**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Job Searching System**

|  |  |
| --- | --- |
| **Group JSS** | |
| **Group members** | Huỳnh Đức Tài – Leader – SE60630  Nguyễn Ngọc Thiện – Member – SE90129  Phạm Quốc Thái – Member – SE60599  Võ Hoàn Thiện – Member – SE60788  Trịnh Văn Hoàng – Member – 01769  Trần Chung Thành – Member – 01611 |
| **Supervisor** | Mr. Trần Đình Trí |
| **Ext. Supervisor** | N/A |
| **Project code** | JSS |

-Hanoi, 05/2015-

# Table of Contents

[Table of Contents 2](#_Toc425867510)

[List of Tables 5](#_Toc425867511)

[Definition, Acronyms and Abbreviations 5](#_Toc425867512)

[2. Project Management Plan 6](#_Toc425867513)

[2.1. Problem Definition 6](#_Toc425867514)

[2.1.1. Name of this Capstone Project 6](#_Toc425867515)

[2.1.2. Problem Abstract 6](#_Toc425867516)

[2.1.3. Project Overview 7](#_Toc425867517)

[2.1.3.1. The Current System 7](#_Toc425867518)

[2.1.3.2. The Proposed System 7](#_Toc425867519)

[2.1.3.3. Boundaries of the System 7](#_Toc425867520)

[2.1.3.4. Development Environment 7](#_Toc425867521)

[2.1.3.4.1. Hardware requirements 7](#_Toc425867522)

[2.1.3.4.2. Software requirements 8](#_Toc425867523)

[2.2. Project Organization 8](#_Toc425867524)

[2.2.1. System Process Model 8](#_Toc425867525)

[2.2.1.1. Iterative Development Model 8](#_Toc425867526)

[2.2.1.2. Project Life Cycle 9](#_Toc425867527)

[2.2.2. Role and Responsibilities 11](#_Toc425867528)

[2.2.3. Tools and Techniques 13](#_Toc425867529)

[2.3. Project Management Plan 14](#_Toc425867530)

[2.3.1. Tasks 14](#_Toc425867531)

[2.3.1.1. Initiating 14](#_Toc425867532)

[Description: 14](#_Toc425867533)

[Output: 14](#_Toc425867534)

[Deliverables: 14](#_Toc425867535)

[Resources Needed: 14](#_Toc425867536)

[Dependencies and Constraints: 14](#_Toc425867537)

[Risks: 14](#_Toc425867538)

[2.3.1.2. Planning 14](#_Toc425867539)

[Description: 14](#_Toc425867540)

[Deliverables: 14](#_Toc425867541)

[Resources Needed: 14](#_Toc425867542)

[Dependencies and Constraints: 15](#_Toc425867543)

[Risks: 15](#_Toc425867544)

[2.3.1.3. Create Software Requirements Specification 15](#_Toc425867545)

[Description: 15](#_Toc425867546)

[Output: 15](#_Toc425867547)

[Deliverables: 15](#_Toc425867548)

[Resources Needed: 15](#_Toc425867549)

[Dependencies and Constraints: 15](#_Toc425867550)

[Risks: 15](#_Toc425867551)

[2.3.1.4. Design Database 15](#_Toc425867552)

[Description: 15](#_Toc425867553)

[Output: 15](#_Toc425867554)

[Deliverables: 15](#_Toc425867555)

[Resources Needed: 15](#_Toc425867556)

[Dependencies and Constraints: 15](#_Toc425867557)

[Risks: 16](#_Toc425867558)

[2.3.1.5. Create Software Design Description 16](#_Toc425867559)

[Description: 16](#_Toc425867560)

[Output: 16](#_Toc425867561)

[Deliverables: 16](#_Toc425867562)

[Resources Needed: 16](#_Toc425867563)

[Dependencies and Constraints: 16](#_Toc425867564)

[Risks: 16](#_Toc425867565)

[2.3.1.6. Layout Design 16](#_Toc425867566)

[Description: 16](#_Toc425867567)

[Output: 16](#_Toc425867568)

[Deliverables: 16](#_Toc425867569)

[Resources Needed: 16](#_Toc425867570)

[Dependencies and Constraints: 16](#_Toc425867571)

[Risks: 16](#_Toc425867572)

[2.3.1.7. Create Coding Framework 17](#_Toc425867573)

[Description: 17](#_Toc425867574)

[Output: 17](#_Toc425867575)

