[COMPANY NAME] REGISTRATION SYSTEM

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.0 GENERAL INFORMATION

The European Innovation, Technology, and Science Center Foundation ([COMPANY NAME])

The European Chamber of Commerce of the Philippines, the German Development Cooperation, and the Asia-Europe Foundation of the Philippines gathered together to promote innovation, technology and science thus forming European Innovation, Technology and Center Foundation ([COMPANY NAME]) – a training center aimed at providing training courses and seminars for students, graduates, teachers, and IT/BPO professionals in cooperation with leading providers.

Training programs include soft skills such as customer relations and leadership as well as technical skills like basic and advanced desktop applications and computer programming.

In cooperation with non-government organizations and local social services departments, the [COMPANY NAME] also provides high school awareness seminars and out-of-school youth development programs.

1.1 Purpose

Design a dynamic Database System for [COMPANY NAME] Registration process.

1.2 Problem Statement

The current process of [COMPANY NAME] when it comes to storing data such as training manuals, trainees' and facilitators' records, progress and performance reports solely relies on manual encoding. [COMPANY NAME] does not have a centralized database or system that will easily track all their records and data. A lot of paper work needs to be accomplished and records/data are manually encoded by the employee, trainees and facilitators. Duplication and loss of data has been inevitable and employees also find it difficult and time consuming to consolidate records and reports.

1.3 Solution Statement

To design a Content Management System that will:

- ✓ Automate the entire documentation process from trainee registration, facilitator schedule arrangement, down to invoicing since the system is integrated to [COMPANY NAME]'s billing system.
- ✓ Allow easy tracking for all types of data such as training schedules, trainees' personal details, success rate of event, etc.
- ✓ Centralize all records, trackers, manuals and files.
- ✓ Allow automatic consolidation of all data thus eliminating duplication and loss of pertinent information.
- ✓ Faster turn-around time when it comes to accomplishing tasks.

1.4 Scope & System Overview

Project Name: **[COMPANY NAME] Registration System**

[COMPANY NAME] CMS web based System is designed to:

- Automate business process
- Easy Tracking
- Data Consolidation
- Report Generation

The system uses Hypertext Pro-Processor (PHP) as its core scripting language, MySQL as the database storage, plus several client scripting languages such as AJAX, JQuery and CSS for other add-on designs and features.

The main functionalities of the system are the following:

- Adding, editing, and deleting data
- User authentication for data access control
- Logs to track user activities
- Filter bar for searching of records
- RTE: Rich Text Editor tool bar (similar to Word document features)

1.5 Points of Contact

Joana Narciso – Business Development Manager

Contact Numbers: 845-1324 loc. 224 / 09151280383
Email Address: joana@[company name].com

European Innovation, Technology, and Science Center Foundation
19/F Philippine AXA Life Centre, Sen. Gil Puyat Ave. cor. Tindalo St., Makati City
Landline: 759-6680/ 759-2246 | Fax: 759-2247 |Website: info@[company name].com

2.0 DATABASE IDENTIFICATION AND DESCRIPTION

2.0 DATABASE IDENTIFICATION AND DESCRIPTION

2.1 Systems Using the Database

1. [COMPANY NAME] Registration System as main data storage
2. [COMPANY NAME] Accounting Database for invoice reference

