# **Project Report on**

# "Time Table Management System"

# **GOVERNMENT POLYTECHNIC MUMBAI**



# **PROJECT BY:**

Pradip Kayande (FD18IF007)
Vinay Machha (FS17IF028)
Mihir Pawar (FS17IF028)

# **SUPERVISOR AND GUIDANCE:**

MRS. Dipali Gosawi Madam

GOVERNMENT POLYTECHNIC MUMBAI
2019-2020

# TABLE OF CONTENTS

# Content

# CHAPTER 1 - Software Project Management Plan (SPMP)

- 1.Introduction
  - 1.1 Project Overview
  - 1.2 Purpose
- 2.Project Organization
  - 2.1 Software Process Model
  - 2.2 Roles and Responsibilities
  - 2.3 Tools and Techniques
- 3. Project Management Plan
  - 3.1 Tasks
  - 3.2 Information Gathering
  - 3.3 Resources Needed
  - 3.4 Dependencies and Constraints
  - 3.5 Risks and Contingencies
- 4.Project Scheduling
  - 4.1Gantt Chart

# CHAPTER 2 - Software Requirements Specifications (SRS)

- 1.Introduction
  - 1.1 Project Overview
- 2. Specific Requirements
  - 2.1 User Interfaces
- 3. Product Description
  - 3.1 Objective
  - 3.2 Intended Audience
  - 3.3 Project Scope
  - 3.4 Benefits over previous systems
- 4. Functionalities
- 5. Features
- 6. Project Justification

# CHAPTER 3 - Software Design Description (SDD) 1.Introduction

- 1.1 Design Overview
- 1.2 UML Diagram
- 1.2.1 Use Case Diagram
- 1.2.2 Class Diagram
- 1.3 Database Design
- 1.4 User Interface
  - 1.4.1 Desktop Application
  - 1.4.2 Web Application

# CHAPTER 4 - Software Test Documentation (STD)

- 1.Testing Objectives
- 2. Quality dimensions of web application testing
- 3. Possible errors in a web-app design
- 4. Possible error aspects in our project
- 5. Testing strategies
- 6. The testing process
- 7. Database testing
- 8. User interface testing
- 9. Testing interface mechanisms
- 10. Basis path testing and Cyclomatic complexity

# **ABSTRACT**

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.

# CHAPTER 1- SOFTWARE PROJECT MANAGEMENT PLAN

## 1. INTRODUCTION

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 key factor in running an educational centre or basically an academic environment is the need for a well-planned, well-throughout and clash-free timetable. Back in the days when technology was not in wide use, (lecture) timetables were manually created by the academic institution.

Every school year, tertiary institutions are faced with the tedious task of drawing up academic timetables that satisfies the various courses and the respective examination being offered by the different departments.

Timetable development process starts when each Head of Department provide the following information to be used for timetable scheduling. The information provides the modules with dates, time and venues suitable in a particular semester:

- Examinable courses in a particular semester.
- Dates for lectures to be held (Lectures can be scheduled between Monday and Friday).
- Specified time for lectures (i.e. Between 8am and 4pm)
- The venue of the scheduled lectures.

A timetabling problem consists of four parameters and they are:

- T (set of time),
- R (set of available resources),
- M (set of scheduled contacts) and
- C (set of constraints).

This problem assigns time and resources to the contacts on such a way that the constraints will be satisfied. In various timetabling problems, educational timetabling has been generally examined from practical standpoint.

The quality of the timetable determines the quality of time dedicated by lecturers, students and administrators to academic activities. Various academic timetabling includes:

- School timetable
- Lecture timetable
- Examination timetable and
- Assignment timetable.

## 1.1 PROJECT OVERVIEW

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.

#### 1.2 PURPOSE

Planning timetables is one of the most complex and error-prone applications. The available system currently builds or generates a set of timetables manually, but most times have issues with generating a clash-free and complete timetable. There are still serious problems like generation of high cost time tables are occurring while scheduling and these problems are repeating frequently. Most educational institutions have resorted to manual generation of their timetables which according to statistics takes much time to get completed and optimal. Even at the optimal stage of the manually generated timetable, there are still a few clashes and it is the lecturer that takes a clashing course that works out the logistics of the course so as to avoid the clash. Therefore there is a great requirement for an application distributing the course evenly and without collisions. Our aim here is to develop a simple, easily understandable, efficient and portable application, which could automatically generate good quality timetables within seconds.

