

Conclusion and Further Work

9.1 Introduction

Utilization of Timetable Management System to Faculty of IT is successfully achieved throughout this research and accomplished tasks by the TMSFIT as follows.

- Generation of the timetable management system
- Resource optimization
- View the resources available

9.2 Conclusion

Utilization of Timetable Management System to Faculty of IT at University of Moratuwa is successfully achieved through this research process. From a thorough study of literature survey a proper algorithm could find as Genetic Algorithm. According to Prof Kaunananda, Genetic Algorithm can be applied to problems which are unsolvable [19]. The proposed timetabling system for this project seeks to generate near optimal timetables using the principles of genetic algorithm (selection, mutation and crossover). It is easily understandable, efficient and automated system, which is helpful for authorities of the IT faculty, lecturers and students. In addition, it is less paper work.

I have pleased to state; that I could get experience in Yii framework and the MVC design patterns. As a result, now I can develop any Yii component by my own. By following, this research my patience quality increased. For example, I spent number of days to find a searching algorithm. Writing about literature review and find the theories of that. Find information about research methodology and writing of interim document also interesting. There, I could learn about number of algorithms, about knowledgebase systems, searching methods, technologies used by others and Genetic algorithm and Programming languages such as PHP and DBMS such as MYSQL.

As conclusion, the generating timetables and viewing available resources tasks are successfully accomplished by the system within five minutes.

9.3 Further Work

Even this solution is ideal for a middle-sized campus as Moratuwa, large universities, cannot get the best results due to complexity of their problem in case of fitness value is getting decrease. Therefore, I propose those universities can use a combination of ruled base and Genetic Algorithm solution to develop such kind of system.

9.4 Summery

This chapter provided a conclusion of overall achievement met through the research project called utilization of timetable management system to faculty of IT at University of Moratuwa and further work in future as an enhancement of the current solution.

References

- [1] E. Burke and P. De Causmaecker, Eds., *Practice and theory of automated timetabling IV: 4th international conference, PATAT 2002, Gent, Belgium, August 21-23, 2002: selected revised papers*. Berlin ; New York: Springer, 2003.
- [2] "timetable Meaning in the Cambridge English Dictionary." [Online]. Available: <http://dictionary.cambridge.org/dictionary/english/timetable>. [Accessed: 10-Mar-2016].
- [3] "Mimosa - Scheduling Software for School and University Timetables." [Online]. Available: <http://www.mimosasoftware.com/>. [Accessed: 08-Mar-2016].
- [4] L. Carpenente, A. Cerdeira-Pena, G. de Bernardo, and D. Seco, "An Integrated System for School Timetabling.," in *ICAART (1)*, 2011, pp. 599–603.
- [5] University of Moratuwa, Faculty of Information Technology - Student Handbook, 2013th ed.
- [6] N. Pillay, "A survey of school timetabling research," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 261–293, Jul. 2014.
- [7] A. Dammak, A. Elloumi, H. Kamoun, and J. A. Ferland, "Course timetabling at a Tunisian University: A case study," *J. Syst. Sci. Syst. Eng.*, vol. 17, no. 3, pp. 334–352, Sep. 2008.
- [8] H. Cambazard, F. Demazeau, N. Jussien, and P. David, "Interactively solving school timetabling problems using extensions of constraint programming." in *Practice and Theory of Automated Timetabling V*, Springer, 2004, pp. 190–207.
- [9] S. Hooshmand, M. Behshameh, and O. Hamidi, "A Tabu Search Algorithm With Efficient Diversification Strategy for High School Timetabling Problem," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 4, pp. 21–34, Aug. 2013.
- [10] E. K. Burke and S. Petrovic, "Recent research directions in automated timetabling," *Eur. J. Oper. Res.*, vol. 140, no. 2, pp. 266–280, 2002.
- [11] E. K. Burke, D. G. Elliman, and R. Weare, "A university timetabling system based on graph colouring and constraint manipulation," *J. Res. Comput. Educ.*, vol. 27, no. 1, pp. 1–18, 1994.
- [12] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee, "A survey of search methodologies and automated system development for examination timetabling," *J. Sched.*, vol. 12, no. 1, pp. 55–89, Oct. 2008.
- [13] A. Colomi, M. Dorigo, and V. Maniezzo, "A genetic algorithm to solve the timetable problem," *Politec. Milano Milan Italy TR*, pp. 90–060, 1992.
- [14] C.-H. Chen, T.-K. Liu, and J.-H. Chou, "A Novel Crowding Genetic Algorithm and Its Applications to Manufacturing Robots," *IEEE Trans. Ind. Inform.*, vol. 10, no. 3, pp. 1705–1716, Aug. 2014.
- [15] J. J. Moreira, "A system for automatic construction of Exam Timetable using Genetic Algorithms," *Rev. Estud. Politécnicos Polytech. Stud. Rev.*, vol. 6, no. 9, 2008.