2.2 Relationship to Other Databases

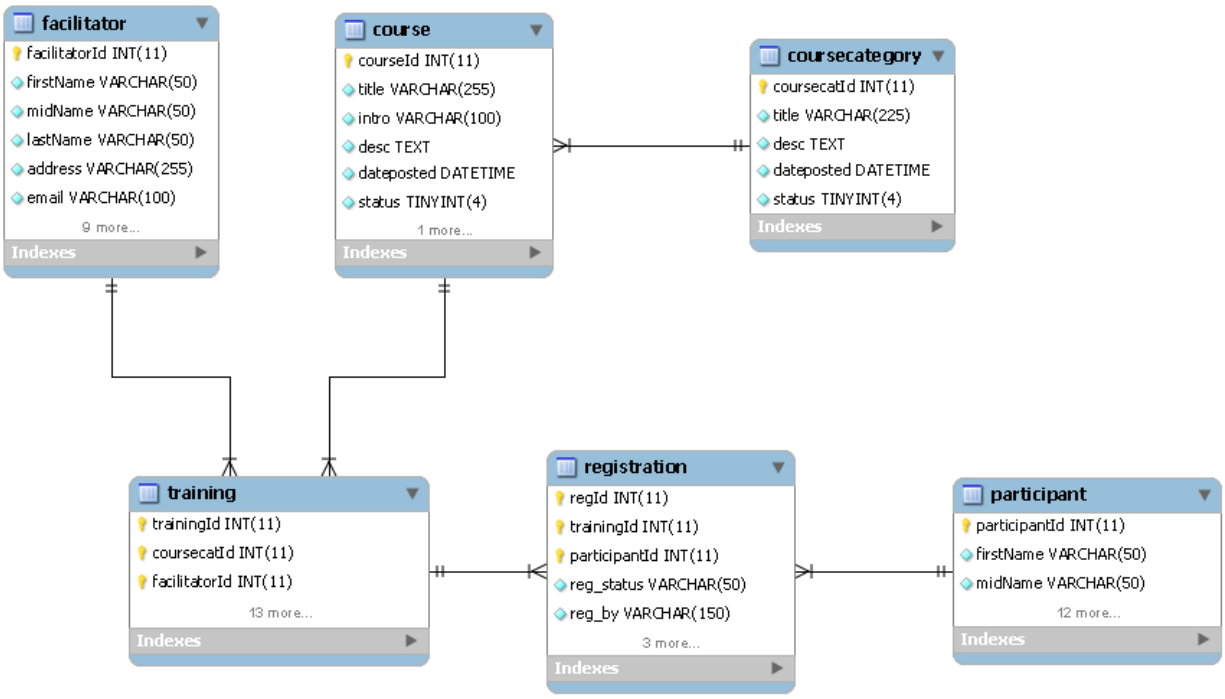
[COMPANY NAME] Registration System is connected to [COMPANY NAME] Accounting Database.

2.3 Database Information

Database Name: **[COMPANY NAME]**

Table Name	Description	Primary Key	Foreign Key
course	Course table	courseId	cousrcatId
coursecategory	Course category, look up table for courses	cousrcatId	
facilitator	Facilitators profile table	facilitatorId	
training	Training table	trainingId	coursed, facilitatorId
users	User Accounts table	user_id	
logs	Audit logs for user activities	logs_id	
participant	Participants profile table	participantId	
registration	Registration table	regId	trainingId, participantId

2.4 ERD Diagram



2.5 Data Dictionary

Table structure for table **course**

Field	Description	Data Type	Null
courseId	Course ID	int(11)	No
title	Course Title	varchar(255)	No
intro	Course Introduction	varchar(100)	No
desc	Course Description	text	No
dateposted	Course Date Posted	datetime	No
status	Course Status	tinyint(4)	No
coursecatId	Course Category ID	int(11)	No

Table structure for table **coursecategory**

Field	Description	Data Type	Null
coursecatId	Course Category ID	int(11)	No
title	Course Category Ttitle	varchar(225)	No
desc	Course Categorry Description	text	No
dateposted	Course Category Date Posted	datetime	No
status	Course Category Status	tinyint(4)	No

Table structure for table **facilitator**

Field	Description	Date Type	Null
facilitatorId	Facilitator's ID	int(11)	No
firstName	Facilitator's First Name	varchar(50)	No
midName	Facilitator's Middle Name	varchar(50)	No
titleName	Facilitator's Title	varchar(50)	No
midName	Facilitator's Middle Name	varchar(50)	No
address	Facilitator's Address	varchar(255)	No
email	Facilitator's Email Address	varchar(100)	No
phoneno	Facilitator's Phone Number	varchar(50)	No
mobile_1	Facilitator's Mobile Number	varchar(50)	No
mobile_2	Facilitator's Mobile Number 2	varchar(50)	No
fax	Facilitator's Fax Number	varchar(50)	No
sex	Facilitator's Gender	char(1)	No
birth_date	Facilitator's Birth Date	date	No
salary	Facilitator's Salary	decimal(9,2)	No
tin_number	Facilitator's Tax Identification Number	int(11)	No
dateposted	Facilitator's Date Posted	datetime	No
status	Facilitator's Status	tinyint(4)	No

Table structure for table **logs**

Field	Description	Data Type	Null
logs_id	Logs ID	int(11)	No
logs_table	Logs Table	varchar(50)	No
logs_rec_id	Logs Record ID	int(11)	No
logs_action	Logs Action	varchar(255)	No
logs_keyword	Logs Keyword	varchar(255)	No
logs_user_id	Logs User ID	int(11)	No
logs_username	Logs Username	varchar(100)	No
logs_ipadd	Logs IP Address	varchar(50)	No
logs_datepost	Logs Date Posted	datetime	No

