

Volunteer Management System

by

Muhammad Fazril Bin Mohd Amin

12063

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Technology (Hons)
(Information Communication Technology)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Volunteer Management System

by

Muhammad Fazril Bin Mohd Amin

12063

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information Communication Technology)

Approved By;

(Mr Helmi Bin Md Rais)

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my

own except as specified in the references and acknowledgements, and that the original work contained

herein have not been undertaken or done by unspecified sources or persons.

Muhammad Fazril Bin Mohd Amin

Abstract

The Volunteer Management System (VMS) aims to partially automate the work scheduling process and the tracking of working hours for volunteers under the employ of a local science center. The currently implemented system involves the volunteers signing up on a weekly list to confirm availability for duty which will then be used by the human resources department's staff to construct a weekly duty roster. This method is tedious and time consuming, for both volunteers and staff. By creating and developing an intuitive system, VMS, the process will be streamlined, joining the availability list sign up and the weekly roster generator into one software application. This will save considerable amount of man hours, as it takes the decision making process of assigning volunteers to a timeslot in the weekly roster, based on their criterion, and relegates it to the application. The target users of this application will be the HR subdivision in charge of managing the volunteers, namely the Volunteers Management Unit (VMU) and the volunteers themselves. The project will adopt the Iterative and incremental development methodology, which promotes constant reevaluation, and will be built on the java programming language and developed on the Netbeans IDE, for the GUI and algorithm implementation. The database section to store the volunteers' data will be handled with Microsoft Access. In conclusion, this software application will save a lot of time and effort on the part of the staff and increase the efficiency with which the volunteers are managed.

Table of Contents

CHAPTER 1	INTRODUCTION	6
	1.1 Background of Study.....	6
	1.2 Problem Statement	7
	1.3 Objective	8
	1.4 Scope of Study	9
	1.5 Relevancy of the Project	10
	1.6 Feasibility Study.....	11
CHAPTER 2	LITERATURE REVIEW	12
	2.1 Terminology	12-13
	2.2 Objectives	13-14
	2.3 Proposed Solution	15
CHAPTER 3	METHODOLOGY	16
	3.1 Choosing Software Development Methodology	17
	3.2 The Phases.....	18-19
	3.3 Tools.....	20
	3.4 Key Milestones / Gant Chart.....	21
CHAPTER 4	RESULT AND DISCUSSION	22-30
CHAPTER 5	CONCLUSION AND RECOMMENDATION.....	31
REFERENCES	32
APPENDICES	33-37

1. INTRODUCTION

1.1. Background of Study

A human resource management system (HRMS) is an integral part in any enterprise seeking to thrive in the current global business climate to strive. HRMS encompasses a wide range of aspects in the successful management of employees. These include; payroll, time and attendance, performance appraisal, scheduling and et cetera [1].

Though HRMS are usually implemented throughout the entire company there are certain branches that are isolated from the main system framework. In a local science center, namely Petrosains, volunteers are employed to aid the staff in its day to day operations. These volunteers are working on a part time basis and may pick the days in which day wish to work. The process of assigning these volunteers to their post and work schedules is handled by a subdivision of Petrosains Human Resource Department, namely the Volunteer Management Unit.

The current process of constructing a weekly roster (work schedule) for the volunteer involves: The volunteers sign up on a weekly availability list, where they would fill in their name, contact number, the departments in which they are assigned to and the days that they wish to work on. At the end of the working week (typically a Friday) the VMU staff would construct a weekly roster manually using Microsoft Excel and then posting it on a notice board outside their office. In the matter of remuneration, the volunteer's receive an honorarium at the end of the month according to the amount of hours. The volunteers working hours are tallied by punch card and the total hours will be tallied manually at the end of the month by the VMU staff.

1.2. Problem Statement

The process of signing up on a weekly basis would lead to redundancy in the case of recurring volunteers where they have to fill in their details every week. The construction of the weekly roster is also tedious as the VMU staff would have to scrutinize over every volunteer that signed up for the week to determine location and timeslot assignment. This is also true for the worksheet/punch card system where the VMU staff would have to tally the hours of every volunteers individually.

The problem statement could be phrased in the following two questions:

- How to simplify the existing volunteer work scheduling process?
- Is there a way to improve the process of tracking work hours to supplement the punch card/timesheet?

1.3. Objective

1. Serve as a platform to **store volunteer information**
2. Provide a more efficient way to **allocate resources**(manpower/volunteer) via scheduling
3. Provide a means to **track timesheet hours.**

1.4. Scope of Study

- The study will cover several aspects of the volunteer management process:
 - **Scheduling.**
 - **Timesheet.**
- User Base:
 - **HR staff**
 - **Volunteers**

1.5. Relevancy of Project

The need to develop customized application software is quite high now-a-days with the ever increasing automation process within many companies

With the use of HRM S and enterprise application software(EAS) like SAP being commonplace in today's business environment, the use of manual practices in human resource management is becoming obsolete, necessitating the use of a software application to facilitate their management of the volunteers.

1.6. Feasibility

A. Technology and system feasibility

- a. Built using an open-source platform framework

B. Operational feasibility

- a. Marked improvement over old pen and paper method of scheduling

C. Schedule feasibility

- a. 28 weeks allotted sufficient for project entirety
- b. 14 weeks for research analysis & 14 weeks for development and implement
- c. Cyclical methodology

2. Literature Review

This section will mainly discuss the general idea of a Volunteer Management System. Several keywords are discussed along with problems addressed in the previous chapter until the idea of the proposed solution.

2.1 Terminology

Human Resource Management (HRM)

HRM is the management of an organization's workforce, or human resources. It is responsible for the attraction, selection, training, assessment, and rewarding of employees, while also overseeing organizational leadership and culture, and ensuring compliance with employment and labor laws.

It was found in a survey of 590 firms (both non-profit and for profit) that positive associations between HRM practices and perceptual firm performance measure [2]. It can be said that by utilizing an application that facilitates HRM practices can increase the productivity of the work force.

Scheduling

Scheduling is the process of deciding how to commit resources between a variety of possible tasks. Time can be specified or floating as part of a sequence of events.

A scheduling discipline does two things; decides order, manages queue of request [9]

Punch card/**Payroll module**

A module that automates the pay process by gathering data on employee time and attendance,. Data is generally fed from the human resources and time keeping modules to calculate automatic deposit and manual cheque writing capabilities. This module can encompass all employee-related transactions as well as integrate with existing financial management systems.