- [16] O. I. Obaid, M. Ahmad, S. A. Mostafa, and M. A. Mohammed. "Comparing performance of genetic algorithm with varying crossover in solving examination timetabling problem," *J Emerg Trends Comput Inf Sci*, vol. 3, pp. 1427–1434, 2012.
- [17] E. Cantú-Paz, *Efficient and accurate parallel genetic algorithms*. Boston, Mass: Kluwer Academic Publishers, 2000.
- [18] B. Sigl, M. Golub, and V. Mornar, "Solving timetable scheduling problem using genetic algorithms," in *Proc. of the 25th int. conf. on information technology interfaces*, 2003, pp. 519–524.
- [19] Professor Ashoka Karunananda Bsc., MPhil, PhD, *Artificial Intelligence*, 2004.05 ed. Tharanji Prints, Highlevel Road, Nawinna, Maharagama, 2004.
- [20] "Free trial : timetable software ... for timetabling school timetables ... for easier scheduling." [Online]. Available: <http://timetabler.com/>. [Accessed: 19-Apr-2016].
- [21] "Open Course Timetabler 0.8.1 - Free download." [Online]. Available: <http://open-course-timetabler.soft112.com/>. [Accessed: 19-Apr-2016].
- [22] "About Yii | Yii PHP Framework." [Online]. Available: <http://www.yiiframework.com/about/>. [Accessed: 12-Mar-2016].
- [23] "Take The Tour | Yii PHP Framework." [Online]. Available: <http://www.yiiframework.com/tour/#>. [Accessed: 12-Mar-2016].
- [24] "Fundamentals: Model-View-Controller (MVC) | The Definitive Guide to Yii | Yii PHP Framework." [Online]. Available: <http://www.yiiframework.com/doc/guide/1.1/en/basics.mvc>. [Accessed: 12-Mar-2016].
- [25] "Getting Started: Installation | The Definitive Guide to Yii | Yii PHP Framework." [Online]. Available: <http://www.yiiframework.com/doc/guide/1.1/en/quickstart.installation#requirements>. [Accessed: 13-Mar-2016].

