|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | **MINISTRY OF EDUCATION AND TRAINING** |  |  | | --- | | Software Project Management Plan | | E-Communication Book for School | |  | | |  |  | | --- | --- | | **Pepper Group** | | | **Group Members** | HoàngNghĩaTùng – 01115 – tunghn01115  PhạmDuyThành – 00876 – thanhpd00876  NguyễnSơnTùng – 01190 – tungns01190  VũNgọcAnh - 00850 – anhvn00850  VũViệtAnh – 01083 – anhvv01083 | | **Supervisor** | Mr. HuỳnhAnhDũng  Mr. NguyễnTấtTrung | | **Ext Supervisor** |  | | **Capstone Project code** | eCB | | |

**Record of Changes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Change Item** | **Description** | **By** | **Version** |
| 2012/05/14 |  | First creation | TungHN | 0.1 |
| 2012/05/18 |  | Minor version | TungHN | 0.7 |
| 2012/05/21 |  | Review document | TungNS | 0.8 |
| 2012/05/22 |  | Update document | TungHN | 1.0 |

Table of Contents

[Definitions and Acronyms 3](#_Toc325634873)

[1. Problem Definition 4](#_Toc325634874)

[1.1. Name of this Capstone Project 4](#_Toc325634875)

[1.2. Problem Abstract 4](#_Toc325634876)

[1.3. Project Overview 4](#_Toc325634877)

[1.3.1. The Current System 4](#_Toc325634878)

[1.3.2. The Proposed System 4](#_Toc325634879)

[1.3.3. Boundaries of the System 5](#_Toc325634880)

[1.3.4. Development Environment 5](#_Toc325634881)

[2. Project organization 6](#_Toc325634882)

[2.1. Software Process Model 6](#_Toc325634883)

[2.2. Roles and Responsibilities 6](#_Toc325634884)

[2.2.1. Project Stakeholders 6](#_Toc325634885)

[2.2.2. RASCI Model 7](#_Toc325634886)

[2.3. Tools and Techniques 8](#_Toc325634887)

[3. Project management plan 8](#_Toc325634888)

[3.1. Tasks 8](#_Toc325634889)

[3.2. Task Sheet: Assignments and Timetable 9](#_Toc325634890)

[3.3. Risk Management 10](#_Toc325634891)

[4. Appendix 11](#_Toc325634892)

[4.1. Coding Convention 11](#_Toc325634893)

[4.1.1. Naming Convention 11](#_Toc325634894)

[4.1.2. Comment 12](#_Toc325634895)

[4.2. Meeting Minutes 14](#_Toc325634896)

[4.2.1. Kick-off Meeting 13/4/2012 14](#_Toc325634897)

[4.2.2. Meeting Minute 13/4/2012 15](#_Toc325634904)

[4.2.3. Meeting Minute 24/4/2012 15](#_Toc325634905)

[4.2.4. Meeting Minute 4/5/2012 16](#_Toc325634906)

[5. Reference 17](#_Toc325634907)

## Definitions and Acronyms

|  |  |  |
| --- | --- | --- |
| Acronym | Definition | Note |
| eCB | E-Communication Book for School |  |
| PM | Project Manager |  |
| PTL | Project Technical Leader |  |
| CM | Configuration Manager |  |

Table 1 - Definitions and Acronyms

# Problem Definition

## Name of this Capstone Project

The official and formal project name is **E-Communication Book for School**.

## Problem Abstract

Nowadays, the traditional communication book for school shows its disadvantages as the raising need of the communication between the school and the students’ parents. It does not bring the informative and visual view of student’s performance and activities nor helps the school and the parents to contact immediately when they need.

In Vietnam, electronic communication book for school is still not popular due to the price, usability, functionality, etc. Therefore, we came up with the idea of creating an electronic communication book for school which will make the communication between school and parents easier and more efficient.

## Project Overview

### The Current System

Here are some points concluded from our researches about existing-communication book for school systems:

* Advantages
  + Reduce effort for teachers to manage students’ grade and attendance by a software application or by a web application.
  + Make the communication between teachers and students’ parents easier by a website.
* Disadvantages
  + The fee for maintain the system is still high.
  + Difficult to use for teachers, and the interface to use is not friendly to user.
  + The interaction between the teacher and the system for inputting grade and attendance is not friendly.