2.2 Objectives

☉ **Store volunteer information**

- A research paper outlines the possibility of database programming using Java for Microsoft-Access database utilizing Java Database Connectivity(JDBC) tool to work with the database.[8]
- Isolated systems in a company is extended, and more importantly, integrated into a company-wide system in its own right. [7]

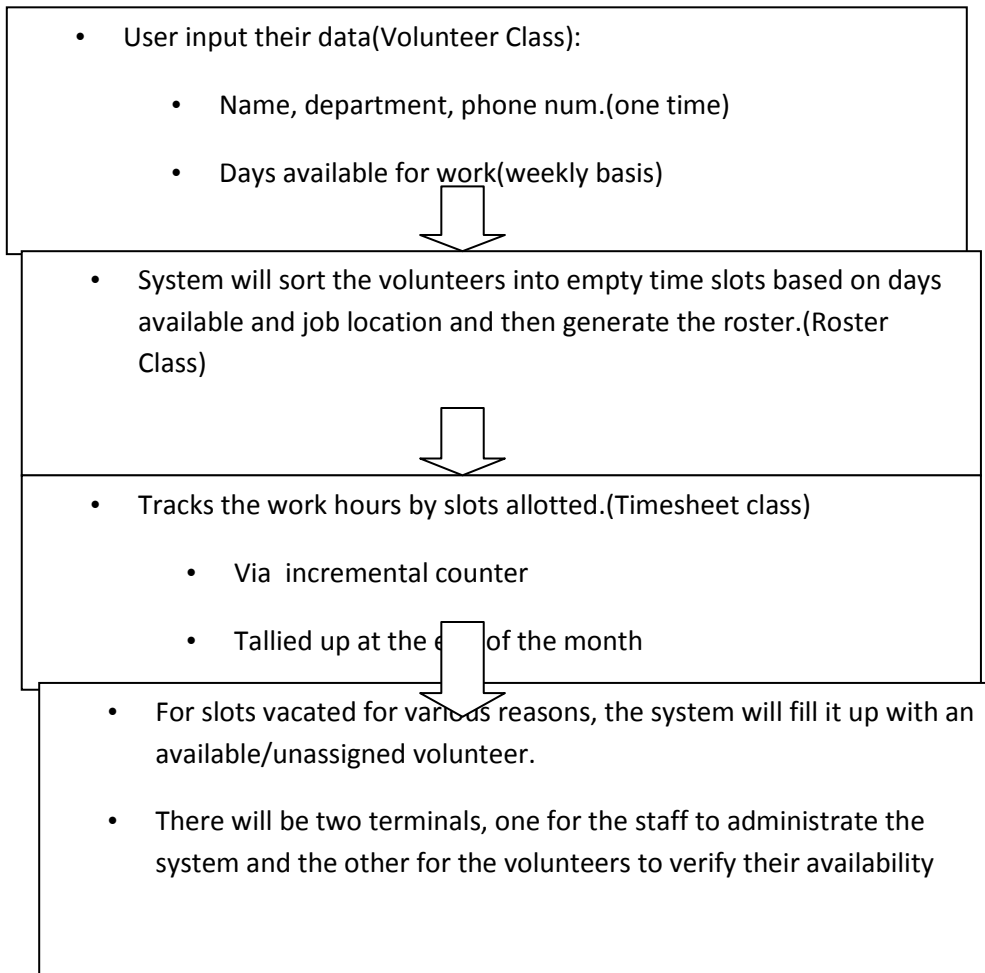
☉ **Track working hours(for payroll/remuneration)**

- Volunteer-based organizations have to keep track of a lot of info, e.g. ; Contact info, schedules ,time sheet, job sites.[4]
- A research paper analyzed, designed, implemented and tested a web-based payroll software for an external company with the Setup module, PI (personnel information) module, OT (over time) module, Payroll module and Report module. The software development life cycle followed is discussed to show how different phases are evolved and how they interact with each other to form the software that the client needs. [10]

◎ Allocate resources/Resource Scheduling

- When project managers determine schedules for resource-constrained projects, they commonly use commercial project management software packages.[5]
- Project Management paradigm will be more concerned with project work and processes, and collaboration will become essential for success [6]
- A research paper proves that it is possible to construct a modular aperiodic scheduling algorithm simulator. The simulator implements EDF scheduling algorithm, but it can easily be extended to support any aperiodic scheduling algorithm[11]
- There was a study that discussed the appointment of managers in enterprises through fuzzy neural network, to construct a new model for evaluation of managerial talent, and accordingly to develop a decision support system in human resource selection [12]. This validates the use of fuzzy logic for assignment of volunteers as the criterion is not exact, as many factors go into the job scheduling process

2.2. Proposed Solution



3. Methodology

For any software development process it is integral to plan the project well by gathering the requirements before designing and implementing the software. There are numerous methodologies designed to cater different types of software with varying constraints. This section will elaborate more on

- Choosing Software Development Methodology
- Planning, Analysis, Design and Implementation Phases
- Tools
- Key Milestone

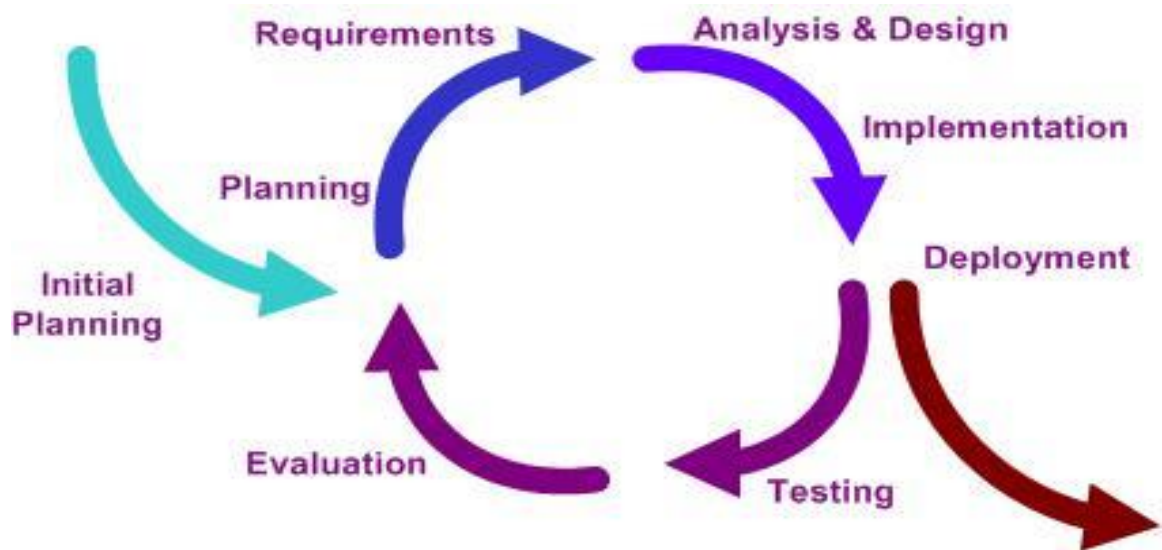
3.1. Choosing Software Development Methodology

This project employs the iterative and incremental development methodology, a constituent of the agile software development framework. It involves constant prototyping and constructing the software in smaller portions at a time, where each portion would constitute a part of the system functionality

It also provides guideline for the analysis and implementation stage, the stages most detrimental to software deployment. These guidelines are:

- Any difficulty in design, coding and testing a modification should signal the need for redesign or re-coding.
- Modifications should fit easily into isolated and easy-to-find modules. If they do not, some redesign is possibly needed.
- Modifications to tables should be especially easy to make. If any table modification is not quickly and easily done, redesign is indicated.
- Modifications should become easier to make as the iterations progress. If they are not, there is a basic problem such as a design flaw or a proliferation of patches.

