|  |
| --- |
| **E:\My Documents\Desktop\Logo_FPT_University_doc.jpg** |
| Software Project Management Plan |
| E-Family |
|  |
| **By: E-Family Team – Supervisor: Mr. Thân Văn Sử** |
| **Ho Chi Minh City, 11th May, 2012** |

**Record of Changes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Changed Item** | **Description** | **By** | **Version** |
| May-11-2012 | All | Create the document | MinhPH | 0.1 |
| May-19-2012 | Header, Footer | Change header and footer | ThangQG | 0.2 |
| May-19-2012 | All | Modify content | ThangQG | 0.2 |
| May-21-2012 | Page Layout | Change page size to A4, change page margins | ThaiVNQ | 0.3 |
| May-21-2012 | Task Sheet | Add Task sheet: assignment and time table | ThaiVNQ | 0.3 |
| May-21-2012 | Meeting Minutes | Add meeting minutes | ThaiVNQ | 0.3 |
| May-22-2012 | Coding Convention | Add coding convention | ThaiVNQ | 0.4 |
| May-23-2012 | All | Spelling and grammar correction | ThangQG | 0.5 |
| May-23-2012 | Tasks | Add deployment risk | MinhPH | 0.5 |
| May-23-2012 | Page Layout | Change page margin to mirrored | ThaiVNQ | 0.5 |
| May-23-2012 | All | Finalize to version 1.0 | ThaiVNQ | 1.0 |

**Table of Contents**

[I. Problem Definition 4](#_Toc325576279)

[1. Name of this Capstone Project 4](#_Toc325576280)

[2. Problem Abstract 4](#_Toc325576281)

[3. Project Overview 4](#_Toc325576282)

[3.1. The Current System 4](#_Toc325576286)

[3.2. The Proposed System 5](#_Toc325576287)

[3.3. Boundaries of the System 5](#_Toc325576288)

[3.4. Development Environment 5](#_Toc325576289)

[II. Project Organization 6](#_Toc325576290)

[1. Software Process Model 6](#_Toc325576291)

[2. Roles and Responsibilities 7](#_Toc325576292)

[3. Tools and Techniques 7](#_Toc325576293)

[III. Project Management Plan 8](#_Toc325576294)

[1. Tasks 8](#_Toc325576295)

[1.1. Researching 8](#_Toc325576296)

[1.2. Planning 8](#_Toc325576297)

[1.3. Creating Software Requirement Specification 8](#_Toc325576298)

[1.4. Designing the Database 9](#_Toc325576299)

[1.5. Designing User Interface 9](#_Toc325576300)

[1.6. Creating Software Design Description 9](#_Toc325576301)

[1.7. Creating Coding Framework 9](#_Toc325576302)

[1.8. Implementation 10](#_Toc325576303)

[1.9. Performing System Test 10](#_Toc325576304)

[1.10. Deployment and Input Initial Data 10](#_Toc325576305)

[1.11. Creating Software User’s Manual 11](#_Toc325576306)

[2. Task Sheet: Assignment and Time Table 11](#_Toc325576307)

[3. All Meeting Minutes 15](#_Toc325576308)

[3.1 Kickoff Meeting May 7th 2012 15](#_Toc325576312)

[3.2 Meeting May 10th 2012 16](#_Toc325576317)

[3.3 Meeting May 14th 2012 16](#_Toc325576323)

[3.4 Meeting May 16th 2012 17](#_Toc325576330)

[3.5 Meeting May 18th 2012 18](#_Toc325576338)

[3.6 Meeting May 21st 2012 18](#_Toc325576347)

[3.7 Meeting May 22nd 2012 19](#_Toc325576357)

[IV. Coding Convention 19](#_Toc325576368)

[1. Naming Convention 19](#_Toc325576369)

[1.1. Capitalization Styles 19](#_Toc325576370)

[1.2. Class Naming Guideline 20](#_Toc325576373)

[1.3. Enumeration Type Naming Guideline 20](#_Toc325576374)

[1.4. Static Field Naming Guideline 21](#_Toc325576375)

[1.5. Parameter Naming Guideline 21](#_Toc325576376)

[1.6. Method Naming Guideline 21](#_Toc325576377)

[1.7. Variable Naming Guideline 21](#_Toc325576378)

[1.8. Control Naming Standard 22](#_Toc325576379)

[1.9. Variable Naming Guideline 22](#_Toc325576380)

[2. Code Formats 22](#_Toc325576381)

[2.1. Declarations 22](#_Toc325576382)

[2.2. Statements 23](#_Toc325576383)

[2.3. White Space 24](#_Toc325576384)

[3. Language Usage 24](#_Toc325576385)

[3.1. Object Lifecycle 24](#_Toc325576386)

[3.2. Control Flow 25](#_Toc325576387)

[3.3. Coding Style 25](#_Toc325576388)

[V. References 25](#_Toc325576389)

# Problem Definition

## Name of this Capstone Project

**Project full name:** E-Family

**Vietnamese name:** Gia Tộc

**Project code:** EFAM

**Released product name:** <http://e-fam.net>

## Problem Abstract

Because of tradition, most of the families in Vietnam have more than 3 generations; some families even have more than hundreds. But the lack of communication leads to the fact that many families are losing track of their ancestors and losing contact with their family members.