Table structure for table **participant**

Field	Description	Data Type	Null
participantId	Participant ID	int(11)	No
firstName	Participant's First Name	varchar(50)	No
midName	Participant's Middle Name	varchar(50)	No
lastName	Participant's Last Name	varchar(50)	No
address	Participant's Address	varchar(255)	No
email	Participant's Email	varchar(100)	No
phoneno	Participant's Phone Number	varchar(50)	No
mobile_1	Participant's Mobile Number	varchar(50)	No
mobile_2	Participant's Mobile Number 2	varchar(50)	No
fax	Participant's Fax	varchar(50)	No
position	Participant's position	varchar(100)	No
company_name	Participant's Company Name	varchar(255)	No
company_address	Participant's Company Address	varchar(255)	No
dateposted	Participant's Date Posted	datetime	No
status	Participant's Status	tinyint(4)	No

Table structure for table **registration**

Field	Description	Data Type	Null
regId	Registration ID	int(11)	No
reg_status	Registration Status	varchar(50)	No
reg_by	Registration Made By	varchar(150)	No
ORID	OR Id number	int	No
reg_balance	Registration Amount Balance	decimal(9,2)	No
remarks	Registration Remarks	text	No
dateposted	Registration Date Posted	datetime	No
status	Registration Status	tinyint(4)	No
trainingId	Registration Training ID	int(11)	No
participantId	Registration Participant ID	int(11)	No

Table structure for table **training**

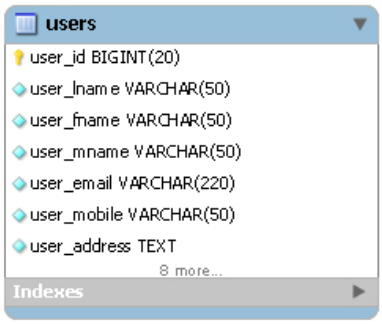
Field	Description	Data Type	Null
trainingId	Training ID	int(11)	No
regular_rate	Training Regular Rate	decimal(9,2)	No
early_bird_rate	Training Early Bird Date	decimal(9,2)	No
location	Training Location	varchar(255)	No
time	Training Time	varchar(50)	No
start_date	Training Start Date	date	No
end_date	Training End Date	date	No
max_participants	Training Maximum Participants	tinyint(4)	No
contact_person	Training Contact Person	varchar(150)	No
contact_phone	Training Contact Phone Number	varchar(50)	No
contact_mobile	Training Contact Mobile Number	varchar(50)	No
remarks	Training Remarks	text	No
dateposted	Training Date Posted	datetime	No
status	Training Status	tinyint(4)	No
courseId	Training Course ID	int(11)	No
facilitatorId	Training Facilitator ID	int(11)	No

Table structure for table **users**

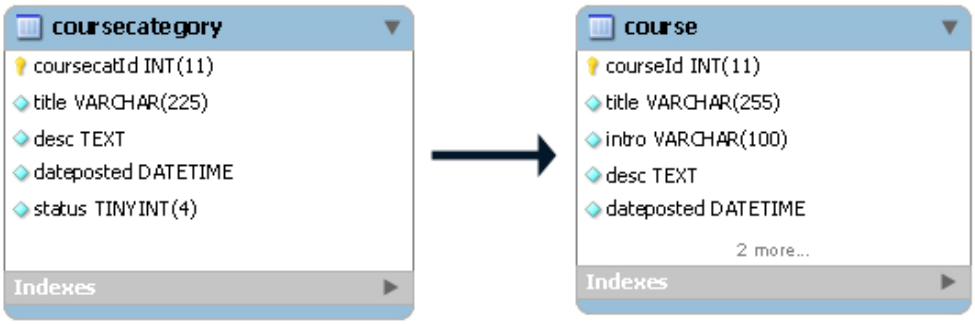
Field	Description	Data Type	Null
user_id	User's ID	bigint(20)	No
user_lname	User's Last Name	varchar(50)	No
user_fname	User's First Name	varchar(50)	No
user_mname	User's Middle Name	varchar(50)	No
user_email	User's Email Address	varchar(220)	No
user_mobile	User's Mobile Number	varchar(50)	No
user_address	User's Address	text	No
user_level	User's Level	tinyint(4)	No
username	User's Username	varchar(200)	No
pwd	User's Password	varchar(220)	No
ipadd	User's IP Address	varchar(200)	No
user_status	User's Record Status	tinyint(4)	No
user_datepost	User's Date Posted	datetime	No
ckey	Cache	varchar(220)	No
ctime	Session Cache	varchar(220)	No

2.6 Special Instructions: Data Process Flow

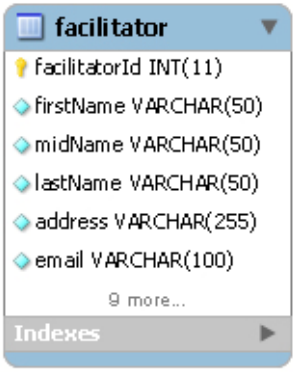
Step 1: Create User Account for CMS access