- Patches should normally be allowed to exist for only one or two iterations. Patches may be necessary to avoid redesigning during an implementation phase.
- The existing implementation should be analyzed frequently to determine how well it measures up to project goals.
- Program analysis facilities should be used whenever available to aid in the analysis of partial implementations.
- User reaction should be solicited and analyzed for indications of deficiencies in the current implementation.



3.2. The Phases

3.2.1. Planning

In the planning phase, background study is conducted thoroughly. Based on the prior studies, several challenges are identified as the main problems in human resource management. To get more thorough understanding on the topic, several journals, website, books and article were reviewed. The problems identified were required to be proven correct and supported by various legitimate sources. In addition, the feasibility analysis also required to assess the project capabilities to be completed. The organizational analysis is also important to ensure the expected product will solve the problems and needs of the user. The most important analysis is the technical analysis where developer needs to identify the feasibility of developing the application. Taking technology familiarity into consideration, the product should not focus on the right customer to be targeted instead; developer should ensure that this application is the right product for the customer.

3.2.2. Analysis

In the analysis phase, the requirement is listed to clarify the 4w1h (who, what, where, when and how). Based on the problems identified in the planning phase, the functions of the application will be identified together with the user of the application. In this phase, developer should understand very well about the project and the expected application to be developed.

3.2.3. Design and Implementation

During this stage of development, the construction of the software will be divided into smaller portions, namely;

- Creating the volunteers database and inputting their data
- Construction of the GUI
- the construction of methods/functions to accommodate the sorting algorithm,
- Constructing a class for the counter tallying work hours
- Constructing a class for generating the weekly roster

3.2.4. Testing and Evaluation

The small portion completed will be tested in succession. If any flaws where found in a certain stage during the development requirements will be re-evaluated

3.2.5. Deployment

Once completed, the software will be deployed for use.

3.3. Tools

- **PC: Windows 7 OS**
- **Java programming language(J2SE)**

Java is an object oriented language; it has many reusable utilities such as mathematical methods, user interface, database, networking, etc. Possibility of portable real time software because of the architecture of Java runtime [11]

- **Netbeans IDE:**
An open-source Cross-platform integrated development environment (IDE) with built-in-support for Java Programming Language and many other languages such as ruby on rails,javascript php,python and et cetera.

3.4: Key Milestones / Gant Chart

The key milestones that need to be achieved in the first semester of Final Year Project (FYP) are as below and refer Gantt chart at the attachment.

Milestone	Date
Progress Report	5 Jun 2012
Pre-sedex	15 July 2012
VIVA	22 July 2012
Technical Report	29 August 2012
Dissertation Submission	29 August 2012

Exhibit 14: The key milestones of FYP 2

4. Results and discussion

4.1. Design of algorithm

Figure 4.1 shows the steps in which the system's timetabling algorithm is going to work. After it begins execution, it will read data from database and initiate the table. This will then be followed by allocation based on pre-determined constraints. The system will then generate the timetable.

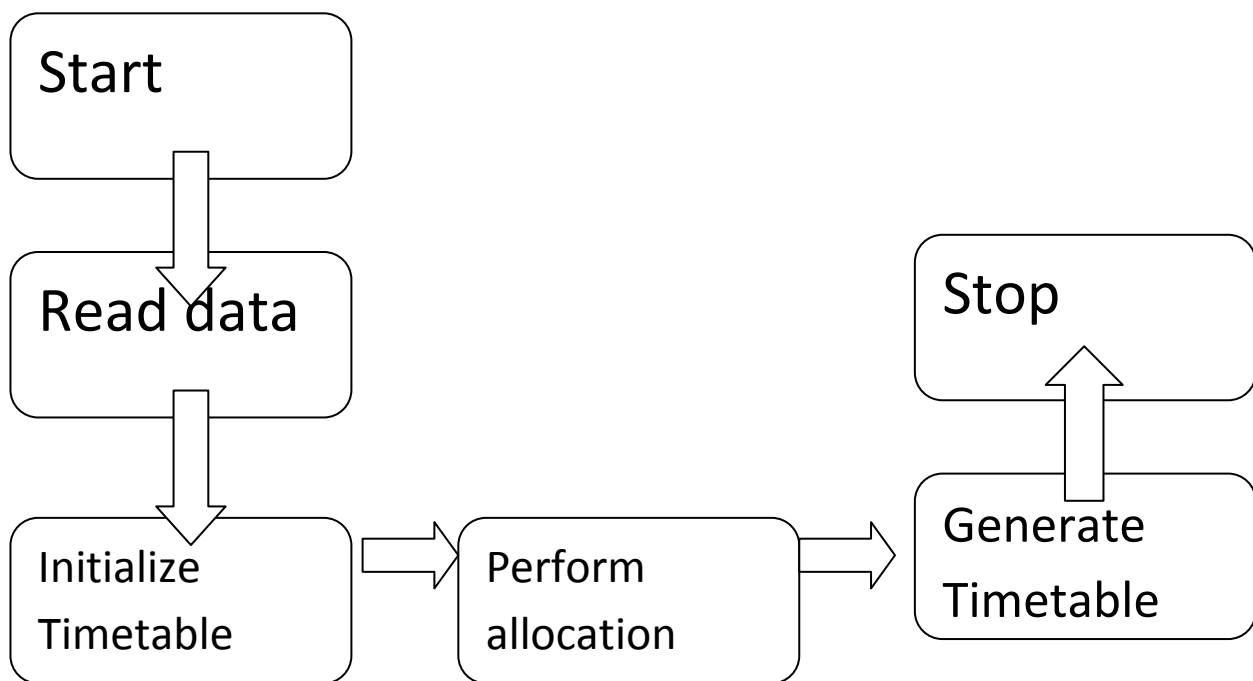


Figure 4.1

4.2 The scheduling method

The algorithm uses sequential first Come First Serve (FCFS) priority to assign volunteers to free areas. It utilizes the simple data structure for storing data. This algorithm was chosen due to its relative simplicity.