E-Family is a small part of the big picture that helps Vietnamese families to unite and keep track of their members. This website is a network for better communication inside the family.

## Project Overview



### The Current System

The most frequently used manner to keep track of family’s members and ancestors is drawing family tree.

People have done it since many years ago with pens and papers but this way is inconvenient: hard to store, hard to update and easy to lose.

There are some applications that support manage and draw family. But the scope of these applications limits to share information.

⇨ So E-Family website is in need. With a website, everyone can easy to access and view information. The only website in Vietnam that can support the need is phahe.vn. But it still has some limitations:

* The user interface is not friendly. There is too much information on screen that makes viewer hard to look for what they need.
* The family tree function is not good:
* Error sometime occurs when the family has too many generations.
* The graphic is based on folder tree technique and it is not ease with the eyes.
* Missing some functions: such as join two family trees…

### The Proposed System

The new system will have ability to:

1. Create and draw family tree
2. Update the family tree easily
3. Join two family trees
4. Manage and store family’s history
5. Support family to manage the announcement, article and event…
6. Provide a way for admin of family to manage their members
7. Provide a friendly way for family members to communicate with each other

### Boundaries of the System

* Vietnamese user-interface
* Only focus on Vietnamese culture and Vietnamese people
* Based on web application
* Family tree algorithm only supports upper and lower 2 generations of a person
* Not support adoption.
* Lunar calendar is not available in this version

### Development Environment

***Hardware:***

* Personal computers for developing with the minimum configuration: CPU Core 2 Duo 2.0GHz, 2GB of RAM, 120GB of hard disk, and internet connection with 6Mbps download and 640Kbps upload speed.
* A Windows hosting for publishing and testing with the minimum configuration: 800MB capacity, 15GB/month bandwidth.

***Software:***

* Microsoft Windows 7: operating system and platform for development
* Microsoft Visual Studio 2010 SP1: used to code system modules
* Microsoft SQL Server 2008: used to create and manage the database for system
* Google Code and TortoiseSVN: used to control source code and documents of project
* IIS 7: web server
* StarUML and Microsoft Visio 2010: used to create models and diagrams
* Microsoft Project 2010: used to manage process and work schedules.
* Skype: used for communication and meeting

# Project Organization

## Software Process Model

This project uses **waterfall model** [1] as its development process model.

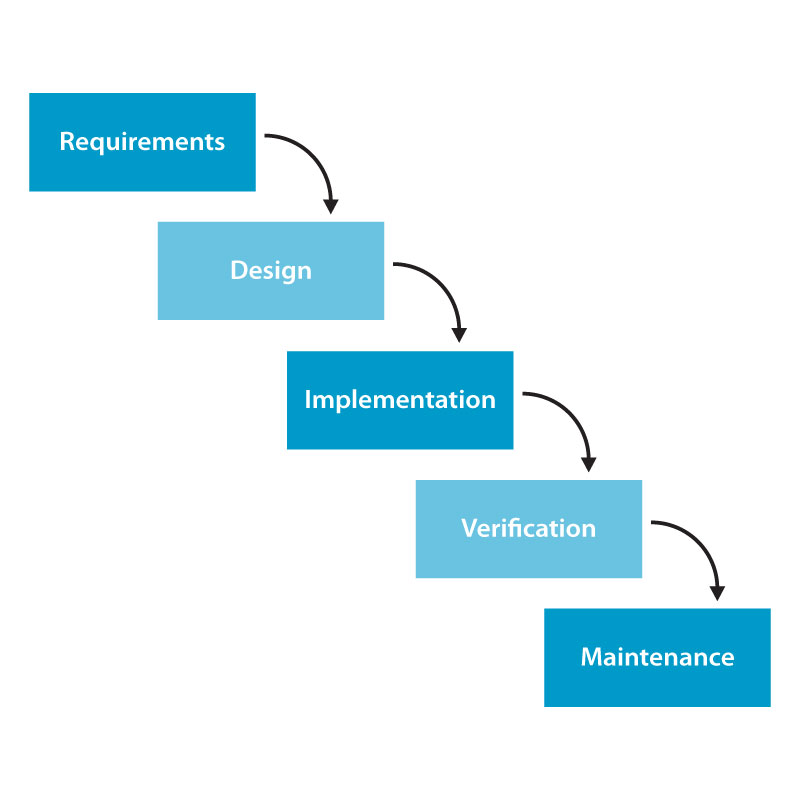


FIGURE II‑1 Waterfall model

All of the phases in the waterfall models are included in the scope of this project and are the responsibilities of the team:

**Requirement:** During this phase research is being conducted which includes brainstorming about the software, what it is going to be and what purpose it is going to fulfill.

**Design:** This step involves formulating the basic design of the software on paper. After the basic design gets approved, then a more elaborated technical design can be planned. Here the functions of each of the part are decided and the engineering units are placed for example modules, etc.