[Deliverables: 17](#_Toc425867576)

[Resources Needed: 17](#_Toc425867577)

[Dependencies and Constraints: 17](#_Toc425867578)

[Risks: 17](#_Toc425867579)

[2.3.1.8. Coding 17](#_Toc425867580)

[Description: 17](#_Toc425867581)

[Output: 17](#_Toc425867582)

[Deliverables: 17](#_Toc425867583)

[Resources Needed: 17](#_Toc425867584)

[Dependencies and Constraints: 17](#_Toc425867585)

[Risks: 17](#_Toc425867586)

[2.3.1.9. System Test 18](#_Toc425867587)

[Description: 18](#_Toc425867588)

[Output: 18](#_Toc425867589)

[Deliverables: 18](#_Toc425867590)

[Resources Needed: 18](#_Toc425867591)

[Dependencies and Constraints: 18](#_Toc425867592)

[Risks: 18](#_Toc425867593)

[2.3.1.10. Input Initial Data 18](#_Toc425867594)

[Description: 18](#_Toc425867595)

[Output: 18](#_Toc425867596)

[Deliverables: 18](#_Toc425867597)

[Resources Needed: 18](#_Toc425867598)

[Dependencies and Constraints: 18](#_Toc425867599)

[Risks: 18](#_Toc425867600)

[2.3.1.11. Deployment 18](#_Toc425867601)

[Description: 18](#_Toc425867602)

[Output: 19](#_Toc425867603)

[Deliverables: 19](#_Toc425867604)

[Resources Needed: 19](#_Toc425867605)

[Dependencies and Constraints: 19](#_Toc425867606)

[Risks: 19](#_Toc425867607)

[2.3.2. Tasks Sheet: Assignment and Timetable 19](#_Toc425867608)

[2.3.3. All Meeting Minutes 21](#_Toc425867609)

[2.4. Convention Rules 21](#_Toc425867610)

[2.4.1. Naming Conventions 21](#_Toc425867611)

[Pascal case 21](#_Toc425867612)

[Camel case 21](#_Toc425867613)

[Uppercase 21](#_Toc425867614)

[Definition: 21](#_Toc425867615)

[2.4.2. Coding Styles 22](#_Toc425867616)

[2.4.3. Comments 22](#_Toc425867617)

[2.4.4. Language Guidelines 23](#_Toc425867618)

[2.3. Other material 23](#_Toc425867619)

# List of Tables

# Definition, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Definition** |
| JSS | Job Searching System |
| App | Application |
| QA | Quality Assurance |
| BA | Business Analysis |
| DEV | Developer |
| GUI | Graphical User Interface |
| OS | Operating System |
| Admin | Administrator |
| API | Application Programming Interface |
| HTTP | Hyper Text Transfer Protocol |

# 2. Project Management Plan

## 2.1. Problem Definition

### 2.1.1. Name of this Capstone Project

The name of our capstone project is “Job Searching System” which means “Hệ thống tìm việc làm” in Vietnamese. The alternative name is JSS. But it might be different once the project is completed and comes to deployment.

### 2.1.2. Problem Abstract

This project Job Searching System (JSS) is a system in which jobseekers can register themselves online, view organization requirements and apply for the suitable job. JSS provides online help to the users all over the world. This kind of system plays an important role in simplifying the recruitment process. The system has facilities where prospective candidates can upload their CV’s and apply for jobs suited to them. It also makes it possible for organization to post their staffing requirements and view profiles of interested candidates. JSS is developed to provide all categories of job and help to get various type of job, to help jobseekers for getting the quick job.

The need for such services is really high. There are currently some other systems providing quite the same services, but all have their own disadvantages. The purpose of this project is to create a system that overcomes all those disadvantages and becomes the most user-friendly system for Vietnamese people.

Once completed, the product must possess the following characteristics:

* Specialized for Vietnamese users
* Provides detailed information about career and job opportunities, application process, selection procedure for various jobs, personality development guides.
* Sorting and filtering best suitable jobseekers based on some criteria. Recruiters will not have to waste his time for finding right employee at right post.
* Jobseekers can create entries in order to share experiences
* Utilized newest technologies to make the system user-friendly and have best performance.

### 2.1.3. Project Overview

#### 2.1.3.1. The Current System

There is no current system. The product is building from scratch as a new idea.

#### 2.1.3.2. The Proposed System

The main system will be a website. Job Searching System is ambitious to overcome the disadvantages of existing systems, based on the analyzing view above.