4.3 The scheduling algorithm

The algorithm is made up of several lists namely the Department List, Area list, Days List, Volunteer list and the Time Slot List. The input will be taken from the user. The algorithm functions sequentially based on two main criteria:

1. Priority based on First Come First Serve(FEFS)
2. Constrains and rule specified by user

4.4 Visual Representation of data structures

Department

- Visitor Management
- Exhibit
- Marketing

Timeslot

- 9.00AM -12.00 PM
- 9.00AM -1.00 PM
- 2.00PM-5.00 PM
- 2.00PM-5.00 PM

Area

- Main entrance
- Exit
- Dark Ride
- Dino-rama
- Space Exhibit
- Oil Rig

Days

- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

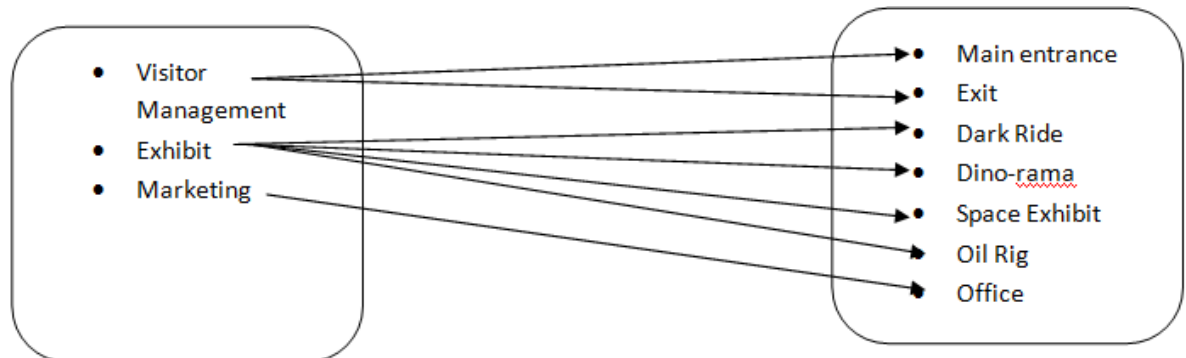
Volunteers

- Ali
- Ahmad
- Abu
-

4.4 Constraints

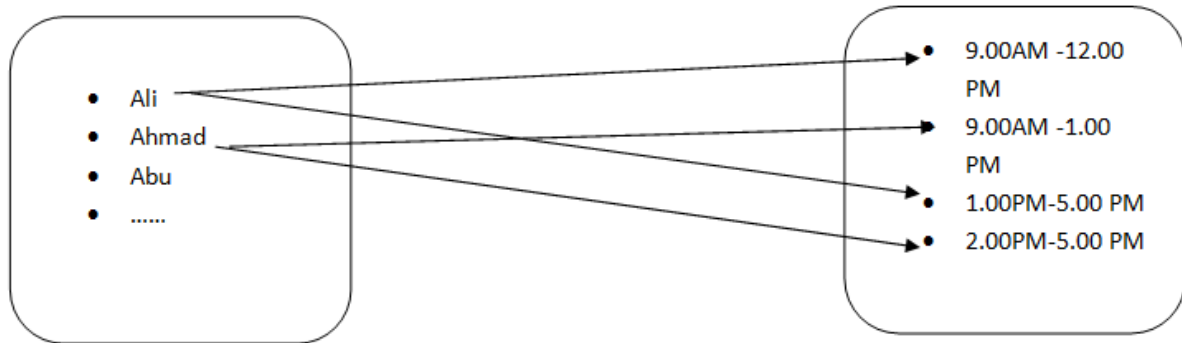
4.4.1. Department –Area Constraint

Every department is responsible for a certain area. The VM department is responsible for crowd control at the entrance and exit. The exhibit department mans the exhibit areas while the marketing volunteers assist the staff in the office area.



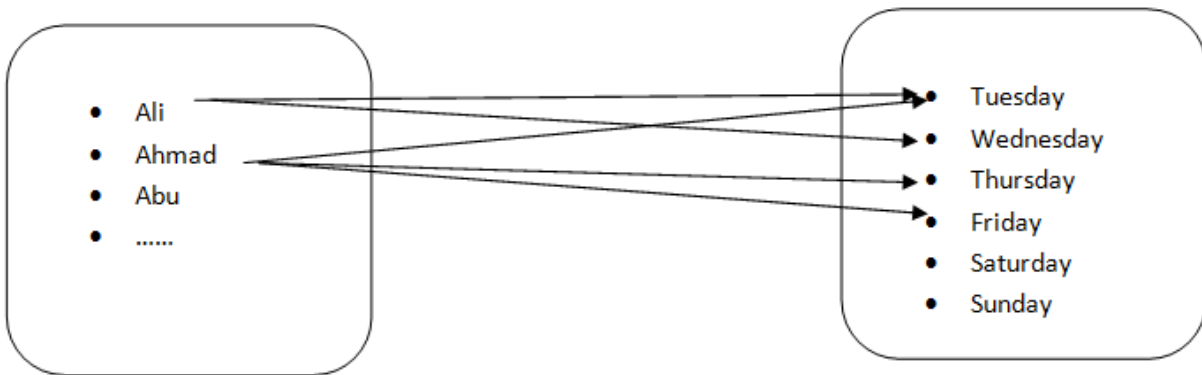
4.4.2. Volunteer –Timeslot Constraint

A volunteer can choose one time slot in the morning and one in the afternoon depending on when they take their lunchbreak

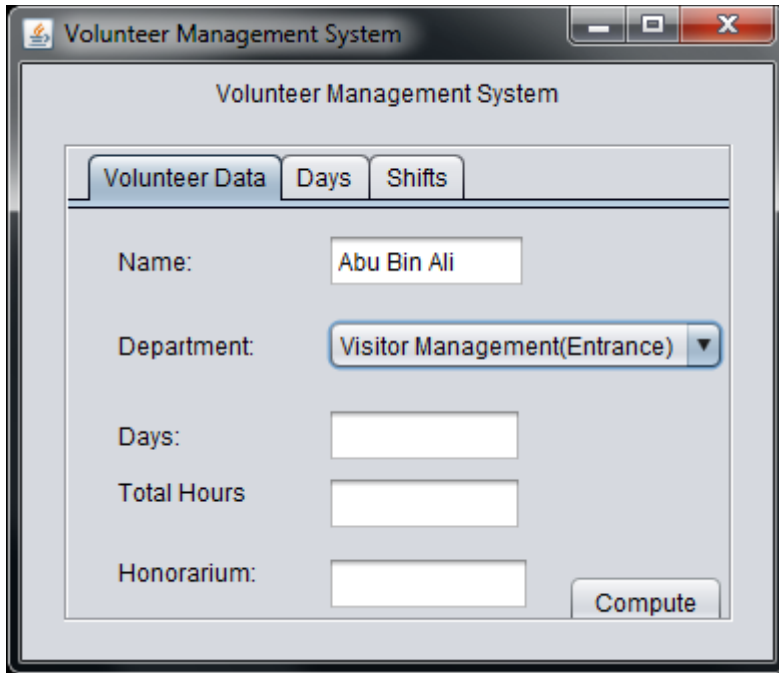


4.4.3. Volunteer -Days Constraint

Volunteers can choose what day of the week they wish to work and how many days they wish to work for that week