**Implementation:** In this phase the source code of the programs is written.

**Verification:** At this phase, the whole design and its construction is put under a test to check its functionality.

**Maintenance:** Maintenance is needed to ensure that the system will continue to perform as desired.

## Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibilities** |
| **Thân Văn Sử** | Supervisor | * Give advice on cultural considerations and technical problems * Review and approve project elements |
| **Võ Nguyễn Quốc Thái** | Team Leader | * Control process * Ensure project going in the right way and will achieve its objectives * Research * Do detailed design * Develop * Test |
| **Đặng Hồ Đăng** | Team Member | * Research * Do high-level system design * Do detailed design * Develop * Test |
| **Phạm Hoàng Minh** | Team Member | * Research * Design user interface * Do detailed design * Develop * Test |
| **Quách Gia Thắng** | Team Member | * Research * Manage documents * Do detailed design * Develop * Test |

TABLE II‑1 Roles and Responsibilities

## Tools and Techniques

Tools: Visual Studio 2010, JetBrain ReSharper 6, SQL Server 2008, Notepad++, IIS 7

Front-end technologies: ASP.NET 4.0, jQuery, AJAX

Programming languages: C#, Javascript

Architecture: 3-tier

Object-Relational Mapper: NHibernate

Design patterns: Repository, data transfer object

# Project Management Plan

## Tasks

### Researching

**Description:** Do cultural research; current systems research; user’s need research; and technology research

**Deliverables:** Project Introduction Report before May 17th 2012

**Resources Needed:** All team members; 8 days

**Dependencies and Constraints:** Follow Information System Capstone Project Document Template

**Risks:** Culture is complicated and researchers are sometimes biased ⇨ it’s not easy to have a good research result.

### Planning

**Description:** Divide modules, create work breakdown structure and plan tasks (duration, predecessors, resources, etc.)

**Deliverables:** Software Project Management Plan Report before May 24th 2012

**Resources Needed:** All team members; 5 days

**Dependencies and Constraints:** Finish researching; follow Information System Capstone Project Document Template

**Risks:** Modules dividing and task planning maybe unbalance

### Creating Software Requirement Specification

**Description:** Create all the documents that necessary to describe the system and write system test cases

**Deliverables:** Software Requirement Specification before June 3rd 2012

**Resources Needed:** All team members; 6 days

**Dependencies and Constraints:** Finish planning; follow Information System Capstone Project Document Template

**Risks:** Missing user requirement may lead to critical fault in designing and developing process; insufficient system test cases may lead to wrong system test results

### Designing the Database

**Description:** Create conceptual, logical and physical database design

**Deliverables:** Database design and database script, deliver with Software Design Description before June 14th 2012

**Resources Needed:** Võ Nguyễn Quốc Thái; 4 days

**Dependencies and Constraints:** Finish the requirement specification task

**Risks:** SRS may not be detailed enough to capture the business rules, causing the database design to be inappropriate or will be changed much in the future

### Designing User Interface

**Description:** Define styles that will be applied to the user interface, design layouts

**Deliverables:** HTML pages and CSS files

**Resources Needed:** Phạm Hoàng Minh; 4 days

**Dependencies and Constraints:** Finish the requirement specification task

**Risks:** Inappropriate interface may lead to user unpleasant and the final product may not success as expected

### Creating Software Design Description

**Description:** Do architecture design and detailed design

**Deliverables:** Software Design Description Report before June 14th 2012

**Resources Needed:** Võ Nguyễn Quốc Thái and Phạm Hoàng Minh: 4 days, Đặng Hồ Đăng and Quách Gia Thắng: 8 days

**Dependencies and Constraints:** Finish the requirement specification task; follow Information System Capstone Project Document Template

**Risks:** Choosing inappropriate architecture and design patterns may cause the system hard to maintain, cost much coding efforts, and slow down system performance

### Creating Coding Framework

**Description:** Map the architecture design into source code, create the project solution files and common classes, and implement common functions

**Deliverables:** ASP.NET project and solution files

**Resources Needed:** Đặng Hồ Đăng: 5 days

**Dependencies and Constraints:** Finish design description task; follow coding convention

**Risks:** Inappropriate coding framework leads to hard implementation and maintenance

### Implementation

**Description:** Implement the system as designed

**Deliverables:** The whole website of E-Family project

**Resources Needed:** All team members: 22 days, especially Đặng Hồ Đăng: 17 days

**Dependencies and Constraints:** Finish design description task, follow coding convention

**Risks:** Behind schedule; unit testing may not be performed thoroughly causing spending many efforts in system test phase

### Performing System Test

**Description:** Do system test, includes function test and performance test

**Deliverables:** Software Test Documentation before July 19th 2012

**Resources Needed:** All team members: 4 days

**Dependencies and Constraints:** Finish implementation; follow Information System Capstone Project Document Template

**Risks:** Developers are also responsible for system testing, this may lead to compromise

### Deployment and Input Initial Data

**Description:** Deploy the system to the Internet and input initial data for the system before going live

**Deliverables:** Complete E-Family website ([www.e-fam.net](http://www.e-fam.net)), project source code, up-to-date documentations