### The Proposed System

We propose a system that has following advantages:

* Support the communication between school and students’ parents by sending SMS and sending emails.
* Available 24/7
* Reduce effort for teacher to manage student grade and inform to family. Increase productivity of teacher.
* Provide informative and visual view of student’s performance reports, these reposts are useful for orient students in the future career.

### Boundaries of the System

The system under development of this Capstone project will include:

* The completed website
* All the process documents involved

### Development Environment

Below is the list of hardware and software requirements needed for development environment:

#### Hardware

* Intel Pentium D3.4 GHz or faster.
* 1 GB of RAM or more.
* 20 GB HDD or more with the speed of 5600 rpm or faster.
* A server computer for testing with the minimum configuration: 4 Gb of RAM, 100Gb of hard disk, Core 2 Duo 2.0 Ghz.

#### Software

* Operating system: Windows 7/Windows Server 2008
* DBMS: MySQL
* Microsoft Office Project Professional 2007
* Source Control: TortoiseSVN, GoogleCode
* Google Chrome 11
* Framework: Java Server Faces, Spring, Hibernate
* Mozilla Firefox 9
* Firebug 1.9
* IDE: Eclipse
* Web Server: Apache Tomcat
* Microsoft Office Project Visio 2007
* Enterprise Architect v7.5
* Microsoft Office Project Professional 2007

# Project organization

## Software Process Model

For the development process, we choose Iterative model.



Figure 2– Iterative development model

## Roles and Responsibilities

### Project Stakeholders

**Project Instructor**

* Mr. Huynh Anh Dung
* Mr. Nguyen Tat Trung

**Project team member**



Figure 2‑2 Project Team Organization

### RASCI Model

**R – Responsible** – Those who are responsible for the task, ensuring that it is done as per the Approver. There is typically one role with a participation type of Responsible, with others delegated to assist as the 'Support' in the work required. (RASCI separately identifies those who participate in a supporting role).

**A – Accountable** – Those who are ultimately accountable for the correct and thorough completion of the deliverable or task, and the one to whom Responsible is accountable. In other words, an Accountable must sign off (Approve) on work that Responsible provides. There must be only one Accountable specified for each task or deliverable.

**S – Support** – Resources allocated to Responsible. Unlike Consulted, who may provide input to the task, Support will assist in completing the task.

**C – Consulted** – Those whose opinions are sought; and with whom there is two-way communication.

**I – Informed** – Those who are kept up-to-date on progress, often only on completion of the task or deliverable; and with whom there is just one-way communication.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | Project Team Member | | | | |
|  |  | **Project Instructor** |  | **Hoang Nghia Tung** | **Nguyen Son Tung** | **Vu Viet Anh** | **Vu Ngoc Anh** | **Pham Duy Thanh** |
| 1 | **Initiation** |  |  |  |  |  |  |  |
|  | Generate project ideas | C |  | A | S | R | S | S |
|  | Create Report 1: Project Introduction | C |  | A |  |  |  |  |
|  | Hold kick-off meeting | I |  | A | R | R | R | R |
| 2 | **Planing** | C |  | A | S | S | S | S |
| 3 | **Executing** |  |  |  |  |  |  |  |
|  | Analysis | C |  | A |  | R |  | R |
|  | Design | C |  | A | R |  | S | R |
|  | Implementation | C |  | A | R |  | R | R |
|  | Testing | C |  | A | R | I | R | R |
|  | Deployment | C |  | A | R | R | R | R |

Table 2‑1: RASCI Model

## Tools and Techniques

* Front-end technologies: ASP.NET MVC 3.0, jQuery, AJAX
* Tools: NetBeans, MySQL
* ORM: Entity Framework
* Architecture and design patterns: Java Server Faces, Spring MVC Framework, Hibernate