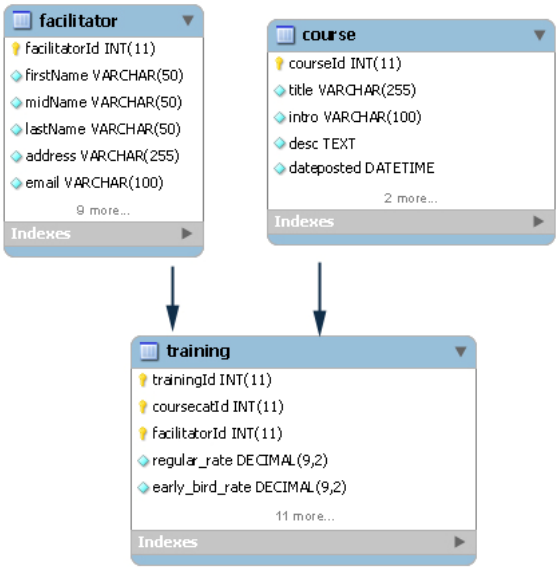
Step 2: Create Course



Step 3: Create Facilitator Profile



Step 4: Create Training Record



Step 5: Register Applicants

NOTE: Add Screen shots

3.0 DATABASE ADMINISTRATIVE INFORMATION

3.0 DATABASE ADMINISTRATIVE INFORMATION

3.1 System Information

Run using Xammp 1.7.0 for windows.

XAMPP 1.7.0 for Windows offers a collection of libraries and other applications useful for management and administration of a website, with all units that are essential for this. It includes, among other functions, PEAR, MiniPerl, mod_ssl, OpenSSL, PHPMyAdmin, Webalizer, FileZilla FTP Server, SQLite, Zend Optimizer, Mercury Mail, as well as a comprehensive control panel designed for XAMPP.

A web server, a MySQL database, PHP, an e-mail server, FTP server and Perl are the key elements of this distribution. Apache 2 and also includes the latest versions of MySQL and PHP, Apache and MySQL will be installed as services.

3.2 Database Management System (DBMS) Configuration

XAMPP 1.7.0

```
Apache 2.2.11
  + MySQL 5.1.30 (Community Server)
  + PHP 5.2.8 + PEAR(Support for PHP 4 has been discontinued)
  + PHP-Switch win32 1.0
    (use "php-switch.bat" in the xampp main directory)
  + XAMPP Control Version 2.5 from www.nat32.com
  + XAMPP Security 1.0
  + SQLite 2.8.15
  + OpenSSL 0.9.8i
  + phpMyAdmin 3.1.1
  + ADOdb 4.990
  + Mercury Mail Transport System v4.52
  + FileZilla FTP Server 0.9.29
  + Webalizer 2.01-10
  + Zend Optimizer 3.3.0
  + eAccelerator 0.9.5.3 for PHP 5.2.8
 (but not activated in the php.ini)
```

3.3 Hardware Configuration

- Download XAMPP 1.7.0
- Install XAMPP in the root of your hard drive. (C:XAMPP)
- Set up permission access

3.4 Database Software Utilities

- PhpMyadmin, SQLite & MySQL 5.1.30

3.5 Security

- ✓ Ensure these XAMPP pages are no longer accessible by the network for everyone.
- ✓ Set up MySQL admin user root password protection.
- ✓ Set up PhpMyAdmin password login is enabled.
- ✓ Database will be available via a local network only.

http://localhost/security/xamppsecurity.php

NOTE: Add also user permission on the system

NOTE: Add Network design

3.6 Storage Requirements

- Windows 98
- Windows ME
- Windows XP Home
- Windows NT
- Windows 2000
- Windows XP Professional (Recommended)
- 64 MB RAM (recommended)
- 200 MB free Fixed Disk

NOTE: Define server & client requirements

3.7 Recovery

- Export database monthly for updated back up

- Restore Data from MySQL dump file.

```
mysql -u #username# -p #database# < #dump_file#      -
```

- Write-ahead logging (WAL) and the recovery protocol are used to undo the actions of aborted transactions and to restore the system to a consistent state after a crash.

- Rollback to `SAVEPOINT` for selected modules

NOTE: This MySQL statement instructs the MySQL server to reverse SQL statements for the current transaction back to a point marked in the transaction by the MySQL statement, `SAVEPOINT`. Any transactions for the session made after the savepoint are undone.

This is in contrast to the MySQL statement, `ROLLBACK` by itself, which undoes all changes since the start of the transaction. Transaction statements are currently supported by the InnoDB, NDB Cluster, and BDB storage engines and are ignored if used with MyISAM tables. Multple savepoints may be set up during a transaction.

3.8 Error Handling

- System has error catcher define on each module.