**Resources Needed:** All team members: 7 days

**Dependencies and Constraints:** Finish system testing

**Risks:** There’re may be problems with the host server.

### Creating Software User’s Manual

**Description:** Create user’s manual

**Deliverables:** Software User’s Manual before August 18th 2012

**Resources Needed:** All team members: 14 days

**Dependencies and Constraints:** Finish deployment; follow Information System Capstone Project Document Template

**Risks:** User manual may not clear and detailed enough leads user to difficulty when using our system

## Task Sheet: Assignment and Time Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Task Name*** | ***Duration*** | ***Start*** | ***Finish*** | ***Resource Names*** |
| **Initiating** | **1 day** | **Mon 5/7/12** | **Mon 5/7/12** |  |
| Identify key stakeholders | 4 hours | Mon 5/7/12 | Mon 5/7/12 | All |
| Hold project kick-off meeting | 4 hours | Mon 5/7/12 | Mon 5/7/12 | All |
| **Researching** | **8 days** | **Tue 5/8/12** | **Thu 5/17/12** |  |
| Cultural Research | 2 days | Tue 5/8/12 | Wed 5/9/12 | All |
| Current Systems Research | 2 days | Thu 5/10/12 | Fri 5/11/12 | All |
| User's Needs Research | 2 days | Mon 5/14/12 | Tue 5/15/12 | All |
| Technology Research | 2 days | Wed 5/16/12 | Thu 5/17/12 | All |
| ***Project Introduction Report*** | ***0*** | ***Thu 5/17/12*** | |  |
| **Planning** | **5 days** | **Fri 5/18/12** | **Thu 5/24/12** |  |
| Prepare team contract | 4 hours | Fri 5/18/12 | Fri 5/18/12 | All |
| ***Team Contract*** | ***0*** | ***Fri 5/18/12*** | |  |
| Create Work Breakdown Structure | 1 day | Fri 5/18/12 | Mon 5/21/12 | MinhPH |
| ***Work Breakdown Structure*** | ***0*** | ***Mon 5/21/12*** | |  |
| **Prepare schedule** | **1.5 days** | **Mon 5/21/12** | **Tue 5/22/12** |  |
| Determine task resources | 4 hours | Mon 5/21/12 | Mon 5/21/12 | All |
| Determine task durations | 2 hours | Tue 5/22/12 | Tue 5/22/12 | All |
| Determine task dependencies | 2 hours | Tue 5/22/12 | Tue 5/22/12 | DangDH |
| Create draft Gantt chart | 2 hours | Tue 5/22/12 | Tue 5/22/12 | ThangQG |
| Review and finalize Gantt chart | 2 hours | Tue 5/22/12 | Tue 5/22/12 | All |
| ***Gantt chart*** | ***0*** | ***Tue 5/22/12*** | |  |
| Develop final project management plan | 2 days | Wed 5/23/12 | Thu 5/24/12 | ThaiVNQ |
| ***Software Project Management Plan*** | ***0*** | ***Thu 5/24/12*** | |  |
| **Executing** | **61 days** | **Fri 5/25/12** | **Fri 8/17/12** |  |
| **Getting requirement** | **6 days** | **Fri 5/25/12** | **Fri 6/1/12** |  |
| Define user requirements | 1 day | Fri 5/25/12 | Fri 5/25/12 | All |
| Define system requirements | 4 days | Mon 5/28/12 | Thu 5/31/12 | All |
| Define non-functional requirements | 4 hours | Fri 6/1/12 | Fri 6/1/12 | DangDH,  ThangQG |
| Complete Software Requirement Specification | 4 hours | Fri 6/1/12 | Fri 6/1/12 | ThaiVNQ,  MinhPH |
| ***Software Requirement Specification*** | ***0*** | ***Fri 6/1/12*** | |  |
| Write system test cases | 1 day | Mon 6/4/12 | Mon 6/4/12 | All |
| ***System Test Cases Document*** | ***0*** | ***Mon 6/4/12*** | |  |
| **Designing** | **8 days** | **Mon 6/4/12** | **Wed 6/13/12** |  |
| Perform architecture design | 2 days | Mon 6/4/12 | Tue 6/5/12 | DangDH,  ThangQG |
| **Design Database** | **4 days** | **Mon 6/4/12** | **Thu 6/7/12** | **ThaiVNQ** |
| Create conceptual database design | 1 day | Mon 6/4/12 | Mon 6/4/12 | ThaiVNQ |
| Create logical database design | 1.5 days | Tue 6/5/12 | Wed 6/6/12 | ThaiVNQ |
| Create physical database design | 1.5 days | Wed 6/6/12 | Thu 6/7/12 | ThaiVNQ |
| ***Database design and database script*** | ***0*** | ***Thu 6/7/12*** | |  |
| **Design User Interface** | **4 days** | **Mon 6/4/12** | **Thu 6/7/12** | **MinhPH** |
| Design global master page | 4 hours | Mon 6/4/12 | Mon 6/4/12 | MinhPH |
| Design family master page | 4 hours | Mon 6/4/12 | Mon 6/4/12 | MinhPH |
| Design detailed pages | 2.5 days | Tue 6/5/12 | Thu 6/7/12 | MinhPH |
| Define page flows | 4 hours | Thu 6/7/12 | Thu 6/7/12 | MinhPH |
| ***All pages' user interface and page flow document*** | ***0*** | ***Thu 6/7/12*** | |  |
| Perform detailed design | 5 days | Wed 6/6/12 | Tue 6/12/12 | All |
| Write unit test cases | 1 day | Wed 6/13/12 | Wed 6/13/12 | All |
| ***Unit Test Cases Documents*** | ***0*** | ***Wed 6/13/12*** | |  |
| Create Software Design Description (SDD) document | 4 hours | Wed 6/13/12 | Wed 6/13/12 | All |
| ***Software Design Description Document*** | ***0*** | ***Wed 6/13/12*** | |  |
| **Development** | **22 days** | **Thu 6/14/12** | **Fri 7/13/12** |  |
| Create coding framework | 5 days | Thu 6/14/12 | Wed 6/20/12 | DangDH |
| ***ASP.NET project and solution files*** | ***0*** | ***Wed 6/20/12*** | |  |
| **Develop Family Relationship Module** | **22 days** | **Thu 6/14/12** | **Fri 7/13/12** | **ThaiVNQ** |