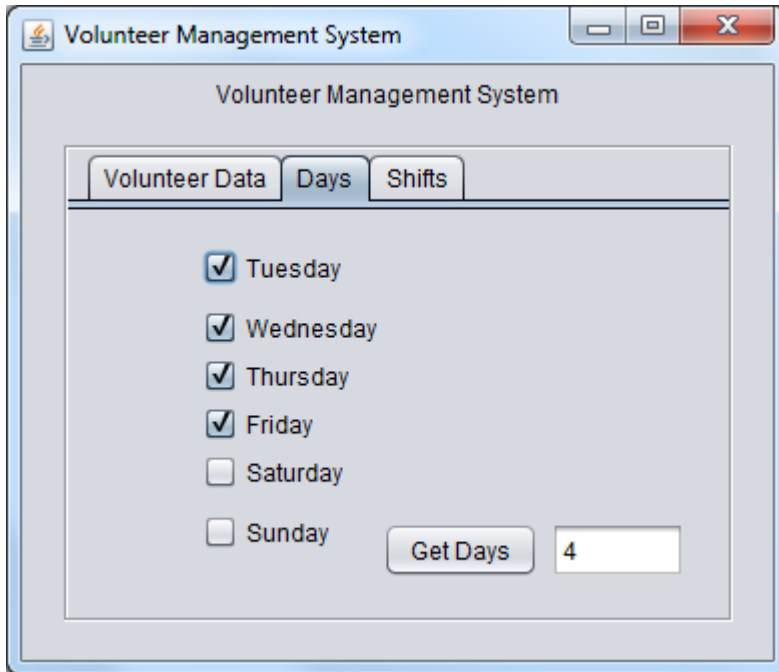
4.5 Prototype



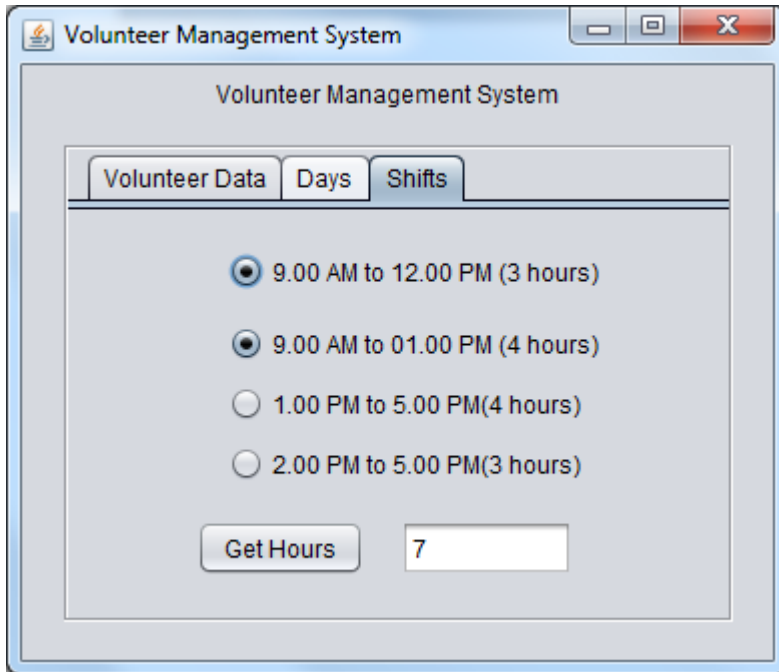
The screenshot shows a window titled "Volunteer Management System" with a standard Windows-style title bar. Inside the window, there is a sub-header "Volunteer Management System" and three tabs: "Volunteer Data", "Days", and "Shifts". The "Volunteer Data" tab is selected. The form contains the following fields and controls:

- Name: Text input field containing "Abu Bin Ali".
- Department: Dropdown menu showing "Visitor Management(Entrance)".
- Days: Empty text input field.
- Total Hours: Empty text input field.
- Honorarium: Empty text input field.
- Compute: Button located at the bottom right of the form area.

*User first enter their name and select their department



*On the day field the will be directed to the Days tab to select days they wish to work on



*The shift tab is to select which shift they wish to work on and thus number of hours they are serving

The screenshot shows a window titled "Volunteer Management System". Inside the window, there are three tabs: "Volunteer Data", "Days", and "Shifts". The "Volunteer Data" tab is selected. Below the tabs, there are several input fields and a button:

Name:	Abu Bin Ali
Department:	Visitor Management(Entrance)
Days:	4
Total Hours:	7
Honarium:	140.0

There is a "Compute" button located at the bottom right of the form area.

*They will then be returned to the main tab, whereby upon clicking the compute button will receive the amount of honorarium they will receive for the week.

*Honorarium is calculated by multiplying total days with total hours with fixed honorarium rate which is RM 5.00 per hour

5. Conclusion

The Volunteer Management System will help facilitate the management of the volunteers in Petrosains by doing away with the manual method of sign up and construction of a weekly duty roster and replacing it with an application that partially automates the process. It is hoped that the application eliminates the redundancy of the old method of weekly signup, avoid any discrepancies in shift assignment due to human error and save the company some resources by relegating the task of making work schedules and tallying work hours to a software application.

The VMS also provides a centralized database for the volunteers which will be a convenience for the VMU should they choose to expand the application to include other components of the Human Resource Management System such as; recruitment, training system, performance appraisal or et cetera aside from the components already built into the VMS which are scheduling and time and attendance components.

References

1. http://en.wikipedia.org/wiki/Human_resource_management_system
2. Delaney, John T, Huselid, Mark A. (1996) The impact of Human Resource Management practices on perceptions of organizational performance.
3. Jayne Cravens (2011). Chart of Volunteer Management software <http://www.pavro.on.ca/resources/documents/chart%20of%20volunteer%20management%20software.pdf>
4. TechSoup® Global, idealware (May 2011), A consumers guide to software for volunteer management http://www.techsoup.org/learningcenter/volunteers/13029_volunteer_management_guide.pdf
5. Trautman N., Baumann P. (2009) Resource Allocation Capabilities of Commercial Project Management Software: an Experimental Analysis
6. Nicholas C. Romano, Jr et al (2002) Collaborative Project Management Software
7. Mathias Ekstedt et al (2004) Consistent Enterprise Software System Architecture for the CIO-A utility-cost Based Approach-
8. Mrutyunjaya Swain et al (2002) Database Programming using Java, South Carolina State University
9. Scheduling An Engineering Approach to Computer Networking <http://www.cs.cornell.edu/skeshav/book/slides/scheduling/scheduling.pdf>
10. Then, H.J. ; Issac, B. (2009) Analysis, Design and Implementation of a web-based payroll application software, Univ. Technol., Kuching, Malaysia
11. Jakovljevic, G. ; Rakamaric, Z. ; Babic, D. ; (2002) Java simulator of real-time scheduling algorithm Fac. of Electron. Eng. & Comput., Zagreb Univ.
12. Liang-Chih Huang, Kuo-Shu Huang (2004) Applying Fuzzy Neural Network in Human Resource Selection System