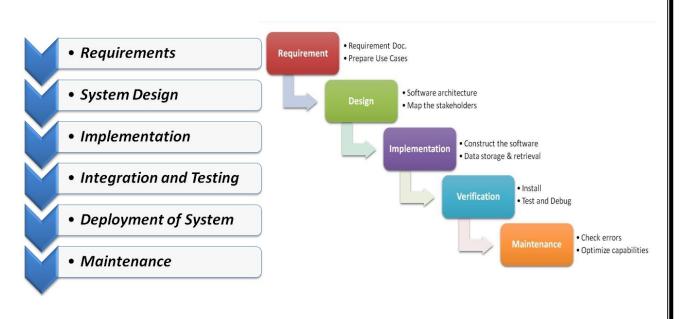
# 2. PROJECT ORGANIZATION

#### 2.1 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. The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance.

As the Waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a Linear-Sequential Life Cycle Model.



#### SEQUENTIAL PHASES IN WATERFALL MODEL

- **Requirements:** The first phase involves understanding what need to be design and what is its function, purpose etc. Here, the specifications of the input and output or the final product are studied and marked.
- **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. The software code to be written in the next stage is created now.
- **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. The software designed, needs to go through constant software testing to find out if there are any flaw or errors. Testing is done so that the client does not face any problem during the installation of the software.
- **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: This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. Client is provided with regular maintenance and support for the developed software.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model".

# 2.2 Roles and Responsibilities

#### **User Roles:**

- Users can access the website/app, they can read the content.
- Users would be able to give feedback.
- Users would be able to manage their account and see their time-table.

#### **Admin Roles:**

- Admin can view all the content (All the timetables i.e. personal as well as general).
- If the need be, admin can remove a timetable from the website/app.
- Admin can view feedback given by users, and solve their queries.
- Admin has the authority to manage the database.

# 2.3 Tools and Techniques

The tools and techniques used are mentioned as:

# **Tools used:**

1. Designing: XAMPP/WAMPP

2. Coding: Notepadd++ / PHP Storm

3. Database: MySQL

4. Creating UML Diagrams: ArgoUML

# **Techniques used:**

Front-End: HTML, CSS
 Back-End: JavaScript, PHP

3. Database: MySQL

# 3. Project Management Plan

#### 3.1 Tasks

The major tasks are:

- Analysis of major deliverables
- Storing the data in database
- Documentation and modelling
- Approval to documentation
- Creating a user interface
- Coding for interface
- Testing the project

# 3.2 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.

#### 3.3 Resources Needed

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

• Hard Disk: Minimum 40 GB or Higher

• Ram: 512MB or Higher

A Web Application on "Time Table Generator-plan ahead" is built up using following techniques:

#### • FRONT-END

- a. HTML
- b. CSS
- c JavaScript
- d. JQuery

## BACK-END

a. PHP

# DATABASE

a. MySQL and PHPMyAdmin

A Desktop Application on "Time Table Generator-plan ahead" is built up using following techniques:

#### > Platforms:

- Apache Server / Tomcat 7.0
- MySQL

# ➤ IDE's:

- Noepad++
- PHP Storm
- MySQL PHPMyAdmin

#### 3.4 CONSTRAINTS

#### 3.4.1 HARDCONSTRAINTS

- Class can only be assigned if room is spare.
- No teacher or student have more than one class at a time.
- A classroom must have enough seats to accommodate all students.

#### 3.4.2 SOFTCONSTRAINTS

- Time of class preferred by teacher.
- Specific Room preferred by any teacher.
- Break time between different lectures preferred by teacher.
- Lesswalk to attendlectures by student.

# 3.5 Risks and Contingencies

Risks are identified, classified and managed before actual execution of program. These risks are classified in different categories.

## Schedule Risk:

Project schedule get slip when project tasks and schedule release risks are not addressed properly. Schedule risks mainly impact on project and finally on company economy and may lead to project failure. Schedules often slip due to following reasons:

- Wrong time estimation.
- Resources are not tracked properly.
- Failure to identify complex functionalities.

## Budget Risk:

- Wrong budget estimation.
- Cost overruns
- Project scope expansion

#### Operational Risks:

- Risks of loss due to improper process implementation
- Failed system or some external events risks.
- Causes of Operational risks:
- Failure to address priority conflicts.
- Failure to resolve the responsibilities.

#### • Technical risks:

Technical risks generally leads to failure of functionality and performance. Causes of technical risks are:

- Continuous changing requirements.
- ➤ No advanced technology available or the existing technology is in initial stages.
- > Product is complex to implement.
- > Difficult project modules integration.

# • External Risks:

These are the external risks beyond the operational limits. These are all uncertain risks are outside the control of the program. These external events can be:

- > Running out of fund.
- Market development.
- > Changing customer product strategy and priority.
- Government rule changes.

