Chapter 6

Implementation

6.1 Introduction

Previous chapter provided the design of the proposed solution. This chapter is about how the solution will be implemented in the context of proposed timetable management system. This chapter will further describe about equipment, tools, software and hardware required for implementing processes of the proposed system. Moreover, Algorithm implementation, Database implementation and Interface implementation are been thoroughly discussed in this chapter. Some of the user interfaces are defined as Appendix A

6.2 Software Requirement for Implementation Process

Since the proposed system is web-based system, some of the following open source software were installed and configured before implement the proposed system. This system front end has been developed using HTML, CSS, JavaScript and Bootstrap. Further, for making server application and communicate with database it used server side language PHP. MYSQL serves the front end.

- WAMP Server
- Yii Framework
- Eclipse for PHP Developers

6.2.1 WAMP Server

While WAMP Server is a Windows web development environment, it permits us to create web applications with Apache2, PHP and a MySQL database. Further, it is combined with PHP MyAdmin for the purpose of database management.

Download WAMP from http://www.wampserver.com/ website. Upon our computer configuration, I have chosen 64bit version WAMP.

WAMP Server installs automatically (with an installer). Therefore, we can fine tune our server.

WAMP Server also has a tray icon to manage our server and its settings. If the icon blinks green, server is running well and can log in to the local host. If the icon does not turn to green color and still in red or orange color, check whether the port 80 sharing with Skype. If so, uncheck the Skype port 80 and restart the all the services of the server.

6.2.2 Yii Framework

Yii framework needs PHP 5.1. Therefore, the server must have PHP 5.1 or above versions and available to the web server. Moreover, Yii has been tested with Apache HTTP server on Windows and Linux [25]

1. Downloaded the latest version of Yii
2. Extracted at most contented route
3. Registered the path at Window Environment Variables as C:\Wamp2.4\bin\php\php5.4.16
4. Logged off windows to take effect
5. Open the Windows command prompt and then at the console environment typed "c:\framework\yiic webapp wamp2.4\www\name_of_the_application".

6.2.3 Eclipse for PHP Developers

Eclipse is tools useful for PHP developers creating Web applications.

6.3 Hardware Requirement for Implementation Process

- Computer with 2GB RAM
- 2GHz or more processing power
- Internet connection
- 15" Monitor
- Router
- Keyboard and Mouse

6.4 Implementation of First Level module

6.4.1 Interface Implementation

There are fifty interfaces include in this TMSFIT system. Following classes are associated with some of those interfaces. Those classes are residing in model and controller of the MVC architecture.

- LoginForm, SiteController and User classes used for login interface.
- DashboardController class used for dashboard interface.
- UserController class used all the interfaces with CRUD operators.
- StudentController, Student and Studentenrolsubject classes used for Create Student, Update Student, Delete Student, Manage Student, View Student Timetable,
- BatchController and Batch classes used for Create Batch, Update Batch, Delete Batch, View Timetable and Manage Batch interfaces.
- SubjectController class Subject classes used for View Subject, Create Subject, Delete Subject, Enrole Students, Manage Subject and Assign Lecturer to Subject interfaces.
- ResourceController class used for Create Resource, Update Resource, Delete Resource and View Resource Allocation interfaces.
- EmployeeController, EmployeeSujectController, Employee and EmployeeSubject classes use Create Lecturer, Update Lecturer, Delete lecturer, View Lecturer Timetable, Lecturer Preferred Time and Manage Lecturers interfaces.
- EmployeeSubject class use for Assign Lecturer to Subject interface
- Algorithm class use for Generate timetable and View timetable interfaces

6.4.1.1 Login Interface

- By entering Admin login name and password, admin dashboard interface can be accessed.
- By entering Lecturer login name and password, admin dashboard interface can be accessed.
- By entering Student login name and password, admin dashboard interface can be accessed.