Interfaces of the TMSFIT

Before saving the Timetable which the fitness value = 1

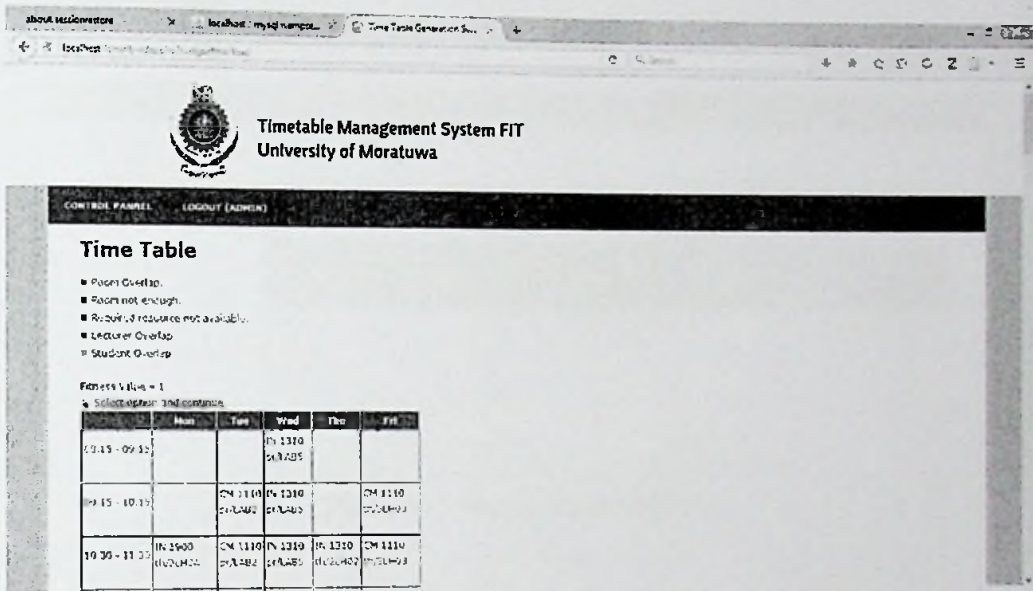


Figure A-1 Generated Timetable Before Saved

Error Messages given by the TMSFIT

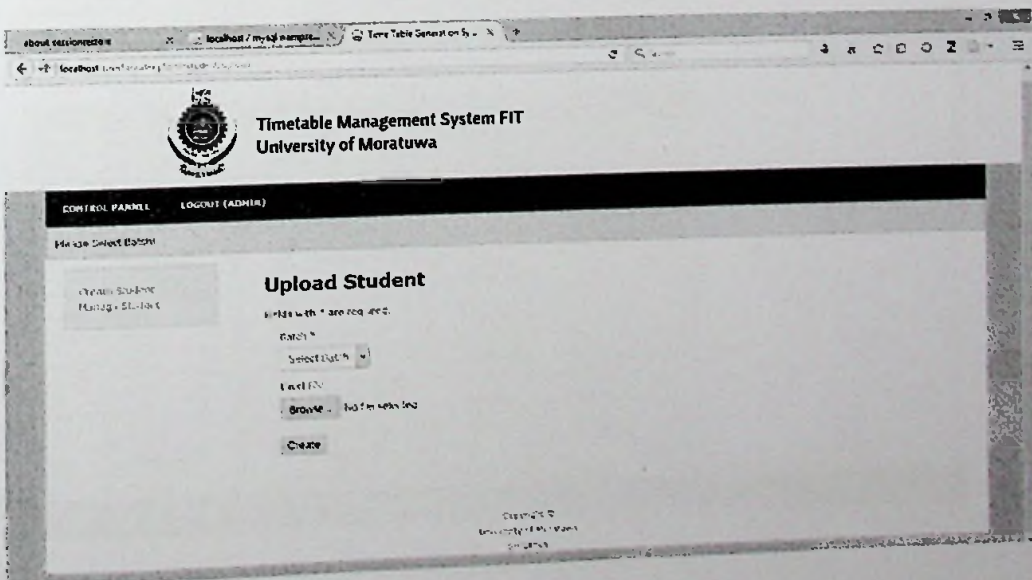


Figure A-2 Error Message

Manage Students Interface

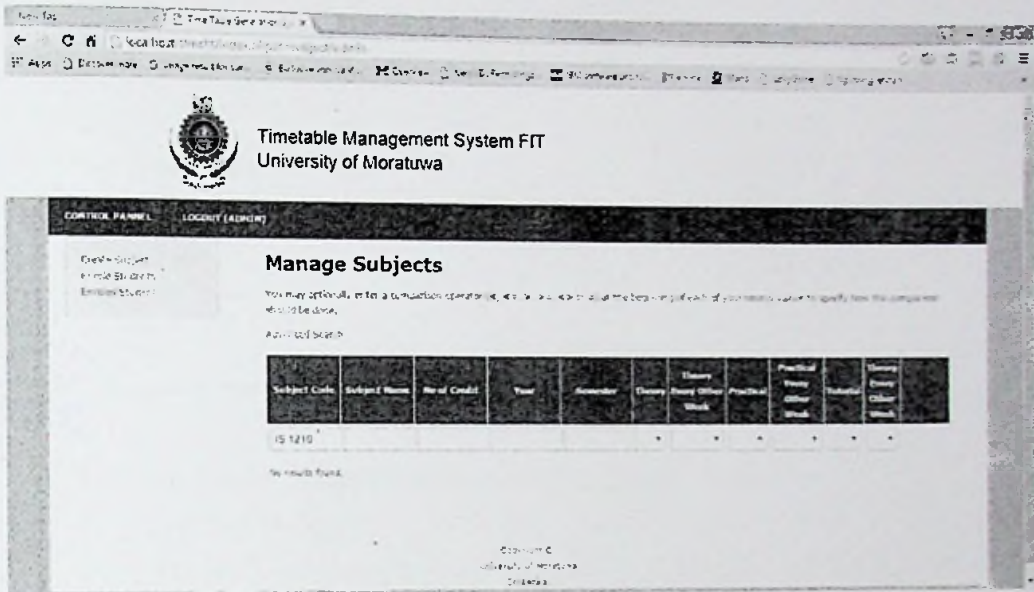


Figure A- 3 Manage Student Interface

Preview of tables

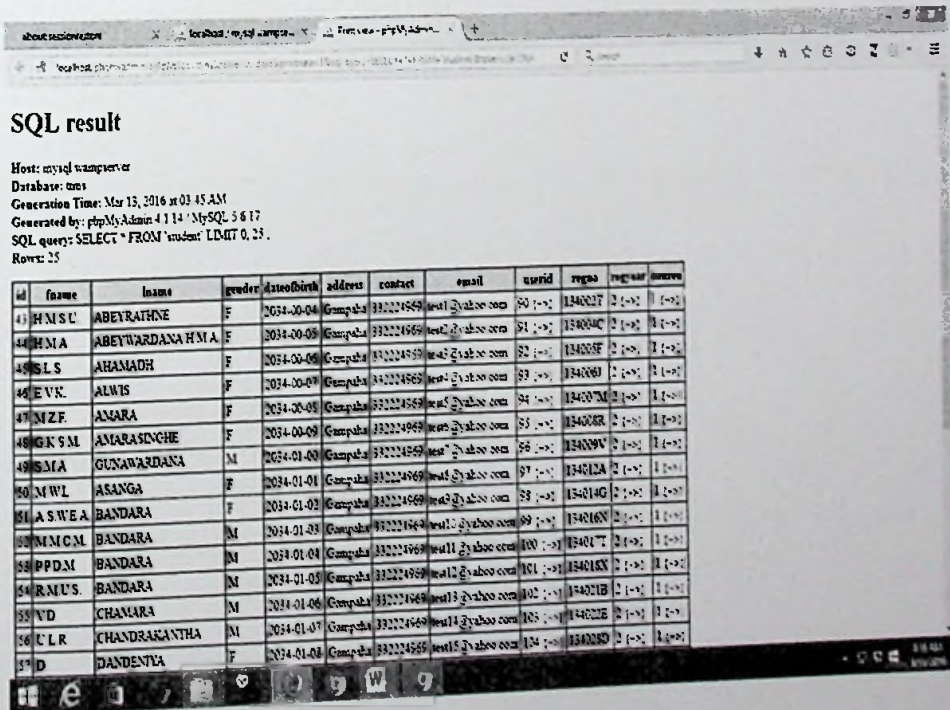


Figure A-4 Table Preview

Code segments of the TMSFIT

Sample code for batch form

```

<?php
/* @var $this BatchController */
/* @var $model Batch */
/* @var $form CActiveForm */
?>

<div class="form">

<?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'batch-form',
    // Please note: When you enable ajax validation, make sure the
corresponding
    // controller action is handling ajax validation correctly.
    // There is a call to performAjaxValidation() commented in
generated controller code.
    // See class documentation of CActiveForm for details on this.
    'enableAjaxValidation'=>false,
)); ?>

    <p class="note">Fields with <span class="required">*</span> are
required.</p>

    <?php echo $form->errorSummary($model); ?>

    <div class="row">
        <?php echo $form->labelEx($model, 'name'); ?>
        <?php echo $form-
>textField($model, 'name', array('size'=>50, 'maxlength'=>50)); ?>
        <?php echo $form->error($model, 'name'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model, 'description'); ?>

```

```

        <?php echo $form-
>textField($model, 'description', array('size'=>60, 'maxlength'=>200));
?>

        <?php echo $form->error($model, 'description'); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($model, 'startyear'); ?>
        <?php echo $form-
>textField($model, 'startyear', array('size'=>4, 'maxlength'=>4)); ?>