# 4. Project Scheduling

# 4.1 Gantt Chart

|                | September 2019<br>(First Week) | October 2019<br>(Second Week) | November 2019<br>(Third Week) |
|----------------|--------------------------------|-------------------------------|-------------------------------|
| Planning       |                                |                               |                               |
| Research       |                                |                               |                               |
| Designing      |                                |                               |                               |
| Implementation |                                |                               |                               |
| Testing        |                                |                               |                               |

# CHAPTER 2- SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)

#### 1. INTRODUCTION

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 key factor in running an educational centre or basically an academic environment is the need for a well-planned, well-throughout and clash-free timetable. Back in the days when technology was not in wide use, (lecture) timetables were manually created by the academic institution.

Every school year, tertiary institutions are faced with the tedious task of drawing up academic timetables that satisfies the various courses and the respective examination being offered by the different departments.

Timetable development process starts when each Head of Department provide the following information to be used for timetable scheduling. The information provides the modules with dates, time and venues suitable in a particular semester:

- Examinable courses in a particular semester.
- Dates for lectures to be held (Lectures can be scheduled between Monday and Friday).
- Specified time for lectures (i.e. Between 8am and 4pm)
- The venue of the scheduled lectures.

A timetabling problem consists of four parameters and they are:

- T (set of time),
- R (set of available resources),
- M (set of scheduled contacts) and
- C (set of constraints).

This problem assigns time and resources to the contacts on such a way that the constraints will be satisfied. In various timetabling problems, educational timetabling has been generally examined from practical standpoint.

The quality of the timetable determines the quality of time dedicated by lecturers, students and administrators to academic activities. Various academic timetabling includes:

- School timetable
- Lecture timetable
- Examination timetable and
- Assignment timetable.

#### 1.1 PROJECT OVERVIEW

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.

#### 2. PURPOSE

Planning timetables is one of the most complex and error-prone applications. The available system currently builds or generates a set of timetables manually, but most times have issues with generating a clash-free and complete timetable. There are still serious problems like generation of high cost time tables are occurring while scheduling and these problems are repeating frequently. Most educational institutions have resorted to manual generation of their timetables which according to statistics takes much time to get completed and optimal. Even at the optimal stage of the manually generated timetable, there are still a few clashes and it is the lecturer that takes a clashing course that works out the logistics of the course so as to avoid the clash. Therefore there is a great requirement for an application distributing the course evenly and without collisions. Our aim here is to develop a simple, easily understandable, efficient and portable application, which could automatically generate good quality timetables within seconds.

# 3. SPECIFIC REQUIREMENTS

#### 3.1 USER INTERFACES

**User:** The User is one who intends to view the Time Table i.e. it can be Student, Teachers, Course/Subject Co-ordinator and even the Head of Department. Multiple users can connect through the software for viewing the time table.

**Admin:** She/he is the one that uses the Software to generate Time-Table according to the needs and specific requirements of the Class for which the time table needs to be formed.

#### 4. PRODUCT DESCRIPTION

#### **4.1 OBJECTIVE**

The main objectives of our project are:

- 1 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.
- 2 To make a timetable system generic so that I can work equally well for different School, Colleges and Universities.
- 3 User defined constraints handling.
- 4 Ease of use for user of system so that he/she can make automatic timetable.
- 5 Focus on optimization of resources i.e. teachers, labs and rooms etc.
- 6 Provide a facility for everyone to view timetable.
- 7 Generate multiple useful views from time table.

#### 4.2 INTENDED AUDIENCE

Intended audience of the project comprises of any individual belonging to organization like school, college, or university who needs to design the schedule of the working of the employees or the schedule of any course curriculum.

# **4.3 PROJECT 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.

#### **4.4 BENEFITS OVER PREVIOUS SYSTEM**

The Timetable Generator System allows users to generate time table for newly occurring changes in less time, with less effort and with more efficiency. It will allow users to work on and view time tables in different platforms and view different information simultaneously. The Software will create Individual Schedules also i.e. it will create Academic Schedule, Examination Schedule, Class Schedule, Professor Schedules individually.

#### 5. FUNCTIONALITIES

The Software made will have Following Functionalities. Some of the Major Functionalities are stated below-

- Creating Recess and Prayer slots
- Generate Multiple Time Tables
- Information Storing in Data-Base
- Vanquishes human errors
- Reduces human efforts
- Academic Calendar
- Systematic planning
- Personal Scheduler
- Lectures Scheduling
- Course Scheduling
- Exam Scheduling

# 6. FEATURES

- **Interface for input:** The system will be having an easy to use and interactive interface to enter all the inputs like the workload for the teachers, how many no. of students of which branches are studying the subject in a semester, the data for the rooms and data for the labs.
- Database Capabilities: The system will have well-designed database to store all the information which
  will be entered in as the input. Separate database maintaining basic information's, subjects, teachers,
  batches and their associations and other details Database for holding generated timetable and for
  storing required timetables.
- Processing Capabilities: The system will have algorithms to process all the data present in the database
  and keeping in view the various constraints like that a teacher should not have two consecutive
  lectures/labs, students have minimum one hour gaps, proper rooms are allocated for the lectures
  and tutorials, labs are used optimally so that they are used for the maximum possible time, it will
  generate the time table.
- **Search Panel:** The system will have an easy to use to search panel to search according his need on the time tables stored in the database. The system would give the response to the user quires in the proper format and errors messages will be shown properly to tell user about his mistakes and to guide him/her for proper use of the system.