The system will provide the following main features in front-end:

* Guest can visit the site, search jobs, search recruiter’s profiles of any kinds
* Jobseeker can upload their CV’s, update skill, apply for jobs suited to them
* Jobseeker can create entries in order to share experiences
* Recruiter can matching CV, sort and filter best suitable jobseekers based on some criteria
* Recruiter can buy some packages online service for them and post jobs
* Authorized users can post problems and contact with customer service via phone

The back-end will allow the administrators to manage the most important activities as well as information on the site.

#### 2.1.3.3. Boundaries of the System

* The system can be used by every people with a laptop/computer, a smart phone with Internet connection.
* The language of the system is English and Vietnamese.
* The complete product includes:
* The website for Admin, Recruiter and Job Seekers.
* All process documents involved.

#### 2.1.3.4. Development Environment

##### 2.1.3.4.1. Hardware requirements

**For server**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| Operating System | Window Server 2008 | Window Server 2008 |
| Computer Processor | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz) |
| Computer Memory | 1GB RAM | 2GB or more |

Table 2-1: Hardware Requirement for Server

**For Mobile**

|  |  |  |
| --- | --- | --- |
| Mobile | Minimum Requirements | Recommended |
| Internet Connection | 2 Mbps | 4 Mbps |
| Operating System | Android 4.0 | Android 4.4.2 |
| Hardware | NFC supported | NFC supported |
| Memory | 512MB | 1GB or more |

Table 2-2: Hardware Requirement for Mobile

##### 2.1.3.4.2. Software requirements

* Window Server 2008: operating system and platform for development.
* SQL Server 2008 Enterprise R2: used to create and manage the database for system.
* Visual Studio 2012: used to implement website and web service.
* Google Code & Tortoise SVN: used for source control.
* Astah: used to create models and diagrams.
* Facebook: used for communication and meeting.

## 2.2. Project Organization

### 2.2.1. System Process Model

Software needs a software process model to guarantee the deadline and quality. Therefore, it is really important to select the suitable software process model. To accomplish the project, our team decided to choose Iterative Development Model.

##### 2.2.1.1. Iterative Development Model

An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This is then repeated, producing a new version of the software each cycle of model.

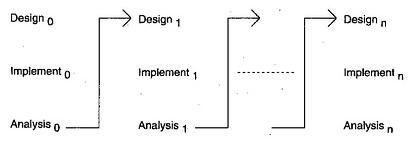


Figure 2-1: Iterative Software Development Model overview

An iterative development approach that allows the functionality to be delivered in parts has become a necessity and an effective way to manage risks. A common iterative approach is to decide what should be developed in iteration and then plan the iteration accordingly. A somewhat different iterative approach is to time box different iterations. In this approach, the length of iteration is fixed and what should be developed in iteration is adjusted to fit the time box. Generally, the time boxed iterations are executed in sequence, with some overlap where feasible. With the project, we decided to choose the second approach – time boxed iteration:

* May 7th – Jul 15th: complete main function of the system.
* Jul 15th – Aug 15th: review and upgrade auxiliary functions.
* Aug 15th - Aug 25th: review and complete the system.

The following advantages of Iterative Development Model are the reason why we choose for this project:

* **Deliver base functionality first.** Iteration can be planned so that the basic, core functions are created first, and the subsequently built upon in future iterations. It guarantees that we will not waste our time on auxiliary functions
* **Defect tracker**: In iterative model, we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
* **Project size (not too big and not too small).** Iterative process focus on smaller projects that can be more quickly started and ended with tangibles deliveries. Some large project can even be split up into a set of smaller projects that can utilize Iterative processes.
* **Good organization and team flexibility.** Iterative Process requires a high degree of flexibility and the ability to manage through change.