        <?php echo $form->error($model, 'startyear'); ?>
    </div>

    <div class="row buttons">
        <?php echo CHtml::submitButton($model->isNewRecord ?
'Create' : 'Save', array('class'=>'btn btn-primary')); ?>
    </div>

<?php $this->endWidget(); ?>

</div><!-- form -->

```

Calculate the fitness of the chromosome

```

    //Calculate fitness value for chromosome
public function calFitness(){
    // chromosome's score
    $score = 0;

    $numberOfRooms = $this->_noOfClassRoom;
    $roomDaySize = Schedule::DAY_HOURS * $numberOfRooms;

    $sci = 0;
    // check criterias and calculate scores for each class in
    schedule
    foreach($this->_classes as $value)
    {
        // coordinate of time-space slot
        $p = $value[0]; //slot number
        $day = (int)($p / $roomDaySize);
    }
}

```



```

$time = $p % $roomDaySize;
$room = (int)($time / Schedule::DAY_HOURS);
$time = $time % Schedule::DAY_HOURS;
//
echo "<br>rooms <del></del>". $room." // slot = ". $p ;
$dur = round($value[1]->duration, 0, PHP_ROUND_HALF_UP);
// check for room overlapping of classes
$ro = false;
for( $i = $dur - 1; $i >= 0; $i-- )
{
    if( array_key_exists ((int)($p + $i), $this->_slots)
    && sizeof($this->_slots[ (int)($p + $i) ]) > 1 )
    {
        $ro = true;
        break;
    }
}
// on room overlapping
if( !$ro ){
    $score++;
    //
    echo "<br>no room overlapping";
}
$this->_criteria[ (int)($ci + 0) ] = !$ro;
$cc = $value[1];
$r = $this->_allClassRoom[$room];
// does current room have enough seats
$roomEnough = false;
if($r->capacity >= count($cc->students)){
    $roomEnough = true;
}
$this->_criteria[ (int)($ci + 1) ] = $roomEnough;
if($roomEnough ){
    $score++;
    //
    echo "<br>Room enough";
}

// does current room have computers if they are required
$labRequired = false;
if($cc->reqLab){
    $labRequired = true;
}
$resourceType = false;