- Features for assigning priorities for subjects.
- Features for editing generated table, saving edited tables and opening saved tables.
- High portability, works on almost all systems available.
- Highly efficient, needs only few minutes to complete whole procedure.

# 7. PROJECT JUSTIFICATION

- University is currently having a Manual Time-Table Generation System Which is very time consuming and even faces clashing.
- To overcome this problem and provide the university a good automated timetable generation system this project has been taken.
- The software made i.e. "TIME-TABLE GENERATOR SYSTEM" will be an easy to use and error free system. It will avoid the Clashes and will made time table for all the Classes.
- It is expected that the project will be completed in this semester, but if due to some problems it is not completed in this semester it will be continued in the next semester till it is completed.
- Regarding the cost there will be no expenditure in buying the new software to install the system as the university is already having MSSQL and Windows operating system which will be required to use the project. The required hardware is already available with the university.

# CHAPTER 3- SOFTWARE DESIGNER DESCRIPTION

## 1. INTRODUCTION

# 1.1 Design Overview

Through this project, we are trying to provide a platform to the users to write about anything which could be useful for other bloggers/users. It can be a life hack, an experience that moved you, or regarding academics. Internet is extremely vast. Internet can be a life saviour and annoyingly chaotic at the same time.

# 1.2 UML Diagrams

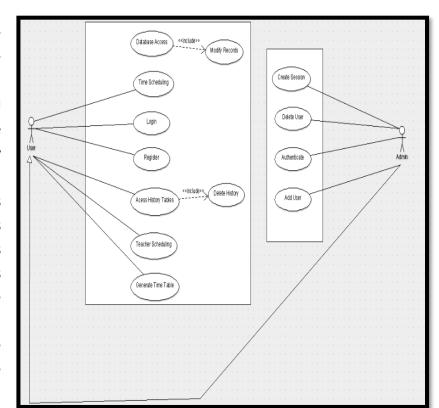
The **Unified Modelling Language** is a standard visual modelling language intended to be used for

- modelling business and similar processes,
- Analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behaviour of artifacts of software systems. UML can be applied to diverse application domains (e.g., banking, finance, internet, aerospace, healthcare, etc.) It can be used with all major object and component software development methods and for various implementation platforms (e.g., J2EE, .NET).

#### 1.2.1 USE-CASE DIAGRAM

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly requirements. design Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view.

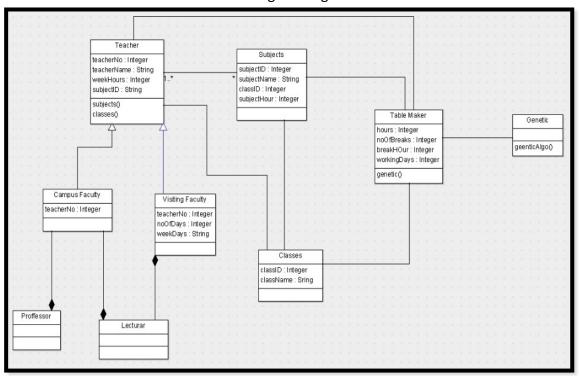


- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

#### 1.2.2 CLASS DIAGRAM

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community. The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.