6.4.1.2 Dashboard Interface

Dashboard interface consist of a menu and icons. Further, Dashboard menu contains control panel and logout. Degree, Students, Lecturer, Subjects, Resources, Batch and Timetable icons are seen on the dashboard. See Appendix A

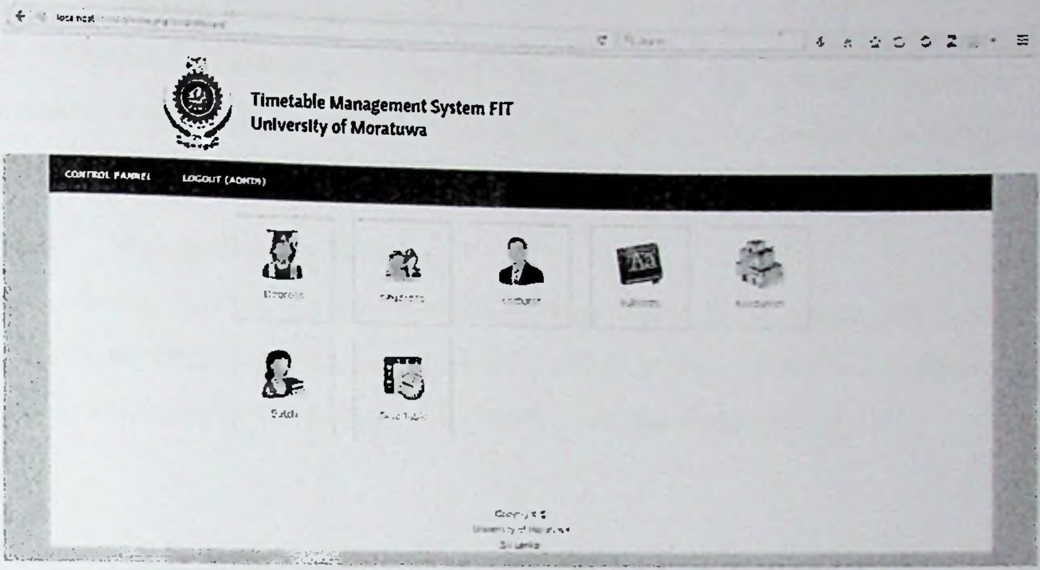


Figure 6-1Dashboard Interface

6.4.1.3 Manage Degrees Interface

This Manage Degrees interface contains Create Degree sub menu and list of degrees. Each degree can be searched, edited and deleted.

6.4.1.4 Manage Students Interface

Manage Students interface includes create and upload sub menus. Moreover, it has a list of students' details and options of searching, updating and deleting.

6.4.1.5 Manage Lecturers Interface

Manage Lecturers interface consist of create lecturer, manage assign subjects and lecturer timetable sub menus. Further, it has a list of lecturers' details and options of searching, updating and deleting.

6.4.1.6 Manage Subjects Interface

Manage Lecturers interface encloses create subject, enroll students and enrolled student sub menus. It also contains a list of subjects' details and options of Assign lecturer to subject, searching, updating and deleting.

6.4.1.7 Manage Resources Interface

Manage Lecturers interface consist of create resource and view time allocation sub menus. It also contains a list of resources details and options searching, updating and deleting.

6.4.1.8 Manage Batches Interface

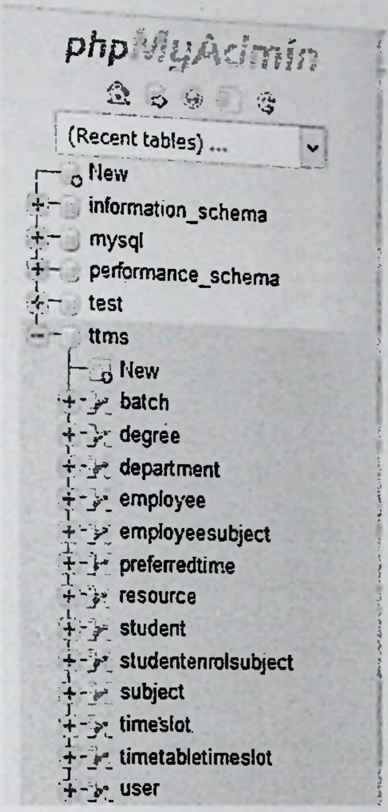
Manage Batches interface contains Manage Batch, Create Batch and Batch Timetable sub menus. It also contains a list of batch details and options searching, updating and deleting. Below figure will show the Manage Batches Interface.

6.4.1.9 Generate Timetable Interface

Generate Timetable interface holds create View Timetable and Generate Timetable sub menus. Further, it contains run algorithm option.