| Code show family tree of a person function | 12 days | Thu 6/14/12 | Fri 6/29/12 | ThaiVNQ |
| Code join family function | 2 days | Mon 7/2/12 | Tue 7/3/12 | ThaiVNQ |
| Code leave family function | 2 days | Wed 7/4/12 | Thu 7/5/12 | ThaiVNQ |
| Code manage family relationship function | 6 days | Fri 7/6/12 | Fri 7/13/12 | ThaiVNQ |
| **Develop Time-Related Module** | **17 days** | **Thu 6/21/12** | **Fri 7/13/12** | **DangDH** |
| Code create event function | 3 days | Thu 6/21/12 | Mon 6/25/12 | DangDH |
| Code view event function | 3 days | Tue 6/26/12 | Thu 6/28/12 | DangDH |
| Code update event function | 3 days | Fri 6/29/12 | Tue 7/3/12 | DangDH |
| Code delete event function | 3 days | Wed 7/4/12 | Fri 7/6/12 | DangDH |
| Code Annals page | 5 days | Mon 7/9/12 | Fri 7/13/12 | DangDH |
| **Develop Communication Module** | **22 days** | **Thu 6/14/12** | **Fri 7/13/12** | **MinhPH** |
| Code family's chat box | 4 days | Thu 6/14/12 | Tue 6/19/12 | MinhPH |
| Code send private message function | 1 day | Wed 6/20/12 | Wed 6/20/12 | MinhPH |
| Code view private messages function | 1.5 days | Thu 6/21/12 | Fri 6/22/12 | MinhPH |
| Code create announcement function | 2.5 days | Fri 6/22/12 | Tue 6/26/12 | MinhPH |
| Code edit announcement function | 1 day | Wed 6/27/12 | Wed 6/27/12 | MinhPH |
| Code activate announcement function | 4 hours | Thu 6/28/12 | Thu 6/28/12 | MinhPH |
| Code de-activate announcement function | 4 hours | Thu 6/28/12 | Thu 6/28/12 | MinhPH |
| Code delete announcement function | 1 day | Fri 6/29/12 | Fri 6/29/12 | MinhPH |
| Code create article function | 3 days | Mon 7/2/12 | Wed 7/4/12 | MinhPH |
| Code view article function | 1 day | Thu 7/5/12 | Thu 7/5/12 | MinhPH |
| Code update article function | 1 day | Fri 7/6/12 | Fri 7/6/12 | MinhPH |
| Code delete article function | 1 day | Mon 7/9/12 | Mon 7/9/12 | MinhPH |
| Code approve article function | 1 day | Tue 7/10/12 | Tue 7/10/12 | MinhPH |
| Code comment function | 3 days | Wed 7/11/12 | Fri 7/13/12 | MinhPH |
| **Develop Management Module** | **22 days** | **Thu 6/14/12** | **Fri 7/13/12** | **ThangQG** |
| Code register function | 1 day | Thu 6/14/12 | Thu 6/14/12 | ThangQG |
| Code login, logout function | 4 hours | Fri 6/15/12 | Fri 6/15/12 | ThangQG |
| Code view user account's information function | 3 days | Fri 6/15/12 | Wed 6/20/12 | ThangQG |
| Code edit user account's information function | 1 day | Wed 6/20/12 | Thu 6/21/12 | ThangQG |
| Code view a person's information function | 4 days | Thu 6/21/12 | Wed 6/27/12 | ThangQG |
| Code edit a person's information function | 2 days | Wed 6/27/12 | Fri 6/29/12 | ThangQG |
| Code request to create family function | 1 day | Fri 6/29/12 | Mon 7/2/12 | ThangQG |
| Code approve join member function | 1 day | Mon 7/2/12 | Tue 7/3/12 | ThangQG |
| Code create family function | 2.5 days | Tue 7/3/12 | Thu 7/5/12 | ThangQG |
| Code activate family function | 1 day | Fri 7/6/12 | Fri 7/6/12 | ThangQG |
| Code de-activate family function | 4 hours | Mon 7/9/12 | Mon 7/9/12 | ThangQG |
| Code configure family home page function | 2.5 days | Mon 7/9/12 | Wed 7/11/12 | ThangQG |
| Code manage admins function | 1 day | Thu 7/12/12 | Thu 7/12/12 | ThangQG |
| Code manage super-admins function | 1 day | Fri 7/13/12 | Fri 7/13/12 | ThangQG |
| ***The whole website of E-Family project*** | ***0*** | ***Fri 7/13/12*** | |  |
| **Testing** | **4 days** | **Mon 7/16/12** | **Thu 7/19/12** |  |
| Perform system test | 4 days | Mon 7/16/12 | Thu 7/19/12 | All |
| ***Software Test Documentation*** | ***0*** | ***Thu 7/19/12*** | |  |
| **Deployment & Input Initial Data** | **7 days** | **Fri 7/20/12** | **Mon 7/30/12** |  |
| Deploy system to the Internet | 1 day | Fri 7/20/12 | Fri 7/20/12 | ThaiVNQ |
| Input initial data | 6 days | Mon 7/23/12 | Mon 7/30/12 | All |
| **Create User's Manual** | **14 days** | **Tue 7/31/12** | **Fri 8/17/12** |  |
| Create user's manual | 14 days | Tue 7/31/12 | Fri 8/17/12 | All |
| ***Software User’s Manual*** | ***0*** | ***Fri 8/17/12*** | |  |
| **Monitoring and Controlling** | **75 days** | **Mon 5/7/12** | **Fri 8/17/12** |  |
| Monitor and control project work | 75 days | Mon 5/7/12 | Fri 8/17/12 | ThaiVNQ  [3%] |
| Perform scope control | 75 days | Mon 5/7/12 | Fri 8/17/12 | ThaiVNQ  [3%] |
| Perform schedule control | 75 days | Mon 5/7/12 | Fri 8/17/12 | ThaiVNQ  [3%] |
| Perform change control | 75 days | Mon 5/7/12 | Fri 8/17/12 | ThaiVNQ  [3%] |
| Perform quality control | 75 days | Mon 5/7/12 | Fri 8/17/12 | ThaiVNQ  [3%] |
| **Closing** | **5 days** | **Mon 8/20/12** | **Fri 8/24/12** |  |
| Prepare final project presentation | 4 days | Mon 8/20/12 | Thu 8/23/12 | All |
| Deliver final report and presentation | 1 day | Fri 8/24/12 | Fri 8/24/12 | All |
| ***Capstone Project Completed*** | ***0*** | ***Fri 8/24/12*** | |  |