# 1.3 Database Design:

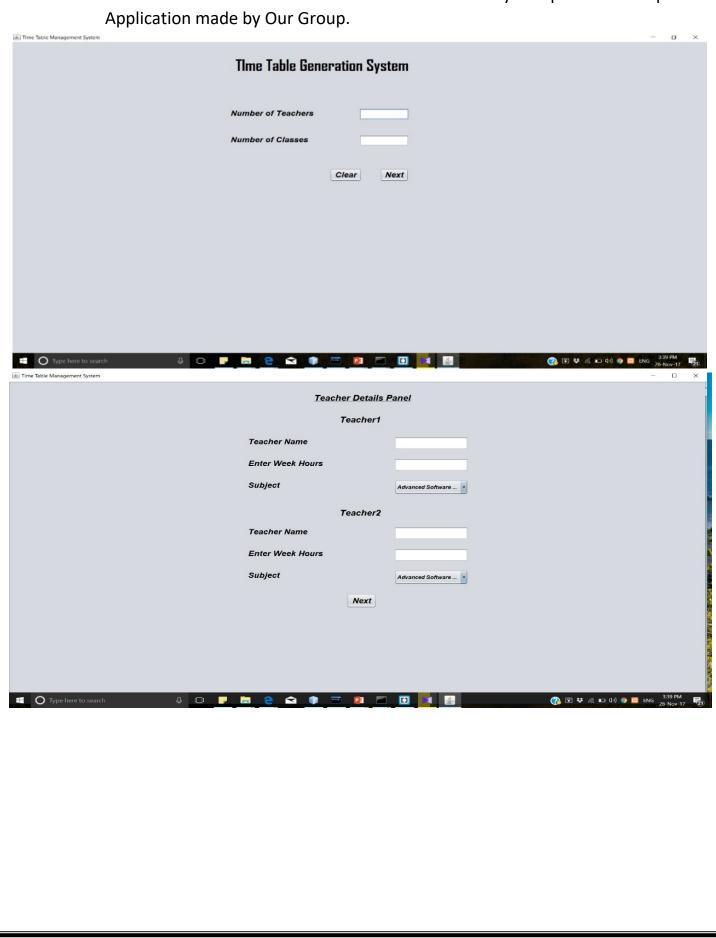
- **❖** TEACHERS
  - TEACHER\_ID
  - TEACHER\_NAME
  - WEEKHOUR
- **❖** SUBJECT
  - SUBJECT\_ID
  - SUBJECT\_NAME
- **❖** TEACHER\_SUBJECT
  - TEACHER\_ID

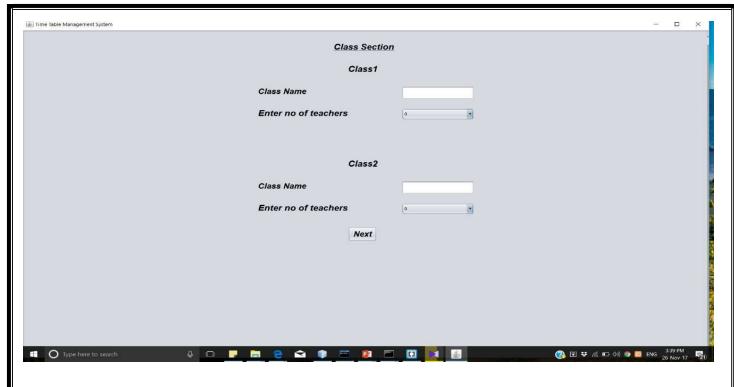
- SUBJECT ID
- CLASSES
  - CLASS ID
  - CLASS\_NAME
- CLASS\_TEACHER
  - CLASS\_ID
  - TEACHER\_ID
  - TEACHER\_HOUR
  - Max\_Hour

# **1.4 USER INTERFACES**

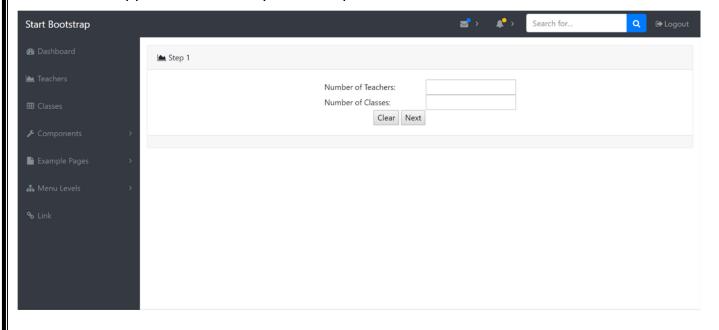
# 1.4.1 Desktop Application:

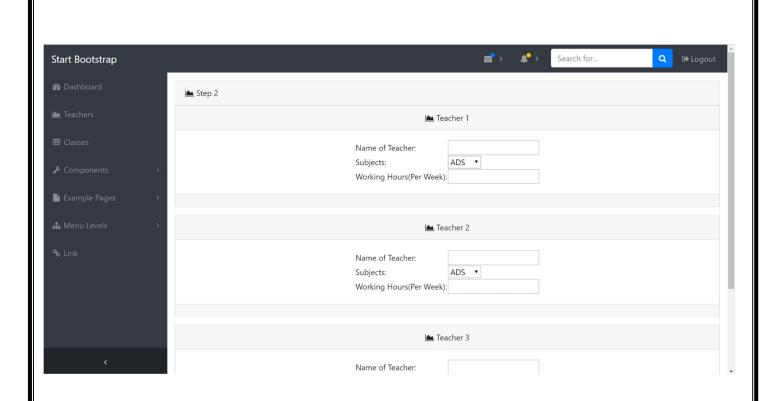
Below Screenshots are the User Interfaces of Partially Completed Desktop