# Project management plan

## Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Task Name | Duration | Predecessors |
| **1** | **eCB project** | **85 days** |  |
| **1.1** | **Initiating** | **8 days** |  |
| 1.1.1 | Generate project ideas | 1 day |  |
| 1.1.2 | Create Report 1: Project Introduction | 3 days | 3 |
| 1.1.3 | Review Report 1 | 1 day | 4 |
| 1.1.4 | Update and finalize Report 1 | 1 day | 5 |
| 1.1.5 | Hold kick-off meeting | 8 hrs |  |
| **1.1.6** | **Release Report 1: Introduction** | **1 day** | **6** |
| **1.2** | **Planning** | **5 days** |  |
| 1.2.1 | Create project plan | 1 day |  |
| 1.2.2 | Create Report 2: Software Project Management Plan | 3 days | 10 |
| 1.2.3 | Review, Update and finalize Report 2 | 0.9 days | 11 |
| **1.2.4** | **Release Report 2: Software Project Management Plan** | **0.1 days** | **12** |
| **1.3** | **Executing** | **70 days** |  |
| **1.3.1** | **Analysis** | **15 days** |  |
| 1.3.1.1 | Write requirements | 3 days |  |
| 1.3.1.2 | Create Software Requirement Specification (SRS) document | 8 days | 16 |
| 1.3.1.3 | SRS completed | 1 day | 17 |
| **1.3.1.4** | **Release Report 3: Software Requirement Specification** | **1 day** | **18** |
| **1.3.2** | **Design** | **10 days** |  |
| 1.3.2.1 | Database design | 3 days |  |
| 1.3.2.2 | Architecture design | 2 days | 21 |
| 1.3.2.3 | Detailed design | 5 days |  |
| 1.3.2.4 | Create Software Design Description (SDD) document | 4.9 days |  |
| **1.3.2.5** | **Release Report 4: Software Detailed Design** | **1 day** | **24** |
| **1.3.3** | **Implementation** |  |  |
| 1.3.3.1 | Create coding framework | 2 days |  |
| 1.3.3.2 | GUI design | 3 days | 27 |
| 1.3.3.3 | Coding | 29 days | 28 |
| 1.3.3.4 | Coding completed | 1 day | 29 |
| 1.3.3.5 | Create Software User Mannual (SUM) document | 10 days |  |
| **1.3.4** | **Testing** | **5 days** |  |
| 1.3.4.1 | Write system test case | 1 day |  |
| 1.3.4.2 | Execute system test | 3 days | 33 |
| 1.3.4.3 | System test completed | 1 day | 34 |
| 1.3.4.4 | Update Software User Mannual (SUM) document | 10 days |  |
| 1.3.4.5 | Create Software Test Documentation (STD) document | 5 days |  |
| **1.3.4.6** | **Release Report 5: Software Test Documentation** | **1 day** |  |
| **1.3.4.7** | **Release Report 6: Software User Manual** | **1 day** |  |
| 1.3.5 | **Deployment** | **5 days** |  |
| 1.3.5.1 | Input initial data | 4 days |  |
| 1.3.5.2 | System deployed | 1 day |  |
| 1.3.6 | **Closing** | **5 days** |  |
| 1.3.6.1 | Prepare final project presentation | 4 days |  |
| 1.3.6.2 | Project presentation | 2 hrs |  |
| 1.3.6.3 | Project completed | 0 days |  |