TABLE III‑1 Task sheet

## All Meeting Minutes



### Kickoff Meeting May 7th 2012

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Kickoff Meeting | **Date** | May-07-2012 |
| **Facilitator** | FPT University HCMC Library | **Time** | 12:00 – 16:00 |
| **Location** | Innovation Building, Quang Trung Software City, District 12, HCMC | **Scribe** | ThaiVNQ |
| **Attendees** | ThaiVNQ (Team Leader), DangDH, MinhPH, ThangQG | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1 | Team member introduction | All team members introduce together |
| 2 | Project introduction | Each team member tell their view on the project |
| 3 | Team member strengths and weaknesses | All team members talk about their strengths and weaknesses on doing this project |
| 4 | Team member working time | All team members talk about what time they can work on this project |
| 5 | Kick-off party | Make all team member closer |

| **Action Plan** | | | |
| --- | --- | --- | --- |
| No. | Action Item(s) | Owner | Target Date |
| 1 | Do Vietnamese culture research | All | May-10-2012 |
| 2 | Prepare Report 1: Introduction | ThangQG | May-10-2012 |



### Meeting May 10th 2012

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Finish cultural research | **Date** | May-10-2012 |
| **Facilitator** | FPT University HCMC Library | **Time** | 14:00 – 16:00 |
| **Location** | Innovation Building, Quang Trung Software City, District 12, HCMC | **Scribe** | MinhPH |
| **Attendees** | ThaiVNQ (Team Leader), DangDH, MinhPH, ThangQG | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1 | Vietnamese culture | Discuss about traditional and modern Vietnam families, relationship among generations and among smaller-families |
| 2 | Introduce current systems research | Discuss about current family-related information systems in Vietnam, assign some systems to team members |

| **Action Plan** | | | |
| --- | --- | --- | --- |
| No. | Action Item(s) | Owner | Target Date |
| 1 | Do current systems research | All | May-14-2012 |
| 2 | Prepare Report 1: Introduction | ThangQG | May-14-2012 |



### Meeting May 14th 2012

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Finish current system research | **Date** | May-14-2012 |
| **Facilitator** | Skype | **Time** | 20:00 – 21:00 |
| **Location** | At each team member’s residence | **Scribe** | MinhPH |
| **Attendees** | ThaiVNQ (Team Leader), DangDH, MinhPH, ThangQG | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1 | Current systems | Discuss about current family-related information systems in Vietnam, what are their strengths and weaknesses |
| 2 | Opportunities | What are our opportunities in this field |
| 3 | Domain name | Discuss about what domain name is appropriate for this project |

| **Action Plan** | | | |
| --- | --- | --- | --- |
| No. | Action Item(s) | Owner | Target Date |
| 1 | Do user’s needs and technical research | All | May-16-2012 |
| 2 | Prepare Report 1: Introduction | ThangQG | May-16-2012 |
| 3 | Buy domain name and hosting | ThaiVNQ | May-16-2012 |



### Meeting May 16th 2012

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Finish user’s needs and technical research | **Date** | May-16-2012 |
| **Facilitator** | Skype | **Time** | 20:00 – 21:30 |
| **Location** | At each team member’s residence | **Scribe** | MinhPH |
| **Attendees** | ThaiVNQ (Team Leader), DangDH, MinhPH, ThangQG | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1 | User’s need | Discuss about user’s demand on family-related information system, what they really need and what they don’t need |
| 2 | Technologies | Discuss what tools and technologies we will use to develop this system |
| 3 | Review Report 1 | All team review report 1 |

| **Action Plan** | | | |
| --- | --- | --- | --- |
| No. | Action Item(s) | Owner | Target Date |
| 1 | Finish Report 1: Introduction | ThangQG | May-17-2012 |
| 2 | Prepare Report 2: Introduction | ThangQG | May-18-2012 |