6.4.2 Database Implementation

- MYSQL is used as the back end of this system. Because it includes number of engines and delivers SQL commands to operate database.
- As the first step of developing this system, a proper database was constructed in phpMyAdmin. In this system, it is called ttms as mentioned in the ER diagram.
- InnoDB engine use for foreign keys, support transactions and row level locking.
- According to the ER diagram in Design chapter with main entities such as student, subject, timeslot, lecturer, degree and department main tables were generated. Moreover, batch, degree, department, employee, employeesubject, preferredtime, resource, student, studentenrolsubject, subject, timeslot, timetabletimeslot and user tables also implemented.
- Primary keys are assigned properly to avoid duplicate fields.
- In addition, user also can back up database, import database and export database any time.



phpMyAdmin

(Recent tables) ...

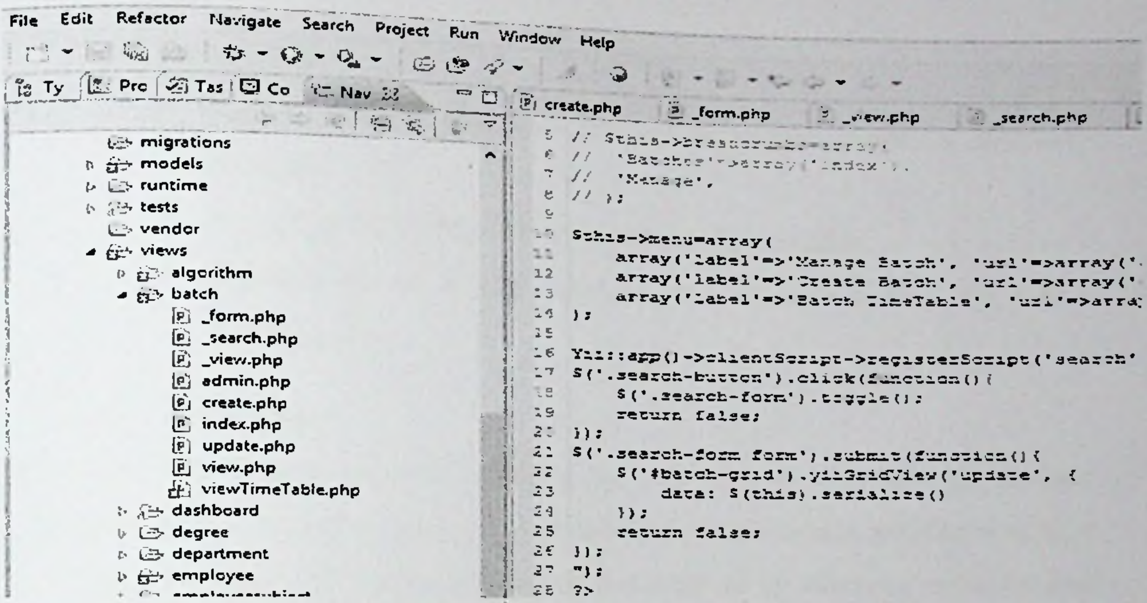
New
information_schema
mysql
performance_schema
test
ttms
New
batch
degree
department
employee
employeesubject
preferredtime
resource
student
studentenrolsubject
subject
timeslot
timetabletimeslot
user

Figure 6-2 Tables of ttms

6.4.3 The Process of Mapping Database Tables with Model class

- By login in to http://localhost/ttmsfit/index.php?r=gii, we come to the Yii code generator.
- Then, we can use the model generator, which generates a model class for the specified database table Eg batch.
- It generates all the appropriate user interfaces mapping the tables of the database.
- Using Curd generator, generate a view script file that displays a form to collect input for the specified model class. Eg batch is under view module
- After creating the appropriate model classes for the tables in the database CURD generator generates a controller and views that implement CRUD operations for the specified data model.