Table 3‑1: Project Work breakdown structure

## Task Sheet: Assignments and Timetable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Task Name | Start | Finish | Resource Names |
| **1** | **eCB project** | **Mon 5/7/12** | **Fri 8/31/12** |  |
| **1.1** | **Initiating** | **Mon 5/7/12** | Wed 5/16/12 |  |
| 1.1.1 | Generate project ideas | Mon 5/7/12 | Mon 5/7/12 | eCB Team |
| 1.1.2 | Create Report 1: Project Introduction | Tue 5/8/12 | Thu 5/10/12 | AnhVV |
| 1.1.3 | Review Report 1 | Fri 5/11/12 | Fri 5/11/12 | AnhVN,ThanhPD,TungHN,TungNS |
| 1.1.4 | Update and finalize Report 1 | Tue 5/15/12 | Tue 5/15/12 | AnhVV |
| 1.1.5 | Hold kick-off meeting | Fri 5/11/12 | Sat 5/12/12 | eCB Team |
| **1.1.6** | **Release Report 1: Introduction** | **Wed 5/16/12** | **Wed 5/16/12** | **TungHN** |
| **1.2** | **Planning** | **Mon 5/14/12** | **Fri 5/18/12** |  |
| 1.2.1 | Create project plan | Mon 5/14/12 | Mon 5/14/12 | TungHN |
| 1.2.2 | Create Report 2: Software Project Management Plan | Tue 5/15/12 | Thu 5/17/12 | TungHN |
| 1.2.3 | Review, Update and finalize Report 2 | Fri 5/18/12 | Fri 5/18/12 | TungHN,TungNS |
| **1.2.4** | **Release Report 2: Software Project Management Plan** | **Fri 5/18/12** | **Fri 5/18/12** | **TungHN** |
| **1.3** | **Executing** | **Mon 5/14/12** | **Fri 8/17/12** |  |
| **1.3.1** | **Analysis** | **Mon 5/14/12** | **Fri 6/1/12** |  |
| 1.3.1.1 | Write requirements | **Wed 5/16/12** | Fri 5/18/12 | AnhVV,ThanhPD |
| 1.3.1.2 | Create Software Requirement Specification (SRS) document | Mon 5/21/12 | Wed 5/30/12 | AnhVV,ThanhPD |
| 1.3.1.3 | SRS completed | Thu 5/31/12 | Thu 5/31/12 | AnhVV,ThanhPD |
| **1.3.1.4** | **Release Report 3: Software Requirement Specification** | **Fri 6/1/12** | **Fri 6/1/12** | **TungHN** |
| **1.3.2** | **Design** | **Mon 5/28/12** | **Fri 6/8/12** |  |
| 1.3.2.1 | Database design | Mon 5/28/12 | Wed 5/30/12 | AnhVN,TungHN,TungNS |
| 1.3.2.2 | Architecture design | Thu 5/31/12 | Fri 6/1/12 | AnhVN,TungHN,TungNS |
| 1.3.2.3 | Detailed design | Mon 6/4/12 | **Fri 6/8/12** | AnhVN,TungHN,TungNS,ThanhPD |
| 1.3.2.4 | Create Software Design Description (SDD) document | Mon 6/4/12 | **Fri 6/8/12** | AnhVN,TungHN,TungNS,ThanhPD |
| **1.3.2.5** | **Release Report 4: Software Detailed Design** | **Fri 6/8/12** | **Fri 6/8/12** | **TungHN** |
| **1.3.3** | **Implementation** |  | **Fri 7/20/12** |  |
| 1.3.3.1 | Create coding framework | Mon 6/4/12 | Tue 6/5/12 | TungHN,TungNS |
| 1.3.3.2 | GUI design | Wed 6/6/12 | Fri 6/8/12 | ThanhPD |
| 1.3.3.3 | Coding | Mon 6/11/12 | Thu 7/19/12 | AnhVN,ThanhPD,TungHN,TungNS |
| 1.3.3.4 | Coding completed | Fri 7/20/12 | Fri 7/20/12 |  |
| 1.3.3.5 | Create Software User Manual (SUM) document | Mon 7/9/12 | Fri 7/20/12 | AnhVV |
| **1.3.4** | **Testing** | **Mon 7/23/12** | **Fri 7/27/12** |  |
| 1.3.4.1 | Write system test case | **Mon 7/23/12** | Mon 7/23/12 | AnhVV,ThanhPD |
| 1.3.4.2 | Execute system test | **Tue 7/24/12** | Thu 7/26/12 | AnhVV,ThanhPD |
| 1.3.4.3 | System test completed | Fri 7/27/12 | Fri 7/27/12 | AnhVV,ThanhPD |
| 1.3.4.4 | Update Software User Manual (SUM) document | Mon 7/9/12 | Fri 7/20/12 | AnhVN |
| 1.3.4.5 | Create Software Test Documentation (STD) document | Mon 7/23/12 | Fri 7/27/12 | AnhVV,ThanhPD |
| **1.3.4.6** | **Release Report 5: Software Test Documentation** | **Fri 7/27/12** | **Fri 7/27/12** | **TungHN** |
| **1.3.4.7** | **Release Report 6: Software User Manual** | **Fri 7/27/12** | **Fri 7/27/12** | **TungHN** |
| 1.3.5 | **Deployment** | **Mon 7/30/12** | **Fri 8/3/12** |  |
| 1.3.5.1 | Input initial data | Mon 7/30/12 | Thu 8/2/12 | eCB Team |
| 1.3.5.2 | System deployed | Mon 7/30/12 | Mon 7/30/12 | eCB Team |
| 1.3.6 | **Closing** | **Mon 8/6/12** | **Fri 8/10/12** |  |
| 1.3.6.1 | Prepare final project presentation | Mon 8/6/12 | Thu 8/9/12 | eCB Team |
| 1.3.6.2 | Project presentation | Mon 8/6/12 | Mon 8/6/12 | eCB Team |
| 1.3.6.3 | Project completed | Mon 8/6/12 | Mon 8/6/12 | eCB Team |