### Meeting May 18th 2012

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Discuss team contract and prepare work breakdown structure | **Date** | May-18-2012 |
| **Facilitator** | Skype | **Time** | 20:00 – 21:30 |
| **Location** | At each team member’s residence | **Scribe** | MinhPH |
| **Attendees** | ThaiVNQ (Team Leader), DangDH, MinhPH, ThangQG | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1 | Team contract | Define the ground rules of the team |
| 2 | Work breakdown structure | Discuss about assigning tasks |

| **Action Plan** | | | |
| --- | --- | --- | --- |
| No. | Action Item(s) | Owner | Target Date |
| 1 | Create work breakdown structure | MinhPH | May-21-2012 |



### Meeting May 21st 2012

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Prepare schedule | **Date** | May-21-2012 |
| **Facilitator** | Skype | **Time** | 20:00 – 22:00 |
| **Location** | At each team member’s residence | **Scribe** | ThaiVNQ |
| **Attendees** | ThaiVNQ (Team Leader), DangDH, MinhPH, ThangQG | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1 | Schedule | Discuss task resources, dependences and durations |

| **Action Plan** | | | |
| --- | --- | --- | --- |
| No. | Action Item(s) | Owner | Target Date |
| 1 | Create Gantt chart | ThangQG | May-22-2012 |
| 2 | Prepare Report 2: Software Project Management Plan | ThaiVNQ | May-22-2012 |



### Meeting May 22nd 2012

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Finalize Gantt chart | **Date** | May-22-2012 |
| **Facilitator** | FPT University HCMC Self Study Area | **Time** | 14:00 – 15:30 |
| **Location** | Innovation Building, Quang Trung Software City, District 12, HCMC | **Scribe** | ThaiVNQ |
| **Attendees** | ThaiVNQ (Team Leader), DangDH, MinhPH, ThangQG | | |

| **Key Points Discussed** | | |
| --- | --- | --- |
| No. | Topic | Highlights |
| 1 | Gantt chart | Discuss and finalize the Gantt chart |

| **Action Plan** | | | |
| --- | --- | --- | --- |
| No. | Action Item(s) | Owner | Target Date |
| 1 | Write Report 2: Software Project Management Plan | ThaiVNQ | May-24-2012 |
| 2 | Prepare User Requirements | DangDH, MinhPH, ThangQG | May-25-2012 |



# Coding Convention

The following conventions are taken from FPT Software, Microsoft and CSharp-Friends.com.

## Naming Convention

### Capitalization Styles

**Pascal case:** The first letter in the identifier and the first letter of each subsequent concatenated word are capitalized. You can use Pascal case for identifiers of three or more characters. For example:

|  |
| --- |
| ArticleComment |

**Camel case:** The first letter of an identifier is lowercase and the first letter of each subsequent concatenated word is capitalized. For example:

|  |
| --- |
| homePhone |

The following table summarizes the capitalization rules and provides examples for the different types of identifiers.

|  |  |  |
| --- | --- | --- |
| **Identifier** | **Case** | **Example** |
| Class | Pascal | PersonalHistory |
| Enum Type | Pascal | RoleName |
| Enum Value | Pascal | SuperAdmin |
| Exception class | Pascal | DevideByZeroException  *Note: Always ends with Exception* |
| Read-only static field | Pascal | SessionFactory |
| Interface | Pascal | IEventRepository *Note: Always begins with I* |
| Method | Pascal | CreateFamily |
| Namespace | Pascal | System.Web |
| Parameter | Camel | firstName |
| Property | Pascal | PermanentAddress |



### Class Naming Guideline

The following rules outline the guidelines for naming classes:

* Use a noun or noun phrase to name a class.
* Use Pascal case.
* Do not use a type prefix, such as C for class, on a class name. For example, use the class name PersonalHistory rather than CPersonalHistory.
* Do not use the underscore character (\_).

The following are examples of correctly named classes:

|  |
| --- |
| public class **PersonalHistory**  public class **Family**  public class **Event** |

### Enumeration Type Naming Guideline

The following rules outline the naming guidelines for enumerations:

* Use Pascal case for enumeration (Enum) types and value names.
* Do not use an Enum suffix on Enum type names.
* Use a singular name for most Enum types. For example, do not name an enumeration type **Roles** but name it **Role** instead.

### Static Field Naming Guideline

The following rules outline the naming guidelines for static fields:

* Use nouns, noun phrases, or abbreviations of nouns to name static fields.
* Use Pascal case.
* Do not use a Hungarian notation prefix on static field names.
* It is recommended that you use static properties instead of public static fields whenever possible.

### Parameter Naming Guideline

It is important to carefully follow these parameter naming guidelines because visual design tools that provide context sensitive help and class browsing functionality display method parameter names to users in the designer. The following rules outline the naming guidelines for parameters:

* Use camel case for parameter names.
* Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
* Use names that describe a parameter's meaning rather than names that describe a parameter's type.
* Do not prefix parameter names with Hungarian type notation.