##### 2.2.1.2. Project Life Cycle

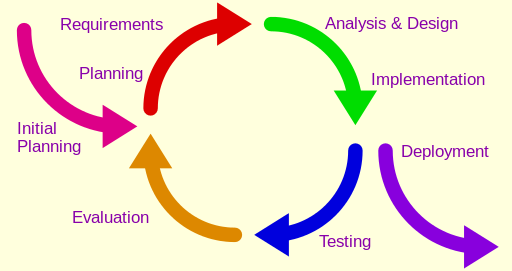


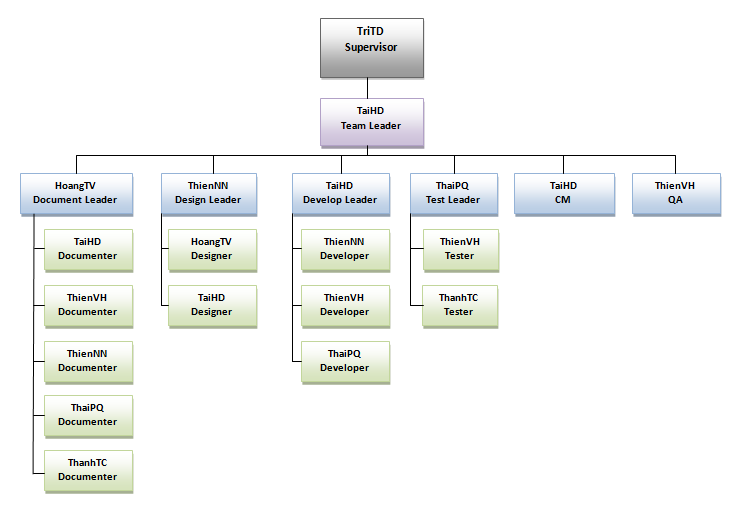
Figure 2-2: Iterative Software Development Model life cycle

The overall flow of the Iterative model is as follows:

* **Initial Plan:**
* Determining the scope, conditions and limitations of the project
* List the main functions of the system
* Choose suitable architecture for the system
* Identify project risks
* Complete Report #1
* **Planning:**
* Create a schedule that describes the activities needed to complete the iteration.
* Decide which function to be completed in the iteration.
* Complete Report #2
* **Requirement:**
* Capture the business requirement
* **Analysis & Design:**
* Detail the functional requirements which are decided to complete before.
* Provide stable system architecture which can implement the functions.
* Complete Report #3, and Report #4
* **Implementation:**
* Build a version of product from what we designed. (Coding)
* Complete a version of software.
* **Testing:**
* Create test plan.
* Test and fix bug.
* Complete Report #5.
* **Deployment:**
* Deploy a version of software with bug fixed.
* Complete Report #6
* **Evaluation:**
* Review and evaluate the iteration.

### 2.2.2. Role and Responsibilities

The list of the specific roles and responsibilities in JSS project are listed below:



**Figure 2-3:** Roles and assigned members in JSS project

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Full name** | **Roles** | **Responsibilities** |
| **1** | Huỳnh Đức Tài | PM, Develope leader, Designer, Documenter, CM | * Managing process * Develop the developing plan based on project plan * Contact with supervisor in providing updated information * Perform the required daily and weekly backups of the team’s working directory * Prepare documents * GUI Design * Training * Coding |
| **2** | Nguyễn Ngọc Thiện | Design Leader, Documenter, Dev | * GUI Design * Designing database * Prepare documents * Clarifying requirements * Coding |
| **3** | Phạm Quốc Thái | Test Leader, Documenter | * Develop the test plan based on project plan * Develop test cases for the system * Prepare documents * Clarifying requirements * Create test plan * Coding * Testing |
| **4** | Võ Hoàn Thiện | QA, Documenter, Dev | * Write and send meeting-minute to all team * Clarifying requirements * Prepare documents * Coding * Testing |
| **5** | Trịnh Văn Hoàng | Document Leader, Designer | * Managing documents * Maintain all documentation by preparing standard templates * Designing database * Clarifying requirements * Prepare documents * GUI Design |
| **6** | Trần Chung Thành | Documenter, Tester | * Designing database * Clarifying requirements * Prepare documents * Creating system test cases Testing |

**Table 2-3:** Roles and Responsibilities

### 2.2.3. Tools and Techniques

|  |  |
| --- | --- |
| **Title** | **Detail** |
| **Software and Framework** | **Software:**  - Microsoft® Office, Microsoft® Project (2007, 2010, 2013)  - Operating System: Microsoft® Windows 7, 8.1  - Source Control: Tortoise SVN 1.7.9  - Adobe Photoshop CS5  - Visual Studio 2013  - Microsoft SQL Server 2012  - Browser: Chrome 41, Firefox 37  **Framework:**  - MVC 5  - C#  - .NET Framework 4.5 |
| **Hardware** | Notebooks for developing/testing with the minimum configuration:  RAM:2GB Hard disk: 150GB Chipset: Core 2 Dual 2.0 GHz |
| A server computer with the minimum configuration:  RAM: 2GB  Hard disk: 150GB  Chipset: Core 2 Dual 2.0 GHz |