Table 3‑2: Project Resource Assignment

## Risk Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Risk Description | Propability | Effect | Status |
| 1 | **New Technology:**  The system uses Java Framework and Technology which the project has not experiences of. | High | Serious | Occurred |
| 2 | **Project Management skill:**  The Project Management lacks of real-working environment experiences. | Moderate | Serious | Potential |
| 3 | **Resource Experience:**  Team member lack of software development and testing experience. | Moderate | Insignificant | Not relevant |
| 4 | **User Involvement:**  The system requires involvement of teachers of school and the students to get the requirement correctly. But the system development time is the school summer vacation; therefore it is hard to get involvement of teachers and students. | High | Tolerable | Responded |

Table 3-3: Risk Management

# Appendix

## Coding Convention

### Naming Convention

Naming conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier—for example, whether it’s a constant, package, or class—which can be helpful in understanding the code.

|  |  |  |
| --- | --- | --- |
| **Identifier Type** | **Rules for Naming** | **Examples** |
| Classes | Class names should be nouns, in mixed case  with the first letter of each internal word capi-  talized. Try to keep your class names simple  and descriptive. Use whole words—avoid  acronyms and abbreviations (unless the abbre-  viation is much more widely used than the  long form, such as URL or HTML). | class Raster;  class ImageSprite; |
| Interfaces | Interface names should be capitalized like  class names. | interface RasterDelegate;  interface Storing; |
| Methods | Methods should be verbs, in mixed case with  the first letter lowercase, with the first letter of  each internal word capitalized. | run();  runFast();  getBackground(); |
| Variables | Except for variables, all instance, class, and  class constants are in mixed case with a lower-  case first letter. Internal words start with capi-  tal letters.  Variable names should be short yet meaning-  ful. The choice of a variable name should be  mnemonic— that is, designed to indicate to the  casual observer the intent of its use. One-char-  acter variable names should be avoided except  for temporary “throwaway” variables. Com-  mon names for temporary variables are i, j, k,  m, and n for integers; c, d, and e for characters. | inti;  char \*cp;  float myWidth; |
| Constants | The names of variables declared class con-  stants and of ANSI constants should be all  uppercase with words separated by under-  scores (“\_”). (ANSI constants should be  avoided, for ease of debugging.) | int MIN\_WIDTH = 4;  int MAX\_WIDTH = 999;  int GET\_THE\_CPU = 1; |

### Comment

#### Implementation Comment Formats

Programs can have four styles of implementation comments: block, single-line, trailing and end-of-line.

#### Block Comments

Block comments are used to provide descriptions of files, methods, data structures and algorithms. Block comments should be used at the beginning of each file and before each method. They can also be used in other places, such as within methods. Block comments inside a function or method should be indented to the same level as the code they describe.

A block comment should be preceded by a blank line to set it apart from the rest of the code. Block comments have an asterisk “\*” at the beginning of each line except the first.

/\*

\* Here is a block comment. \*/

Block comments can start with /\*-, which is recognized by **indent**(1) as the beginning of a block comment that should not reformatted. Example:

/\*

* Here is a block comment with some very special
* formatting that I want indent(1) to ignore.

\*

\* one

\* two

\* three

\*/

**Note:** If you don’t use**indent**(1), you don’t have to use/\*-in your code or make any otherconcessions to the possibility that someone else might run **indent**(1) on your code.

