe=active&client=firefox-b-d&ei=VNekXIeSEYX2rQG8hJ4Y&q=phpmyadmin&oq=PHP&gs_l=psy-ab.1.4.0i131i67j0i131i273l2j0i273l2j0i131i67j0i273j0i67l3.10900.14991..78669...0.0..0.466.3633.0j5j6j1j2....1..0....1..gws-wiz (3RD APRIL 2019)
5. https://www.google.com/search?client=firefox-b-d&q=GITHUB (3RD APRIL 2019)
6. https://www.google.com/search?safe=active&client=firefox-b-d&ei=YdikXNqTONb39QPDro4Y&q=time+table+management+system+source+code&oq=TIMETABLE+SYSTEM&gs_l=psy- (3RD APRIL 2019)
7. https://www.slideshare.net/AdityaJain335/time-table-management-system-software-report. (3RD APRIL 2019)