**Table 2-4:** Tools and Techniques

## 2.3. Project Management Plan

### 2.3.1. Tasks

#### 2.3.1.1. Initiating

##### Description:

* Kick of meeting
* Define how to communicate
* Register Capstone Project
* Prepare development environment
* Study technology and framework

##### Output:

A new project

##### Deliverables:

N/A

##### Resources Needed:

6 people for a week

##### Dependencies and Constraints:

None

##### Risks:

Some one’s didn’t come

#### 2.3.1.2. Planning

##### Description:

* Define project and system scope
* Study the other websites functions

Output:

Introduction and Software Project Management Plan

##### Deliverables:

N/A

##### Resources Needed:

6 people for 3 days

##### Dependencies and Constraints:

None

##### Risks:

Conflict ideas and be changed in future

#### 2.3.1.3. Create Software Requirements Specification

##### Description:

Create software requirements specification

##### Output:

Software Requirements Specification (SRS) document

##### Deliverables:

Deliver SRS document before 04/06/2015

##### Resources Needed:

6 people for 17 days

##### Dependencies and Constraints:

None

##### Risks:

Because there is no actual user, and the requirements come from all the team members, conflicts may happen regularly.

#### 2.3.1.4. Design Database

##### Description:

Create logical and physical database design

##### Output:

Database script and Database Design

##### Deliverables:

Deliver with software design description before 11/06/2015

##### Resources Needed:

3 people for a week

##### Dependencies and Constraints:

Depends on the completion of SRS

##### Risks:

SRS may not be detailed enough to capture the business rules, the Database Design will be changed much in the future.

#### 2.3.1.5. Create Software Design Description

##### Description:

Design the system in C#

##### Output:

Architecture design, detailed design, diagrams and design description

##### Deliverables:

SDD before 18/06/2015

##### Resources Needed:

6 people for a week

##### Dependencies and Constraints:

Depends on the completion of SRS

##### Risks:

Risks may include causing the system hard to maintain and choosing inappropriate architecture.

#### 2.3.1.6. Layout Design

##### Description:

Create the main GUI for layout, and create global styles that will be applied to the GUI

##### Output:

HTML layout and CSS files

##### Deliverables:

None

##### Resources Needed:

3 people for 3 days

##### Dependencies and Constraints:

None

##### Risks:

None

#### 2.3.1.7. Create Coding Framework

##### Description:

Create the project solution files and common classes, implement common functions.

##### Output:

MVC 5 project and solution files containing coding framework

##### Deliverables:

None

##### Resources Needed:

4 people for a week

##### Dependencies and Constraints:

Depends on the completion of SRS and Architecture design

##### Risks:

Because all the technologies used in this project are so new then estimates become even harder to get right.

#### 2.3.1.8. Coding

##### Description:

Coding and implement the system to reflect all the requirements

##### Output:

Source code of the project

##### Deliverables:

Executable programs and source code before 20/07/2015

##### Resources Needed:

4 people for 3 weeks

##### Dependencies and Constraints:

Depends on the completion of SRS, SDD, Database Design, Coding framework

##### Risks:

Team members may not easily get used to new technologies. The team does not have a common place and may lack of time to implement all the requirements.