```

```

//      echo "<br>labRequired ".$labRequired ;
//      echo "<br>type>>> ".$r->type ;
      if( $labRequired ){

          if($r->type == 'lab'){
              $score++;
              $resourceType = true;
//          }
//          }

          }else{
              if($r->type != 'lab'){
                  $score++;
                  $resourceType = true;
//              }
//              }

          }else{
//              }
//              }

          }

      $this->_criteria[(int) ($ci + 2) ] = $resourceType;
      set_time_limit(20);
      $po = false;
      $go = false;
//      check overlapping of classes for professors and student groups
      for( $ii = $numberOfRooms, $t = $day * $roomDaySize +
$time; $ii > 0; $ii--, $t += Schedule::DAY_HOURS )
      {
//      for each hour of class
          for( $i = $dur - 1; $i >= 0; $i-- )
          {
//      check for overlapping with other classes at same time
              if (array_key_exists((int)($t + $i), $this->_slots) && isset($this->_slots[(int)($t + $i)]) {
//echo "spk"-----
>". ($t + $i);
                  $classes = $this->_slots[ $t + $i ];
//              }
//              }

          }

      foreach ($classes as $cls){

```

```

                                if( $cc != $cls )
                                {
// professor overlaps?
if( !$po && $cc->lecturerOverlaps( $cls ) )
                                $po = true;

// student group overlaps?
if( !$go && $cc->studentOverlap( $cls ) )
                                $go = true;

// both type of overlapping? no need to check more
                                if( $po && $go )
                                goto total overlap;
                                }
                                }
                                }
}

```

total_overlap:

```

// professors have no overlapping classes?
if( !$po ){
$lectpref = $cc->checkLectPrefTime($cc->lecturer-
>id,$day,$time,$dur,$cc->semester,$cc->year);
if($lectpref){
    $score++;
}else {
    $po = true;
}
}

```

```

$this->_criteria[ $ci + 3 ] = !$po;

```

```

// student groups has no overlapping classes?
if( !$go ){
    $score++;
    echo "<br>Student overlap";
}

```

```

        $this->_criteria[ $ci + 4 ] = !$go;

        $ci += 5;

    //

    }

    // calculate fitness value based on score
    $this->_fitness = $score / ( count($this->_subjectClass) *
Schedule::DAYS_NUM );

}

```

Cross over operation

```

// Performes crossover operation using 2 chromosomes and returns 2
offspring

public function crossover($parent2){
    // check probability of crossover operation
    if( rand() % 100 > $this->_crossoverProbability ){
        // no crossover, just copy first parent
        return $this->makeCopy(false );
    }

    //echo

    "<br><br>crossover=====
===<br>";
    // new chromosome object, copy chromosome setup
    $n = $this->makeCopy(true );

    // number of classes
    $size = (int)count($this->_classes);

    $cp = array();

    // determine crossover point (randomly)
    for( $i = $this->_numberOfCrossoverPoints; $i > 0; $i-- )
        {
            while( true )
            {
                $p = rand() % $size;

```

```

        if(!array_key_exists($p, $cp))
        {
            $cp[ $p ] = true;
            break;
        }
    }
}

$it1 = $this->_classes;
$it2 = $parent2->_classes;
$pkindex1 = 0;
$pkindex2 = 0;
// make new code by combining parent codes
$first = rand() % 2 == 0;
for( $i = 0; $i < $size; $i++ )
{
    if( $first )
    {
// insert class from first parent into new chromosome's class table
        $n->_classes[$pkindex1] =
array($it1[$pkindex1][0],$it1[$pkindex1][1]);

// all time-space slots of class are copied
for( $ii = round($it1[$pkindex1][1]->duration, 0, PHP_ROUND_HALF_UP)
- 1; $ii >= 0; $ii-- ){

if($n->_slots[(int)($it1[$pkindex1][0] + $ii )] == null){
//if array is null then initialize new array and add class
        $n->_slots[(int)($it1[$pkindex1][0] + $ii )] =
array($it1[$pkindex1][1]);
    }else{
        $array = $n->_slots[(int)($it1[$pkindex1][0] + $ii )];
        $array[] = $it1[$pkindex1][1];
        $n->_slots[(int)($it1[$pkindex1][0] + $ii )] = $array;
    }
}
    }
}
else
{
// insert class from second parent into new chromosome's class table