Appendix

Technical report

The software used to develop the application was the Netbeans IDE version 7.1.2 using the java programming language. The following details the coding employed to build the graphical user interface and methods used in the application.

```
import javax.swing.ButtonGroup;
```

```
import java.lang.*;
```

```
public class MainUI extends javax.swing.JFrame {
```

```
    public MainUI() {
```

```
        super ("Volunteer Management System");
```

```
        initComponents();
```

```
        groupButton( );
```

```
    }
```

```
    private void groupButton( ) {
```

```
ButtonGroup bg1 = new ButtonGroup( );
```

```
bg1.add(jRadioButton1);
```

```
bg1.add(jRadioButton2);
```

```
bg1.add(jRadioButton3);
```

```
bg1.add(jRadioButton4);
```

```
}
```

```
@SuppressWarnings("unchecked")
```

```
// <editor-fold defaultstate="collapsed"  
desc="Generated Code">
```

```
private void initComponents() {
```

```
    jTabbedPane1 = new javax.swing.JTabbedPane();
```

```
    jPanel1 = new javax.swing.JPanel();
```

```
    jPanel5 = new javax.swing.JPanel();
```

```
jComboBox1 = new javax.swing.JComboBox();  
jLabel1 = new javax.swing.JLabel();  
jLabel2 = new javax.swing.JLabel();  
jLabel3 = new javax.swing.JLabel();  
jLabel4 = new javax.swing.JLabel();  
jButton1 = new javax.swing.JButton();  
jTextField1 = new javax.swing.JTextField();  
jTextField2 = new javax.swing.JTextField();  
jTextField3 = new javax.swing.JTextField();  
jLabel5 = new javax.swing.JLabel();  
jTextField4 = new javax.swing.JTextField();  
jPanel2 = new javax.swing.JPanel();  
jCheckBox1 = new javax.swing.JCheckBox();  
jCheckBox2 = new javax.swing.JCheckBox();  
jCheckBox3 = new javax.swing.JCheckBox();  
jCheckBox4 = new javax.swing.JCheckBox();  
jCheckBox5 = new javax.swing.JCheckBox();  
jCheckBox6 = new javax.swing.JCheckBox();
```

```
btnCheckBoxes = new javax.swing.JButton();  
jScrollPane1 = new javax.swing.JScrollPane();  
getdays = new javax.swing.JTextArea();  
jPanel3 = new javax.swing.JPanel();  
jRadioButton1 = new javax.swing.JRadioButton();  
jRadioButton2 = new javax.swing.JRadioButton();  
jButton3 = new javax.swing.JButton();  
jRadioButton3 = new javax.swing.JRadioButton();  
jRadioButton4 = new javax.swing.JRadioButton();  
jLabel6 = new javax.swing.JLabel();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants  
.EXIT_ON_CLOSE);
```

```
jTabbedPane1.setBorder(javax.swing.BorderFactory.creat  
eEtchedBorder());
```

```
jComboBox1.setModel(new  
javax.swing.DefaultComboBoxModel(new String[]  
{ "Visitor Management(Entrance)", "Visitor  
Management(Exit)", "Exhibits(Dark Ride)",  
"Exhibits(Dinorama)", "Exhibits(Oil Rig)",  
"Exhibits(Space Exhibit)", "Marketing", " " }));
```

```
jLabel1.setText("Name:");
```

```
jLabel2.setText("Department:");
```

```
jLabel3.setText("Days:");
```

```
jLabel4.setText("Total Hours");
```

```
jButton1.setText("Compute");
```

```
jButton1.addActionListener(new  
java.awt.event.ActionListener() {  
    public void  
    actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jButton1ActionPerformed(evt);
    }
});
```

```
        jTextField1.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1ActionPerformed(evt);
    }
});
```

```
        jTextField2.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField2ActionPerformed(evt);
    }
});
```

```
jLabel5.setText("Honorarium:");
```

```
javax.swing.GroupLayout jPanel5Layout = new  
javax.swing.GroupLayout(jPanel5);
```

```
jPanel5.setLayout(jPanel5Layout);
```

```
jPanel5Layout.setHorizontalGroup(
```

```
jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.LEADING)
```

```
    .addGroup(jPanel5Layout.createSequentialGroup()  
    )
```

```
        .addGap(19, 19, 19)
```

```
        .addGroup(jPanel5Layout.createParallelGroup(  
javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addComponent(jLabel2)
```

```
            .addComponent(jLabel1)
```

```
            .addComponent(jLabel4)
```

```
            .addComponent(jLabel3)
```

```
.addComponent(jLabel5))

.addPreferredGap(javax.swing.LayoutStyle.Co
mponentPlacement.RELATED, 35,
Short.MAX_VALUE)

.addGroup(jPanel5Layout.createParallelGroup(j
avax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(javax.swing.GroupLayout.Alignm
ent.TRAILING, jPanel5Layout.createSequentialGroup())

.addComponent(jTextField4)

.addGap(18, 18, 18)

.addComponent(jButton1))

.addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jTextField3)

.addComponent(jTextField2))
```



```

        .addContainerGap()
    );
    JPanel5Layout.setVerticalGroup(
jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel5Layout.createSequentialGroup(
)
            .addContainerGap()
            .addGroup(jPanel5Layout.createParallelGroup(
javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(15, 15, 15)
            .addGroup(jPanel5Layout.createParallelGroup(
javax.swing.GroupLayout.Alignment.BASELINE)

```

```
.addComponent(jComboBox1,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
.addComponent(jLabel2))  
.addGap(18, 18, 18)  
.addGroup(jPanel5Layout.createParallelGroup(j  
avax.swing.GroupLayout.Alignment.BASELINE)  
.addComponent(jLabel3)  
.addComponent(jTextField2,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
.addPreferredGap(javax.swing.LayoutStyle.Co  
mponentPlacement.RELATED)  
.addGroup(jPanel5Layout.createParallelGroup(j  
avax.swing.GroupLayout.Alignment.LEADING)  
.addComponent(jTextField3,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```

        .addComponent(jLabel4))
        .addPreferredGap(javax.swing.LayoutStyle.Co
mponentPlacement.UNRELATED)
        .addGroup(jPanel5Layout.createParallelGroup(j
avax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel5Layout.createSequentialGr
oup()
        .addGap(0, 9, Short.MAX_VALUE)
        .addComponent(jButton1))
        .addGroup(jPanel5Layout.createSequentialGr
oup()
        .addGroup(jPanel5Layout.createParallelGr
oup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel5)
        .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(0, 0, Short.MAX_VALUE)))
        .addContainerGap())