#### 2.3.1.9. System Test

##### Description:

Create the test plan, perform system test for the system

##### Output:

System test report

##### Deliverables:

Software Test Documentation before 06/08/2015

##### Resources Needed:

3 people for 2 weeks

##### Dependencies and Constraints:

Depends on Coding is finished

##### Risks:

Developers are also responsible for system testing

#### 2.3.1.10. Input Initial Data

##### Description:

Collect and input initial data for the system before going live

##### Output:

Initial data

##### Deliverables:

None

##### Resources Needed:

6 people for 3 days

##### Dependencies and Constraints:

Coding and System test are finished

##### Risks:

None

#### 2.3.1.11. Deployment

##### Description:

Deploy the system to the Internet

##### Output:

Running website with domain and hosting

##### Deliverables:

None

##### Resources Needed:

6 people for 3 days

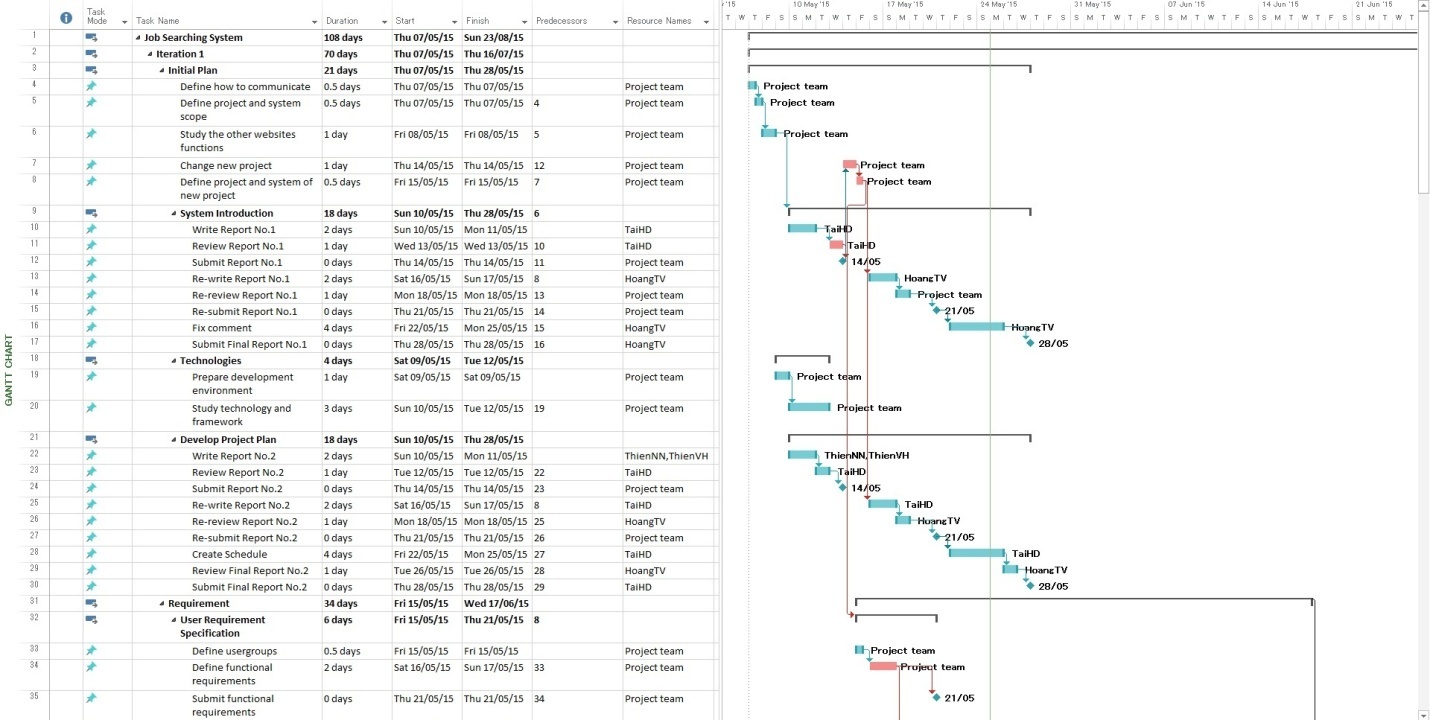
##### Dependencies and Constraints:

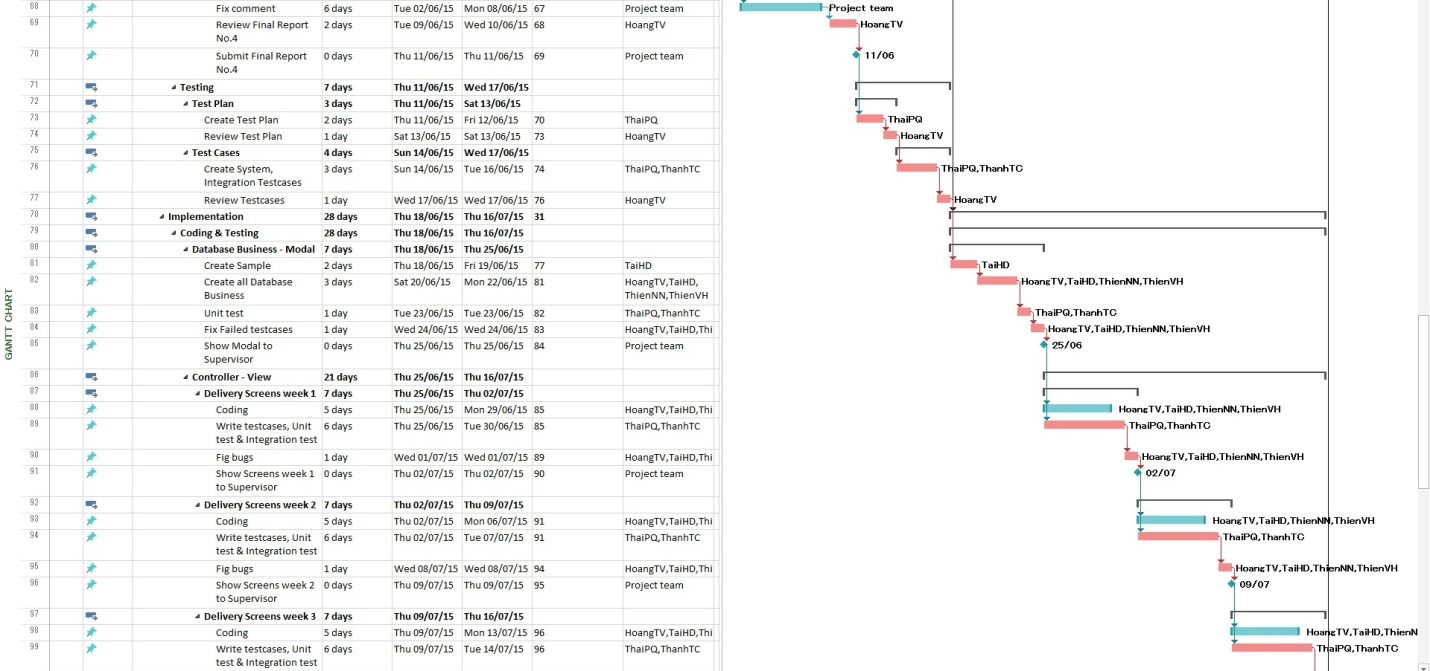
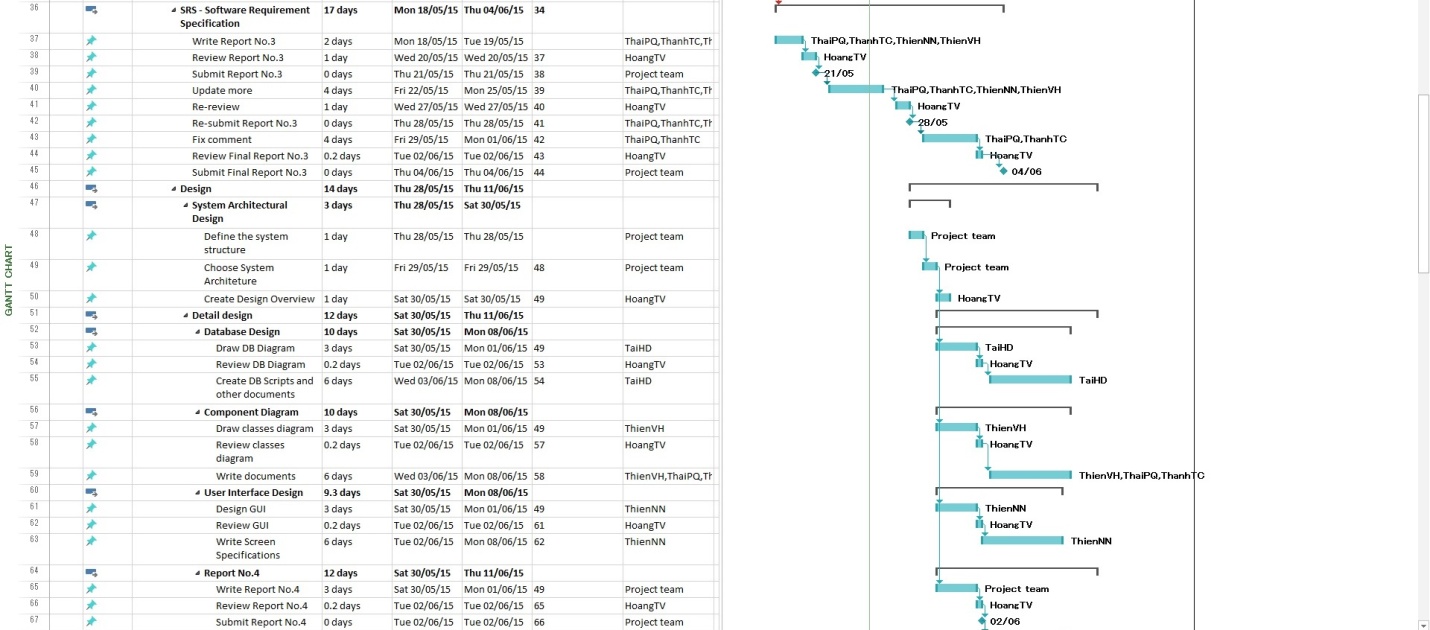
Coding and System test are finished, Initial data is inputted