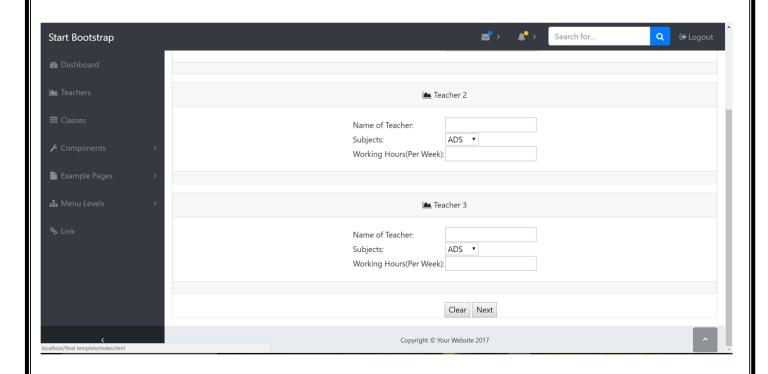


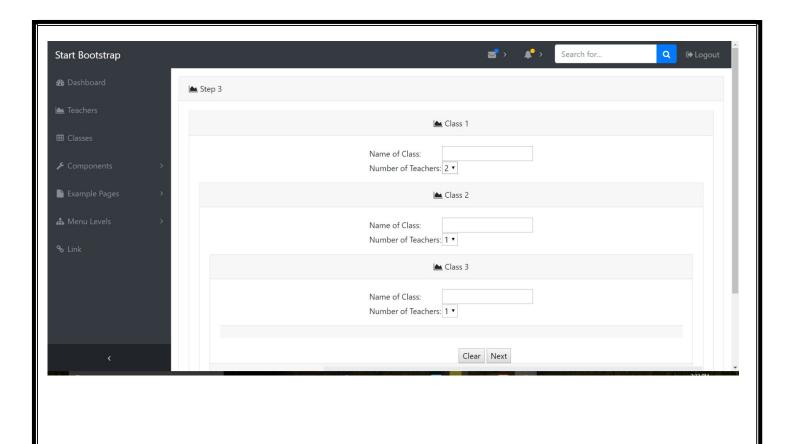


**1.4.2 Dynamic Web Application:** User Interface for Partially Completed Web Application made by our Group.









# CHAPTER 4- SOFTWARE TEST DOCUMENTATION (STD)

# 1. Testing Objectives

The objective for using black box testing is to validate that the application works as the user will be operating it and in the environments of their systems. Black-box testing checks that the user interface and user inputs and outputs all work correctly. Part of this is that error handling must work correctly. It's used in functional and system testing.

# 2. QUALITY DIMENSIONS OF WEB APPLICATION TESTING:

Following are the key quality aspects to keep in mind for testing of web-apps, and test cases are to be framed accordingly.

- Content is evaluated at both a syntactic and semantic level.
  - ◆ Syntactic level—spelling, punctuation and grammar are assessed for text-based documents.
  - ◆ Semantic level—correctness (of information presented), consistency (across the entire content object and related objects) and lack of ambiguity are all assessed.
- ❖ <u>Function</u> is tested for correctness, instability, and general conformance to appropriate implementation standards (e.g., Java or XML language standards).
- ❖ <u>Structure</u> is assessed to ensure that it
  - properly delivers WebApp content and function
  - ♦ is extensible
  - can be supported as new content or functionality is added.
- Usability is tested to ensure that each category of user
  - ♦ is supported by the interface
  - ♦ can learn and apply all required navigation syntax and semantics
- Navigability is tested to ensure that all navigation syntax and semantics are exercised to uncover any navigation errors (e.g., dead links, improper links, erroneous links).
- Performance is tested under a variety of operating conditions, configurations, and loading to ensure that
  - ♦ the system is responsive to user interaction
  - ♦ the system handles extreme loading without unacceptable operational degradation

- ❖ Compatibility is tested by executing the WebApp in a variety of different host configurations on both the client and server sides. The intent is to find errors that are specific to a unique host configuration.
- ❖ Interoperability is tested to ensure that the WebApp properly interfaces with other applications and/or databases.
- Security is tested by assessing potential vulnerabilities and attempting to exploit each.
  Any successful penetration attempt is deemed a security failure.

# 3. POSSIBLE ERRORS IN A WEB-APP DESIGN:

- ♦ Because many types of WebApp tests uncover problems that are first evidenced on the client side, you often see a symptom of the error, not the error itself.
- ◆ Because a WebApp is implemented in a number of different configurations and within different environments, it may be difficult or impossible to reproduce an error outside the environment in which the error was originally encountered.
- ◆ Although some errors are the result of incorrect design or improper HTML (or other programming language) coding, many errors can be traced to the WebApp configuration.
- ♦ Because WebApps reside within a client/server architecture, errors can be difficult to trace across three architectural layers: the client, the server, or the network itself.
- ◆ Some errors are due to the *static operating environment* (i.e., the specific configuration in which testing is conducted), while others are attributable to the dynamic operating environment (i.e., instantaneous resource loading or time-related errors).