#### Single-Line Comments

Short comments can appear on a single line indented to the level of the code that follows. If a comment can’t be written in a single line, it should follow the block comment format  [(se](#page11)e  [section 5.1.1](#page11)). A single-line comment should be preceded by a blank line. Here’s an example of a single-line comment in Java code (also see  [“Documentation Comments” on page](#page13) 9):

if (condition) {

/\* Handle the condition. \*/

...

}

#### Trailing Comments

Very short comments can appear on the same line as the code they describe, but should be shifted far enough to separate them from the statements. If more than one short comment appears in a chunk of code, they should all be indented to the same tab setting. Avoid the assembly language style of commenting every line of executable code with a trailing comment.

Here’s an example of a trailing comment in Java code

if (a == 2) {

return TRUE; /\* special case \*/

} else {

return isprime(a); /\* works only for odd a \*/

}

#### End-Of-Line Comments

The // comment delimiter begins a comment that continues to the newline. It can comment out a complete line or only a partial line. It shouldn’t be used on consecutive multiple lines for text comments; however, it can be used in consecutive multiple lines for commenting out sections of code. Examples of all three styles follow:

if (foo > 1) {

// Do a double-flip.

...

}

else

return false;// Explain why here.

//if (bar > 1) {

//

* // Do a triple-flip.
* ...

//} //else

* + return false;

## Meeting Minutes

### Kick-off Meeting 13/4/2012

**Meeting Minutes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Meeting supervisor | **Date** | 11/4/2012 |
| **Facilitator** | Restaurant | **Time** | 13:00 – 14:00 |
| **Location** | Duy Tan Str. | **Scribe** | TungHN |
| **Attendees** | DungHA (Supervisor), TungHN (Leader), AnhVV, AnhVN, TungNS, ThanhPD | | |
| **Absent** |  | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1. | Team member introduction | Introduce all members of team to supervisor |
| 2. | Kick-off party | Make supervisor and team members closer |



### Meeting Minute 13/4/2012

**Meeting Minutes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Capstone Subject Brainstorming | **Date** | 13/4/2012 |
| **Facilitator** | Miriam’s Coffee | **Time** | 15:00 – 17:00 |
| **Location** | Ton That Thuyet Str. | **Scribe** | TungHN |
| **Attendees** | TungHN (Leader), AnhVV, AnhVN, TungNS, ThanhPD | | |
| **Absent** |  | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1. | Choosing Capstone Subject | **Brainstorming and choose a subject for Capstone Project** |

| **Action Plan** | | | |
| --- | --- | --- | --- |
| No. | Action Item(s) | Owner | Target Date |
| 1. | Writing Capstone Registration | TungHN | 14/4 |

### Meeting Minute 24/4/2012

**Meeting Minutes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Capstone Project Introduction | **Date** | 24/4/2012 |
| **Facilitator** | Miriam’s Coffee | **Time** | 15:00 – 17:00 |
| **Location** | Ton That Thuyet Str. | **Scribe** | TungHN |
| **Attendees** | TungHN (Leader), AnhVV, AnhVN, TungNS, ThanhPD | | |
| **Absent** |  | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1. | Project introduction | Introduce eCB Project by ThanhPD |
| 2. | System Functions | Discuss about what functions the system should have |
| 3. | Assign tasks | Assign researching tasks for each members |

| **Action Plan** | | | |
| --- | --- | --- | --- |
| No. | Action Item(s) | Owner | Target Date |
| 1. | Writing Project Introduction | TungHN | 24/4 |
| 2. | Writing Plan |  |  |

### Meeting Minute 4/5/2012

**Meeting Minutes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Team name brainstorming | **Date** | 4/5/2012 |
| **Facilitator** | Miriam’s Coffee | **Time** | 15:00 – 17:00 |
| **Location** | Ton That Thuyet Str. | **Scribe** | TungHN |
| **Attendees** | TungHN (Leader), AnhVV, AnhVN, TungNS, ThanhPD | | |
| **Absent** |  | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1. | Choose team name | Choose a name for team |
| 2. | Choose working place | Choose a place to work full-time. |

# Reference

[1] Wikipedia – Iterative and incremental development

<http://en.wikipedia.org/wiki/Iterative_and_incremental_development>

[2] Oracle – Coding Convention

<http://www.oracle.com/technetwork/java/codeconv-138413.html>

- End of document -