```

);

```
    javax.swing.GroupLayout jPanel1Layout = new  
    javax.swing.GroupLayout(jPanel1);
```

```
    jPanel1.setLayout(jPanel1Layout);
```

```
    jPanel1Layout.setHorizontalGroup(
```

```
    jPanel1Layout.createParallelGroup(javax.swing.GroupLa  
    yout.Alignment.LEADING)
```

```
        .addGroup(jPanel1Layout.createSequentialGroup(  
        )
```

```
            .addContainerGap()
```

```
            .addComponent(jPanel5,  
            javax.swing.GroupLayout.PREFERRED_SIZE,  
            javax.swing.GroupLayout.DEFAULT_SIZE,  
            javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
            .addContainerGap(20, Short.MAX_VALUE))
```

```
        );
```

```
    jPanel1Layout.setVerticalGroup(
```

```
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup(
)
    .addContainerGap()
    .addComponent(jPanel5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

jTabbedPane1.addTab("Volunteer Data", jPanel1);

jCheckBox1.setText("Tuesday");

jCheckBox2.setText("Wednesday");
```

```
jCheckBox3.setText("Thursday");
```

```
jCheckBox4.setText("Friday");
```

```
jCheckBox5.setText("Saturday");
```

```
jCheckBox6.setText("Sunday");
```

```
btnCheckBoxes.setText("Get Days");
```

```
btnCheckBoxes.addActionListener(new  
java.awt.event.ActionListener() {  
    public void  
actionPerformed(java.awt.event.ActionEvent evt) {  
        btnCheckBoxesActionPerformed(evt);  
    }  
});
```

```
getdays.setColumns(20);
```

```
getdays.setRows(5);
```

```
jScrollPane1.setViewportView(getdays);
```

```
javax.swing.GroupLayout jPanel2Layout = new  
javax.swing.GroupLayout(jPanel2);
```

```
jPanel2.setLayout(jPanel2Layout);
```

```
jPanel2Layout.setHorizontalGroup(  

```

```
jPanel2Layout.createParallelGroup(javax.swing.GroupLa  
yout.Alignment.LEADING)
```

```
    .addGroup(jPanel2Layout.createSequentialGroup(  
    )
```

```
        .addGap(67, 67, 67)
```

```
        .addGroup(jPanel2Layout.createParallelGroup(j  
avax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addGroup(jPanel2Layout.createSequentialGr  
oup()
```

```
                .addGroup(jPanel2Layout.createParallelGr  
oup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
                    .addComponent(jCheckBox2)
```

```

        .addComponent(jCheckBox1))
        .addContainerGap())
        .addGroup(jPanel2Layout.createSequentialGroup()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jCheckBox5)
        .addComponent(jCheckBox4)
        .addComponent(jCheckBox3)
        .addGroup(jPanel2Layout.createSequentialGroup()
        .addComponent(jCheckBox6)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnCheckBoxes)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE)

```



```

        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(115, 115, 115))))
);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLa
yout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup(
)
        .addGap(20, 20, 20)
        .addGroup(jPanel2Layout.createParallelGroup(j
avax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(jPanel2Layout.createSequentialGr
oup()

```

```
.addComponent(jCheckBox1)
    .addPreferredGap(javax.swing.LayoutStyle
.ComponentPlacement.UNRELATED)
    .addComponent(jCheckBox2)
    .addPreferredGap(javax.swing.LayoutStyle
.ComponentPlacement.RELATED)
    .addComponent(jCheckBox3)
    .addPreferredGap(javax.swing.LayoutStyle
.ComponentPlacement.RELATED)
    .addComponent(jCheckBox4)
    .addPreferredGap(javax.swing.LayoutStyle
.ComponentPlacement.RELATED)
    .addComponent(jCheckBox5)
    .addPreferredGap(javax.swing.LayoutStyle
.ComponentPlacement.UNRELATED)
    .addGroup(jPanel2Layout.createParallelGr
oup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequent
ialGroup()
    .addComponent(jCheckBox6)
```

```

        .addGap(15, 15, 15))
        .addComponent(btnCheckBoxes,
javax.swing.GroupLayout.Alignment.TRAILING))))
        .addContainerGap(27, Short.MAX_VALUE))
    );

jTabbedPane1.addTab("Days", jPanel2);

jRadioButton1.setText("9.00 AM to 12.00 PM (3
hours)");

jRadioButton1.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton1ActionPerformed(evt);
    }
});

```

```
jRadioButton2.setText("9.00 AM to 01.00 PM (4  
hours)");
```

```
jButton3.setText("Get Hours");
```

```
jButton3.addActionListener(new  
java.awt.event.ActionListener() {
```

```
    public void  
    actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jButton3ActionPerformed(evt);
```

```
    }
```

```
});
```

```
jRadioButton3.setText("1.00 PM to 5.00 PM(4  
hours)");
```

```
jRadioButton4.setText("2.00 PM to 5.00 PM(3  
hours)");
```



```

        .addComponent(jRadioButton1))
        .addComponent(jRadioButton4)))
        .addGroup(jPanel3Layout.createSequentialGroup(
oup()

        .addGap(119, 119, 119)
        .addComponent(jButton3)))
        .addContainerGap(72, Short.MAX_VALUE))
    );
    jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel3Layout.createSequentialGroup(
)

        .addGap(22, 22, 22)
        .addComponent(jRadioButton1)
        .addGap(18, 18, 18)
        .addComponent(jRadioButton2)

```

```
        .addPreferredGap(javax.swing.LayoutStyle.Co  
mponentPlacement.UNRELATED)  
        .addComponent(jRadioButton3)  
        .addPreferredGap(javax.swing.LayoutStyle.Co  
mponentPlacement.UNRELATED)  
        .addComponent(jRadioButton4)  
        .addGap(18, 18, 18)  
        .addComponent(jButton3)  
        .addContainerGap(27, Short.MAX_VALUE))  
    );
```

```
jTabbedPane1.addTab("Shifts", jPanel13);
```

```
jLabel6.setText("Volunteer Management System");
```

```
javax.swing.GroupLayout layout = new  
javax.swing.GroupLayout(getContentPane());  
getContentPane().setLayout(layout);
```

```
layout.setHorizontalGroup(
```

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(layout.createSequentialGroup()
```

```
        .addGap(97, 97, 97)
```

```
        .addComponent(jLabel6)
```

```
        .addContainerGap())
```

```
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
```

```
        .addContainerGap(21, Short.MAX_VALUE)
```

```
        .addComponent(jTabbedPane1,  
javax.swing.GroupLayout.PREFERRED_SIZE, 334,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addGap(20, 20, 20))
```

```
);
```

```
layout.setVerticalGroup(
```

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```



```
        .addGroup(javax.swing.GroupLayout.Alignment.  
TRAILING, layout.createSequentialGroup()  
        .addContainerGap()  
        .addComponent(jLabel6)  
        .addGap(18, 18, 18)  
        .addComponent(jTabbedPane1,  
javax.swing.GroupLayout.PREFERRED_SIZE, 238,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
        .addContainerGap(19, Short.MAX_VALUE))  
    );  
  
    pack();  
} // </editor-fold>
```

```
private void  
jRadioButton1ActionPerformed(java.awt.event.ActionEv  
ent evt) {  
    // TODO add your handling code here:  
}
```

```
private void  
jTextField1ActionPerformed(java.awt.event.ActionEvent  
evt) {  
    // TODO add your handling code here:  
}
```

```
private void  
jTextField2ActionPerformed(java.awt.event.ActionEvent  
evt) {  
    // TODO add your handling code here:  
}
```

```
private void  
jButton1ActionPerformed(java.awt.event.ActionEvent  
evt) {  
    // TODO add your handling code here:  
    float totaldays,totalhours,honorarium;  
    totaldays = Float.parseFloat(jTextField2.getText());
```