Code regarding _form.php can be referred in Appendix B and Interface according to the _form.php is as below in Figure 6 -3.



```
// $this->breadcrumbs=array(
//   'Batches'=>array('index'),
//   'Manage',
// );

$this->menu=array(
    array('label'=>'Manage Batch', 'url'=>array('
    array('label'=>'Create Batch', 'url'=>array('
    array('label'=>'Batch TimeTable', 'url'=>arra
);

Yii::app()->clientScript->registerScript('search'
$('.search-button').click(function(){
    $('.search-form').toggle();
    return false;
});
$('.search-form form').submit(function(){
    $('#batch-grid').yiiGridView('update', {
        data: $(this).serialize()
    });
    return false;
});
");
?>
```

Figure 6-3 Interface of_form.php

6.5 Second Level Module

Implementation of the logic of the timetable (Algorithm), constraints or rules, verification, timetable generation, view, delete and edit is describe as below.

6.5.1 Algorithm Development

- AlgorithmController class extends the controller class.
- Algorithm class developed under model class by extending CFormModel.
- Code regarding the algorithm implementation can be found in Appendix B

6.5.2 Hard Constraints of TMSFIT

Hard constraints cannot be violated when the generation of the timetable. These hard constraints are reside in the calFiness() and always use to evaluate the fitness value of the timetable schedule. See the Apendix B for fitness function

- Room Overlap – Check if there are two lectures in one room
- Room not enough – No of students of a class is > seats of room
- Required resource not available - Does the lab available the required no of Computers?

- Lecturer Overlap – One lecturer can't be in two rooms at the same time
- Student Overlap - One student can't be in two rooms at the same time

6.5.3 Initialization

- Create an initial population and this population is usually randomly generated.
- Regarding this solution 100 chromosomes must be taken.
- Chromosome or a class schedule must be defined first. Eg public $_chromosomes

6.5.4 Evaluation

The fitness value is calculated by how well it fits with our desired requirements. Evolution's goal is to find better individuals in each generation using a fitness function. The process of evolution is maintained by selection, crossover and mutation. Below code segment will illustrate the chromosome construction.

```
FunctioninitObject($numberOfCrossoverPoints,$mutationSize,$crossoverProbability,
$mutationProbability,$fitness,$subjectClass)
    {
        // reserve space for time-space slots in chromosomes code
        $this->_slots      =      new      SplFixedArray(Schedule::DAYS_NUM      *
    Schedule::DAY_HOURS * count(Resource::model()->findAll()));
        $this->_criteria = new SplFixedArray( 5 * count($subjectClass));
        $this->_mutationSize = $mutationSize;
        $this->_numberOfCrossoverPoints = $numberOfCrossoverPoints;
        $this->_crossoverProbability = $crossoverProbability;
        $this->_mutationProbability = $mutationProbability;
        $this->_fitness = $fitness;
        $this->_subjectClass = $subjectClass;
        $labs = Resource::model()->findAll("type = 'lab'");
        $lHalls = Resource::model()->findAll("type != 'lab'");
        $this->_allClassRoom = array_merge($lHalls,$labs);
        $this->_noOfClassRoom = count($this->_allClassRoom);
        $this->_noOfLabs = count($labs);
        $this->_noOfHalls = count($lHalls);
    }
```

Class table for chromosome and used to determine first time-space slot used by class

```
    public $_classes, public $_subjectClass, public $_noOfClassRoom,    public
    $_noOfLabs,        public $_noOfHalls,  public $_allClassRoom;
```

Example:

Each hour of a class has a separate entry in the array, but there is only one entry per class in the class index map. For instance, if a class starts at 1pm and lasts for three hours, it has entries in the 8.15am, 9.15am, and 10.30 slots. Below Table 6-1 will illustrate the Chrimosome