```

```

$n->_classes[$pkindex1] =
array($it2[$pkindex2][0],$it2[$pkindex2][1]);

// all time-space slots of class are copied
for( $ii = round($it2[$pkindex2][1]->duration, 0, PHP_ROUND_HALF_UP)
- 1; $ii >= 0; $ii-- ){
if($n->_slots[(($it2[$pkindex2][0] + $ii )] == null){
//if array is null then initialize new array and add class
$n->_slots[(($it2[$pkindex2][0] + $ii )] = array($it2[$pkindex2][1]);
}else{
$array = $n->_slots[(($it2[$pkindex2][0] + $ii )];
$array[] = $it2[$pkindex2][1];
$n->_slots[(($it2[$pkindex2][0] + $ii )] = $array;
}
}
}

// crossover point
if( array_key_exists($i, $cp)){
// change source chromosome
    $first = !$first;
}

    $pkindex1++;
    $pkindex2++;

}
    $n->calFitness();
//
    echo
"<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<br>";
    return $n;

```

Mutation code segment

```

// Performs mutation on chromosome
    public function mutation()
    {
//echo "<br>.....mutation.....";
// check probability of mutation operation
if( rand() % 100 > $this->_mutationProbability )
    return;

```

```

// number of classes
$numberOfClasses = (int)count($this->_classes);
// number of time-space slots
$size = (int)count($this->_slots);
// move selected number of classes at random position
for( $ii = $this->_mutationSize; $ii > 0; $ii-- )
    {
// select random chromosome for movement

$mpos = rand() % $numberOfClasses;
$pos1 = 0;
$classeIndex = array_keys($this->_classes);

// current time-space slot used by class
$pos1 = $this->_classes[$mpos][0];

$scl = $this->_classes[$mpos][1];

// determine position of class randomly
$nr = $this->_noOfClassRoom;
$dur = round($scl->duration, 0, PHP_ROUND_HALF_UP);
        $day = rand() % Schedule:::DAYS_NUM;
        $room = rand() % $nr;
        $time = rand() % ( Schedule:::DAY_HOURS + 1 - $dur );
$pos2 = $day * $nr * Schedule:::DAY_HOURS + $room *
Schedule:::DAY_HOURS + $time;
// move all time-space slots
    for( $i = $dur - 1; $i >= 0; $i-- )
        {
// remove class hour from current time-space slot
            $cl = $this->_slots[ $pos1 + $i ];
            foreach ( $cl as $key=>$cls ){
                if( $cls == $scl )
                    {
                        unset($cl[$key]);
                        break;
                    }
            }
        }

// move class hour to new time-space slot
if($this->_slots[(int)($pos2 + $i) ] == null){
//if array is null then initialize new array and add class

```

```

$this->_slots[(int)($pos2 + $i) ] = array($scl);

        }else{
$array = $this->_slots[(int)($pos2 + $i) ];
        $array[] = $scl;
        $this->_slots[(int)($pos2 + $i) ] = $array;

        }

    }

}

// change entry of class table to point to new time-space slots
        unset($this->_classes[$mpos]);
        $this->_classes[ $mpos ] = array($pos2,$scl);
    }

    $this->calFitness();
}
}

```

Dashboard code segment

```

<?php
/* @var $this DashboardController */

// $this->breadcrumbs=array(
//     'Dashboard',
// );
?>
<div class="dashboard">
    <div class="dashboard-wrapper">

        <!-- Dashboard for Administrators -->
        <?php if (Yii::app()->user->getState('roles') ==
"admin"): ?>
        <!--
            <div class="dash-icon">
                <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('candidate/admin'); ?>">
                    