```
totalhours = Float.parseFloat(jTextField3.getText());

honorarium= totaldays*totalhours*5;

jTextField4.setText(String.valueOf(honorarium));
}
```

```
private void
btnCheckBoxesActionPerformed(java.awt.event.ActionE
vent evt) {
    // TODO add your handling code here:

    String s1 = "";

    if (jCheckBox1.isSelected()){
        s1=s1+""+ jCheckBox1.getText()+"\n";
    }

    if (jCheckBox2.isSelected()){
        s1=s1+""+ jCheckBox2.getText()+"\n";
    }
}
```

```
}  
if (jCheckBox3.isSelected()){  
    s1=s1+""+ jCheckBox3.getText()+"\n";  
}  
if (jCheckBox4.isSelected()){  
    s1=s1+""+ jCheckBox4.getText()+"\n";  
}  
if (jCheckBox5.isSelected()){  
    s1=s1+""+ jCheckBox5.getText()+"\n";  
}  
if (jCheckBox6.isSelected()){  
    s1=s1+""+ jCheckBox6.getText()+"\n";  
}  
getdays.setText(s1);  
}
```

```
private void
jButton3ActionPerformed(java.awt.event.ActionEvent
evt) {

    // TODO add your handling code here:

    String radioText="";

    if (jRadioButton1.isSelected()){

        radioText = jRadioButton1.getText();

    }

    if (jRadioButton2.isSelected()){

        radioText = jRadioButton2.getText();

    }

    if (jRadioButton3.isSelected()){

        radioText = jRadioButton3.getText();

    }

    if (jRadioButton4.isSelected()){
```

```

        radioText = jRadioButton4.getText();

    }

    javax.swing.JOptionPane.showMessageDialog(MainUI.this, radioText );

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

    try {

        for (javax.swing.UIManager.LookAndFeelInfo
info :
javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

```

```
javax.swing.UIManager.setLookAndFeel(info.getClassName()  
me());  
  
        break;  
    }  
}  
  
} catch (ClassNotFoundException ex) {
```

```
java.util.logging.Logger.getLogger(MainUI.class.getName()  
e()).log(java.util.logging.Level.SEVERE, null, ex);  
  
    } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(MainUI.class.getName()  
e()).log(java.util.logging.Level.SEVERE, null, ex);  
  
    } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(MainUI.class.getName()  
e()).log(java.util.logging.Level.SEVERE, null, ex);  
  
    } catch  
(javax.swing.UnsupportedLookAndFeelException ex) {
```

```

java.util.logging.Logger.getLogger(MainUI.class.getName()
e()).log(java.util.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>

/*
 * Create and display the form
 */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        new MainUI().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton btnCheckBoxes;

```



```
private javax.swing.JTextArea getdays;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton3;
private javax.swing.JCheckBox jCheckBox1;
private javax.swing.JCheckBox jCheckBox2;
private javax.swing.JCheckBox jCheckBox3;
private javax.swing.JCheckBox jCheckBox4;
private javax.swing.JCheckBox jCheckBox5;
private javax.swing.JCheckBox jCheckBox6;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
```

```
private javax.swing.JPanel jPanel3;  
private javax.swing.JPanel jPanel5;  
private javax.swing.JRadioButton jButton1;  
private javax.swing.JRadioButton jButton2;  
private javax.swing.JRadioButton jButton3;  
private javax.swing.JRadioButton jButton4;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTabbedPane jTabbedPane1;  
private javax.swing.JTextField jTextField1;  
private javax.swing.JTextField jTextField2;  
private javax.swing.JTextField jTextField3;  
private javax.swing.JTextField jTextField4;  
// End of variables declaration  
}
```

VOLUNTEER AVAILABILITY LIST

* Please write Full Name and Phone No.		JUNE							2010
No	Full Name	Dept	T 27	W 28	TH 29	F 30	ST 31	S 1	
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									

EXHIBITS TECHNOLOGY DEPARTMENT

DATE: 18 APRIL - 19 APRIL 2009

Updated 17/4/2009

18 APRIL09(SATURDAY)		Operational Volunteers Deployment Sheet (OVDS)									
No	Vol	9 - 10	10 - 11	11 - 12	12 - 1	1 - 2	2 - 3	3 - 4	4 - 5	5 - 6	6 - 7
1	Nursaadah	CR5	CR5			H1	H1	H1	ST1	ST1	X
2	Nur Fatehah	CR4	CR4		CR4		AF	AF	AF	X	X
3	Amirul Fauzi	H1	H1			GP	H2	CR4	X	X	X
4	Noor Farah Idayu	10:00	H2	H2	H2	H2		ST1	ST2	ST2	X
5	Wan M Azreen	12 NOON			H1	CR4	ST1		CR4	CR4	CR4
6	Jur Shazwani Sham	12 NOON			CR5	CR5	CR5	CR5		AF	AF
7	Munirah	11:00 AM	ST1	ST1	ST1			H2	H2	H2	X
8	Ratnasari	11:00 AM	AF	AF	AF	CR4		CR5	CR5	CR5	
9	Nor Fadillah	11:00 AM	ST2	ST2	ST2	ST2		H1	H1	X	
10				FE	FE	FE		FE	FE	FE	X
11	Safwan Hussin	10:00	INVT	INVT	INVT/GP		INVT/GP	INVT/GP	INVT	INVT	X
12	Ismail Ishak	10:00	GP	GP	GP	INVT		ST2	GP	GP	X
										Total manhours	

19 APRIL09 (SUNDAY)		Operational Volunteers Deployment Sheet (OVDS)									
No	Vol	9 - 10	10 - 11	11 - 12	12 - 1	1 - 2	2 - 3	3 - 4	4 - 5	5 - 6	6 - 7
1	Amirul Fauzi	CR5	CR5			H1	H1	H1	ST1	ST1	X
2	Nor Fadillah	CR4	CR4		CR4		AF	AF	AF	X	X
3	Khairul Najmi	H1	H1			GP	CR4	CR4	X	X	X
4	Jur Shazwani Sham	10:00	H2	H2	H2	H2		ST1	ST2	ST2	X
5	Faris Nasruddin	12 NOON			H1	CR4	ST1		CR4	CR4	CR4
6	Munirah	12 NOON			CR5	CR5	CR5	CR5		AF	AF
7	Wan M Azreen	11:00 AM	ST1	ST1	ST1			H2	H2	H2	X
8	Noor Farah Idayu	11:00 AM	AF	AF	AF	H2		CR5	CR5	CR5	
9	Nursaadah	11:00 AM	ST2	ST2	ST2	ST2		H1	H1	X	
10				FE	FE	FE		FE	FE	FE	X
11	Nur Lyana	10:00	INVT	INVT	INVT/GP		INVT/GP	INVT/GP	INVT	INVT	X
12	Nur Fatehah	10:00	GP	GP	GP	INVT		ST2	GP	GP	X