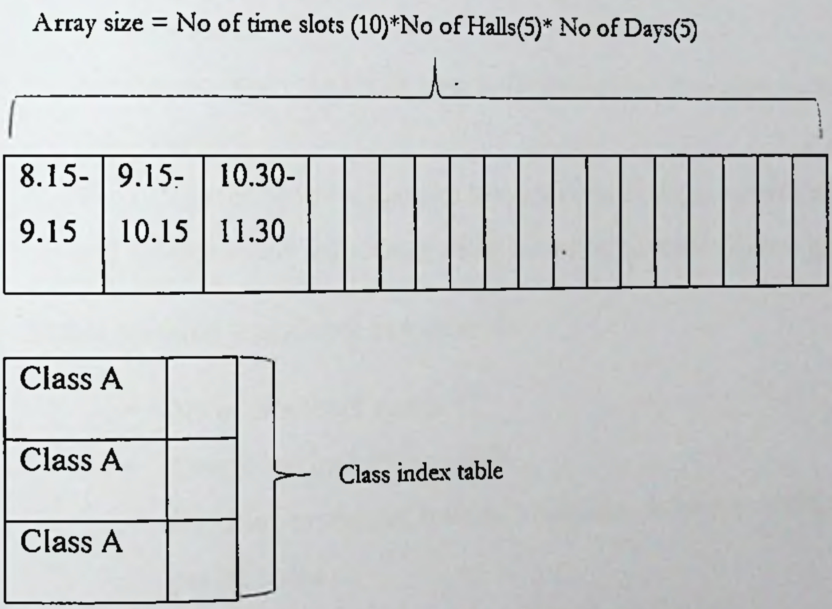


Table 6-1 Chromosome Representation

6.5.5 Evaluation

The fitness value is calculated by how well it fits with our desired requirements. Evolution's goal is to find better individuals in each generation using a fitness function. The process of evolution is maintained by selection, crossover and mutation. Below diagram, illustrates

6.5.6 Selection

- Selection chooses superior individuals in every generation
- We want to be constantly improving our populations overall fitness.
- Selection helps us to do this by discarding the bad designs and only keeping the best individuals in the population

In this research, find the Apendix B for calFitness function code segment

- Use 100 Chromosomes as this pool(gene)
- Calculate fintess value for chromosome by using calFitness()
- calFitness() uses the hard constraints and increase the score one by one 0 to 5
- Then select the most suitable 5 chromosomes as timetable schedules.

6.5.7 Crossover

- During crossover we create new individuals by combining aspects of our selected individuals.
- The crossover operator chooses two individuals from current population (parents) and creates a new individual (child) based on parents' genetic material.

In this research it describes as Figure -14

- No of crossover points =2
- Crossover probability =80%
- By using crossover($parent2) function, make new offspring by combining parent codes
- Then check the fitness again using the calFitness()
- If found a fitter chromosome than a previous selected one change it to new one

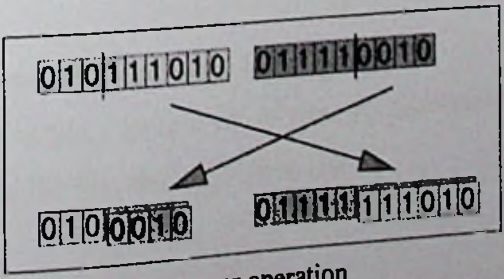


Figure 6-4 Crossover operation

6.5.8 Mutation

- Mutation typically works by making very small changes at random to an individual's genome.
- The mutation operator changes the value of some genes in an individual and helps to search other parts of problem space

With regards to this solution, see the Apendix B for mutation code segment.

- Mutation probability = 3
- By using mutation() generate new chromosomes
- Then again check their fitness with previous 5 best chromosomes using calFitness()
- If found a fitter chromosome than a previous selected one change it to new one

After these three steps selection, crossover, mutation, 5 best chromosomes (timetable schedules) have been generated.