# 4. Possible error aspects in our project:

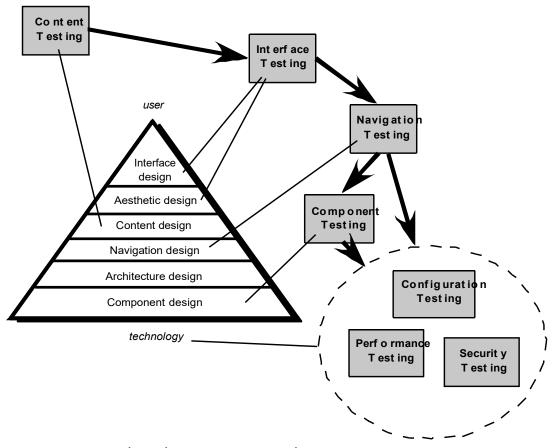
- ◆ The quality of the content (both semantic and syntactic) could be compromised if errors creep in.
- ◆ The login criteria could be violated; i.e, every login must create a session which will expire after a certain time of inactivity. If this fails, this would be a major security breach.
- ◆ Process of going from *index -> teacher\_panel -> class\_panel -> table\_layout\_panel* is a transaction. So all the ACID properties of transaction must be ensured.
- ♦ The navigation constraints should be uniformly implemented.
  - The user cannot go back or refresh the webpage once the generation of timetable process has begun.

- The sequence of the time-table generation process should not be violated or else the generated time table might be faulty.
- ◆ The algorithm (logic) could be faulty or falsely implemented.
- ◆ The number of entities to be generated in the dropdown must be precise and accurate or else the whole generation would be rendered useless.

# 5. WEB-APP TESTING STRATEGIES:

- ◆ The content model for the WebApp is reviewed to uncover errors.
- ♦ The interface model is reviewed to ensure that all use-cases can be accommodated.
- ◆ The design model for the WebApp is reviewed to uncover navigation errors.
- ◆ The user interface is tested to uncover errors in presentation and/or navigation mechanics.
- Selected functional components are unit tested.
- Navigation throughout the architecture is tested.
- ◆ The WebApp is implemented in a variety of different environmental configurations and is tested for compatibility with each configuration.
- ◆ Security tests are conducted in an attempt to exploit vulnerabilities in the WebApp or within its environment.
- Performance tests are conducted.
- ◆ The WebApp is tested by a controlled and monitored population of end-users
- ◆ The results of their interaction with the system are evaluated for content and navigation errors, usability concerns, compatibility concerns, and WebApp reliability and performance.

# 6. THE TESTING PROCESS:

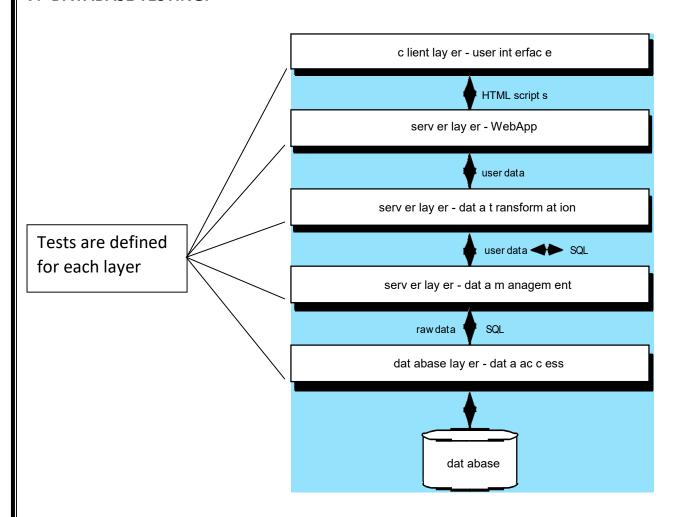


Content testing has three important objectives:

- ◆ To uncover syntactic errors (e.g., typos, grammar mistakes) in text-based documents, graphical representations, and other media
- ◆ To uncover semantic errors (i.e., errors in the accuracy or completeness of information) in any content object presented as navigation occurs, and
- ◆ To find errors in the organization or structure of content that is presented to the enduser.

For content testing, the testing will be done manually by all the team members by thoroughly going through all the content of the web-app.