### Method Naming Guideline

The following rules outline the naming guidelines for methods:

* Use verbs or verb phrases to name methods.
* Use Pascal case.

The following are examples of correctly named classes:

|  |
| --- |
| CreateArticle();  GetFirstName();  Decrypt(); |

### Variable Naming Guideline

The following rules outline the naming guidelines for variables:

* Use a noun or noun phrase to name variables.
* Use Camel case.
* Use i, j, k for counting variables.

### Control Naming Standard

The following rules outline the naming standards for controls:

|  |  |  |
| --- | --- | --- |
| **Control Type** | **Prefix** | **Example** |
| Check box | chk | chkApprove |
| Combo box, drop-down list box | cbo | cboNationality |
| Command button | btn | btnCancel |
| Common dialog | dlg | dlgFileUpload |
| Date picker | dtp | dtpBirth |
| Label | lbl | lblErrorMessage |
| Rich textbox | rtf | rtfDescription |
| Text box | txt | txtFirstName |
| Tree view | tre | treNavigation |

### Variable Naming Guideline

To clearly distinguish constants from other elements, use all uppercase when naming them.

An underscore can be used to separate terms when necessary.

Example:

|  |
| --- |
| public const int **MAX\_UPPER\_GENERATION** = 3; |

## Code Formats

### Declarations

**Number Per Line:** One declaration per line is recommended since it encourages commenting. In other words,

|  |
| --- |
| private int year = 2012; // year of birth  private int age = 12; // this person’s age |

**Initialization:** Try to initialize local variables where they're declared. The only reason not to initialize a variable where it's declared is if the initial value depends on some computation occurring first.

**Placement:** Put declarations only at the beginning of blocks. (A block is any code surrounded by curly braces "{" and "}".) Don't wait to declare variables until their first use; it can confuse the unwary programmer and hamper code portability within the scope.

The one exception to the rule is indexes of **for** loops, which in C# can be declared in the **for** statement:

|  |
| --- |
| for (int i = 0; i < n; ++i)  {  // Do something  } |

**Class and Interface Declarations:** When coding C# classes and interfaces, the following formatting rules should be followed:

* No space between a method name and the parenthesis "(" starting its parameter list
* Open brace "{" appears at the beginning of the line following declaration statement and is indented to the beginning of the declaration.
* Closing brace "}" starts a line by itself indented to match its corresponding opening statement.
* For null statements, the "}" should appear immediately after the "{" and both braces should appear on the same line as the declaration with 1 blank space separating the parentheses from the braces.

### Statements

**Single Statements:** Each line should contain at most one statement.

**Compound Statements:** Compound statements are statements that contain lists of statements enclosed in braces “{statements}”.

* The enclosed statements should be indented one more level than the compound statement.
* The opening brace should be at the beginning of the line following the line that begins the compound statement and be indented to the beginning of the compound statement.
* The closing brace should begin a line and be indented to the beginning of the compound statement.
* Braces are used around all statements, even single statements, when they are part of a control structure, such as an if-else or for statement. This makes it easier to add statements without accidentally introducing bugs due to forgetting to add braces.

### White Space

**Blank Lines:** Blank lines improve readability by setting off sections of code that are logically related. One blank line should always be used in the following circumstances:

* Between the local variables in a method and its first statement.
* Between logical sections inside a method to improve readability.
* After the closing brace of a code block that is not followed by another closing brace.

**Blank Spaces:** Blank spaces should be used in the following circumstances:

* A keyword followed by a parenthesis should be separated by a space. Example:

|  |
| --- |
| while (true)  {  // Do something  } |

* Note that a blank space should not be used between a method name and its opening parenthesis.
* A blank space should appear after commas in argument lists.
* All binary operators except “.” should be separated from their operands by spaces.
* Blank spaces should never separate unary operators such as unary minus, increment ("++"), and decrement ("--") from their operands.
* The expressions in a ‘**for’** statement should be separated by blank spaces. Example:

|  |
| --- |
| for (expr1; expr2; expr3) |

## Language Usage

### Object Lifecycle

* Declare and initialize variables close to where they are used.
* If possible, initialize variables at the point of declaration.
* Declare each variable in a separate declaration statement.
* Use a public static read-only field to define predefined object instances.
* Set a reference field to null to tell the Garbage Collector that the object is no longer needed.
* Set a reference field to *null* to tell the GC that the object is no longer needed.

### Control Flow

* Do not change a loop variable inside a *for* loop block.
* All flow control primitives (*if, else, while, for, do, switch*) shall be followed by a block, even if it is empty.
* All *switch* statements shall have a *default* label as the last *case* label.
* An *else* sub-statement of an ‘*if’* statement shall not be an ‘*if’* statement without an *else* part.
* Do not make explicit comparisons to *true* or *false*.
* Do not use selection statements (*if, switch*) instead of a simple assignment or initialization.

### Coding Style

* The *public*, *protected*, and *private* sections of a *class* shall be declared in that order.
* Use spaces instead of tabs.
* Do not create source lines longer than 80 characters.

# References

[1] Waterfall model: <http://en.wikipedia.org/wiki/Waterfall_model>

* C# coding convention:

<http://msdn.microsoft.com/en-us/library/ff926074.aspx>; <http://www.csharpfriends.com/articles/getarticle.aspx?articleid=336>