```



```
        <div class="link-text">New Applicants</div>
    </a>
</div> -->
```

```
    <div class="dash-icon">
        <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('degree/admin'); ?>">
            
            <div class="link-text">Degrees</div>
        </a>
    </div>
```

```
    <div class="dash-icon">
        <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('student/admin'); ?>">
            
            <div class="link-text">Students</div>
        </a>
    </div>
```

```
    <div class="dash-icon">
        <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('employee/admin'); ?>">
            
            <div class="link-text">Lecturer</div>
        </a>
    </div>
```

```
    <div class="dash-icon">
        <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('subject/admin'); ?>">
            
            <div class="link-text">Subjects</div>
        </a>
    </div>
```

```

        <div class="dash-icon">
            <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('resource/admin'); ?>">
                
                <div class="link-text">Resources</div>
            </a>
        </div>

        <div class="dash-icon">
            <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('batch/admin'); ?>">
                
                <div class="link-text">Batch</div>
            </a>
        </div>

        <div class="dash-icon">
            <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('algorithm/load'); ?>">
                
                <div class="link-text">Time Table</div>
            </a>
        </div>

<?php endif; ?>
<!-- Lecturer Dashboard -->
<?php if (Yii::app()->user->getState('roles') ==
"lecturer"): ?>
    <div class="dash-icon">
        <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('employee/time', array('id'=>Yii::app()->user-
>getState('loggeduserid'))); ?>">
            
            <div class="link-text">Preferred Time</div>
        </a>
    </div>

```

```

<!-- <div class="dash-icon"> -->
<!-- <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('employee/timetable'); ?>">
    
<!-- <div class="link-text">Time Table</div> -->
<!-- </a> -->
<!-- </div> -->

<div class="dash-icon">
    <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('employee/view', array('id'=> Yii::app()->user-
>getState('loggeduserid'))); ?>">
        
        <div class="link-text">Profile</div>
    </a>
</div>

<div class="dash-icon">
    <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('employee/timetable'); ?>">
        
        <div class="link-text">Time Table</div>
    </a>
</div>

<?php endif; ?>

<!-- Dashboard for Students -->
<?php if (Yii::app()->user->getState('roles') ==
"student"): ?>
<div class="dash-icon"><a class="dash-link" href="<?php echo
Yii::app()->createUrl('student/view', array('id'=> Yii::app()->user-
>getState('loggeduserid'))); ?>"> 
    <div class="link-text">Profile</div>

```

```
        </a>
    </div>

    <div class="dash-icon">
        <a class="dash-link" href="<?php echo Yii::app()-
>createUrl('student/timetable'); ?>">
            
            <div class="link-text">Time Table</div>
        </a>
    </div>

    <?php endif; ?>
</div>
</div>
```