# 7. DATABASE TESTING:



# **Tool for Database testing:**

# **SQL Test**



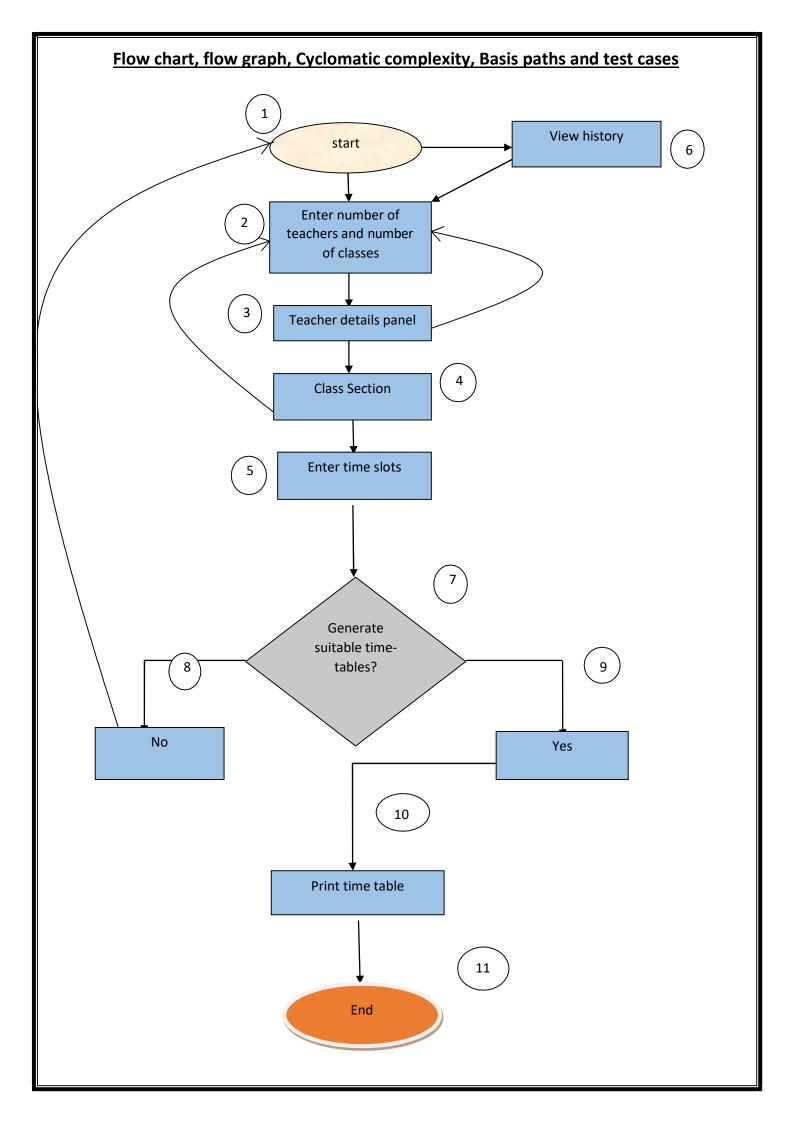
- ◆ SQL Test uses open-source tSQLt framework, views, stored procedures and functions
- This tool stores database object in separate schema and if changes occur there no need for clean-up process
- Allows to run unit tests for SQL Server databases

# 8. USER INTERFACE TESTING:

- ♦ Interface features are tested to ensure that design rules, aesthetics, and related visual content is available for the user without error.
- ◆ Individual interface mechanisms are tested in a manner that is analogous to unit testing.
- ◆ Each interface mechanism is tested within the context of a use-case or NSU for a specific user category.
- ◆ The complete interface is tested against selected use-cases and NSUs to uncover errors in the semantics of the interface.
- ◆ The interface is tested within a variety of environments (e.g., browsers) to ensure that it will be compatible.

# 9. Testing Interface Mechanisms:

- ◆ Links: All the links in the web app will be tested that they are all functional and accurate.
- ♦ Forms: The forms on various webpages of the webapp will be tested by giving sample data and submitting the form.
- ◆ Cookies: We are not using cookies for our Time Table Generation webapp, so there is no need for testing for cookies.
- ◆ Dynamic HTML: testing leads to content objects that are manipulated on the client side using scripting or cascading style sheets (CSS).



NODES: II  
EDGES: 14  

$$V(G) = EDGES - NODES + 2$$
  
Therefore,  
 $V(G) = 13 - 9 + 2 = 5$ 

# **BASIS PATHS**

Since V(G)=6, therefore number of basis paths=5

```
# 1-2-3-4-5-7-8-1

# 1-2-3-4-5-7-9-10-11

# 1-6-2-3-4-5-7-8-1

# 1-6-2-3-4-5-7-9-10-11

# 1-2-3-2-4-5-7-9-10-11
```