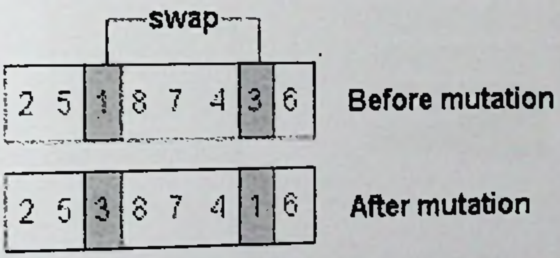


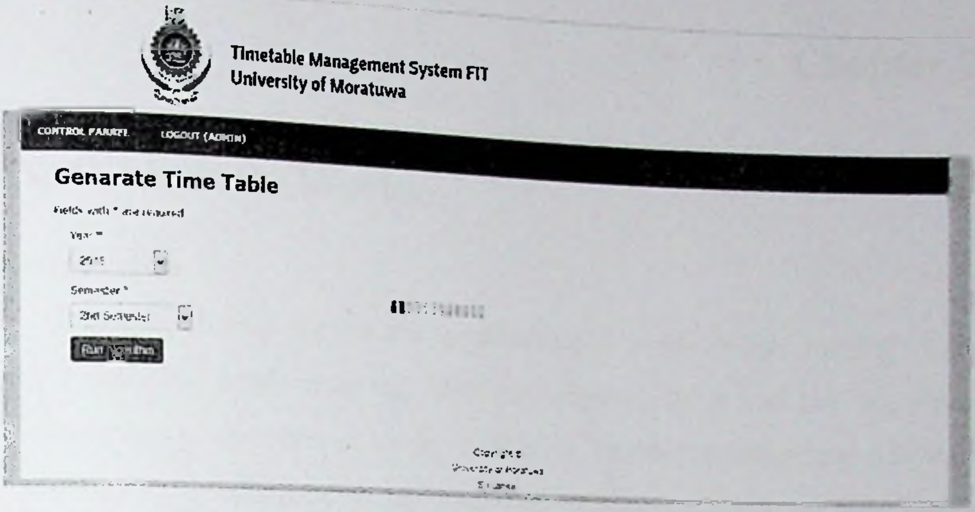
Figure 6-5 Mutation operation

6.6 Third Level Module

This includes generated timetable, view the timetable on the web and print the timetable.

6.6.1 Generate Timetable Interface

While it takes quite a lot of time to generate a timetable, use a progress bar as waiting facility. See the Figure 6-6



Timetable Management System FIT
University of Moratuwa

Genarate Time Table

Fields with * are required

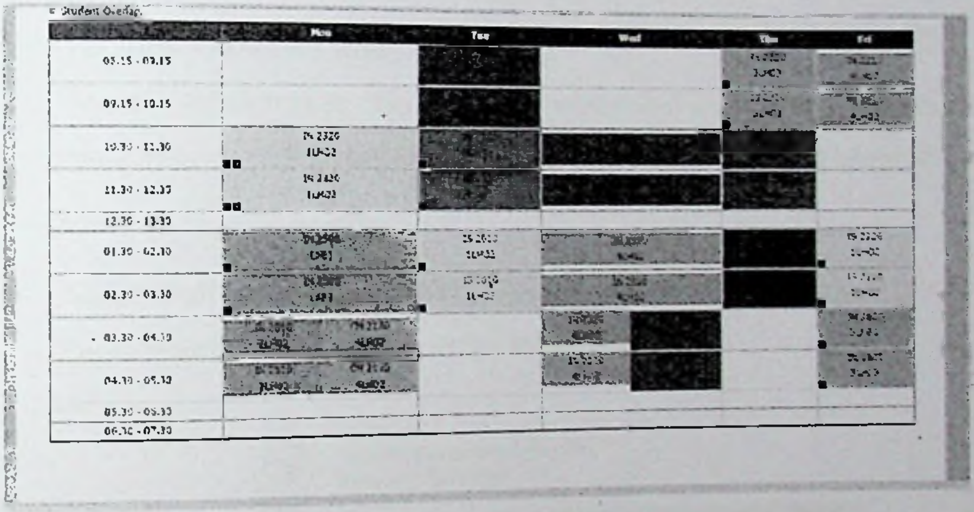
Year *

2015

Semester *

2nd Semester

Figure 6-6 Timetable Generating State



Student Overlap

	Mon	Tue	Wed	Thu	Fri
08.15 - 09.15		[illegible]		[illegible]	[illegible]
09.15 - 10.15		[illegible]		[illegible]	[illegible]
10.30 - 11.30	IN 2320 10-02	[illegible]	[illegible]	[illegible]	
11.30 - 12.30	IN 2320 10-02	[illegible]	[illegible]	[illegible]	
12.30 - 13.30					
01.30 - 02.30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
02.30 - 03.30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
03.30 - 04.30	[illegible]		[illegible]		[illegible]
04.30 - 05.30	[illegible]		[illegible]		[illegible]
05.30 - 06.30					
06.30 - 07.30					

Figure 6-7 Generated Timetable

6.7 System Deployment

As agreed earlier with timetable administrator, parallel deployment was conducted. As a result, both previous system and new system work together. If the new system is better than the previous one and success, then, they can smoothly shift to the new system.

6.8 Summery

This chapter was about implementation details such as implementation of software, hardware, and algorithm, modules of the top-level design architecture, main system user interfaces, sample codes and some more.