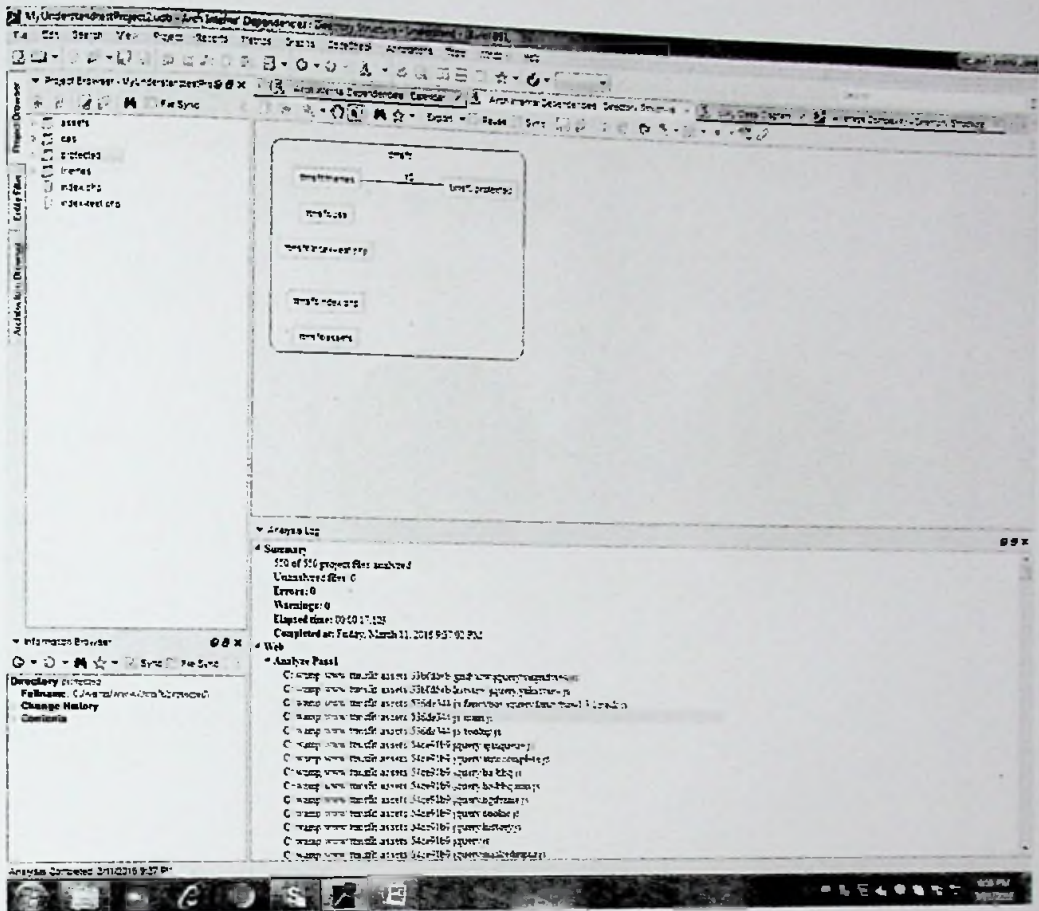
Testing and Evaluation with Test data

Sample Test cases for Black box Testing

Test No	Test Data	Expected Results	Actual Results	Conclusion
1	Logging with incorrect user name	Operation must be rejected	Message - Incorrect Username or password	Achived
2	Logging with incorrect password	Operation must be rejected	Message - Incorrect Username or password	Achived
3	Logging with incorrect user name and password	Operation must be rejected	Message - Incorrect Username or password	Achived
4	Logging with empty user name	Operation must be rejected	Message - Incorrect Username or password	Achived
5	Logging with empty password	Operation must be rejected	Message - Incorrect Username or password	Achived
6	Logging with empty Username and password	Operation must be rejected	Message - Incorrect Username or password	Achived
7	Search with incorrect name	Empty results	Message - No results found	Achived

8	Input Empty Field Name to a field with astric	Operation must be rejected	Message - Name cannot be blank.	Achived
9	Input value to a field with astric	Operation must be success	No error Message	Achived
10	Input submit number of Empty Fields with astric	Operation must be rejected	Message - Name cannot be blank.	Achived
11	Search with correct No	Operation success	Show the searched data	Achived
12	Search with correct batch No	Operation success	Show the searched data	Achived
13	Advanced Search with one the data Name Perera	Operation success	Show all the searched name with Perea	Achived
14	Update student without registration No	Operation success	Message- Refistration No cannot be blank	Achived

Table C -1 Test Cases



Metric Summary

The 'Metric Summary' dialog box contains the following data:

Metric	Value
Blank Lines	21,260
Classes	230
Code Lines	213,852
Comment Lines	27,284
Comment to Code R.	0.24
Declarative Statements	17,875
Executable Statements	66,201
Files	350
Functions	12,516
Lines	302,271

Buttons: Copy, Close, What do the metric values mean?

My Understand Project - Project Overview - Understand (File #1)

File Edit Search View Project Reports Metrics Graphs CodeView Annotations Tools Windows Help

Architecture Browser
Calendar
Directory Structure

SciTools Understand
Make your software better with advanced elements

Code Breakdown

Graph View Table View

Code Break Down

- Code Break Down
- Errors
- Warnings
- Comments

Function Breakdown

Graph View Table View

Function Breakdown

- Function Breakdown
- Errors

Analysis Log

Summary

150 of 150 project files analyzed
 Unanalyzed files: 0
 Errors: 0
 Warnings: 0
 Elapsed time: 00:01:10
 Completed on Friday, March 11, 2016 9:37:02 PM

4 Web

Analysis Pass!

- C:\temp\www\results\assets\116d56b\analysis_query\analysis.js
- C:\temp\www\results\assets\116d56b\analysis_query\analysis.js
- C:\temp\www\results\assets\116d56b\analysis_query\analysis\131\pack.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js
- C:\temp\www\results\assets\116d56b\analysis.js

File Edit Search View Project Reports Metrics Graphs CodeView Annotations Tools Windows Help

Architecture Browser
Calendar
Directory Structure

Class Breakdown

Graph View Table View

Class Break Down

- Errors

Most Complex Functions

Graph View Table View

Complex Errors

- Complex Errors
- Complex Errors
- Complex Errors
- Complex Errors
- Complex Errors

Analysis Log

Summary

150 of 150 project files analyzed



Evaluation Questionnaire

Evaluation of the Timetable Management System of Faculty of IT at University of Moratuwa

1. Strongly agree
2. Fairly agree
3. Agree
4. Disagree
5. Strongly disagree

		1	2	3	4	5	NA
1	I'm satisfying with the feature of effective use of time						
2	This system is simple and easy to use						
3	Interfaces of the system are good and attractive						
4	It is easy to search information from this system						
5	It is easy to recover and correct from the system mistakes						
6	System gives error messages which clearly tells me how to fix problems						
7	I am satisfying with the functionalities of this system						
8	System functionalities are user friendly						
9	System's manually changing part also, can be tolerate						
10	Overall, I am satisfied with TMSFIT						

Table C-2 Questionnaire Evaluation Table