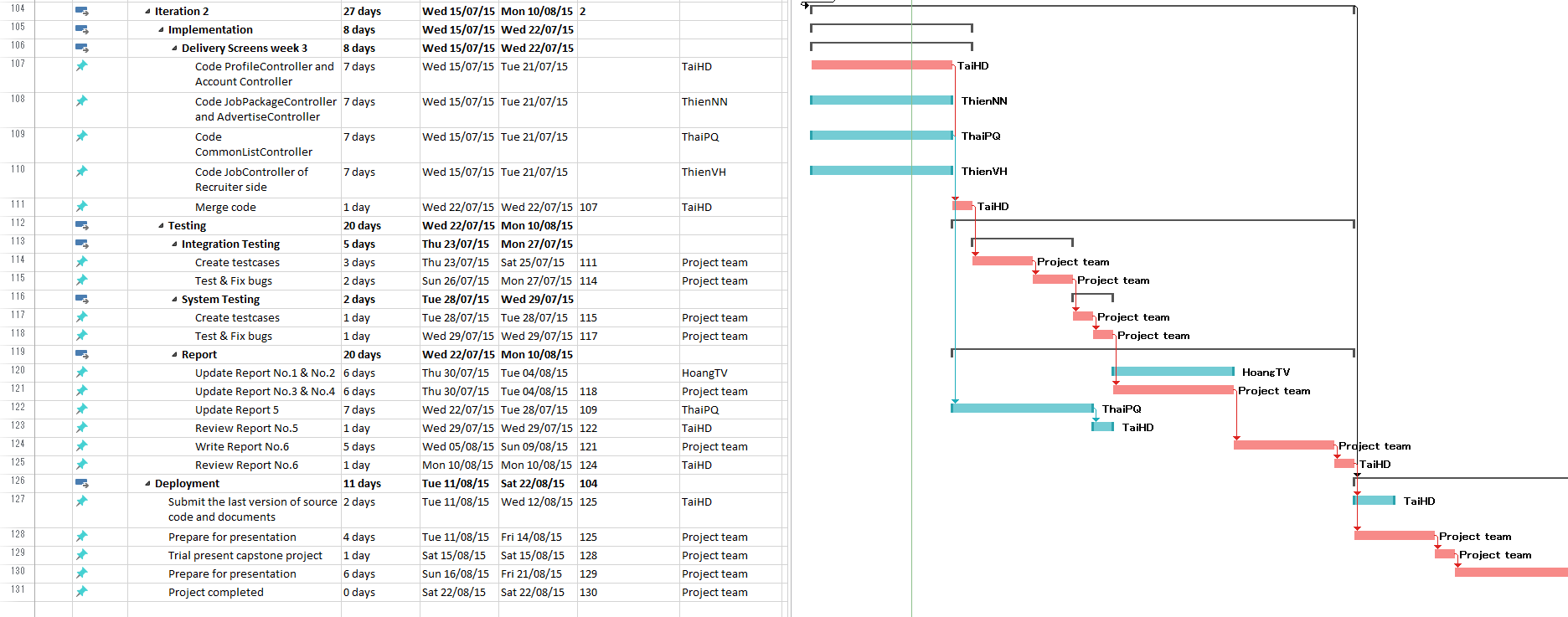
##### Risks:

None

### 2.3.2. Tasks Sheet: Assignment and Timetable







### 2.3.3. All Meeting Minutes

## 2.4. Convention Rules

The original of this document was developed by the Microsoft special interest group. We made some add-ons. The following rules follow the standard rules for developing applications using .NET and C#.

### 2.4.1. Naming Conventions

The following terms describe different ways to case identifiers.

#### Pascal case

The first letter in the identifier and the first letter of each subsequent concatenated word are capitalized. For example:

BackColor, DataSet

#### Camel case

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

numberOfDays, isValid

#### Uppercase

All letters in the identifier are capitalized. For example:

ID, PI

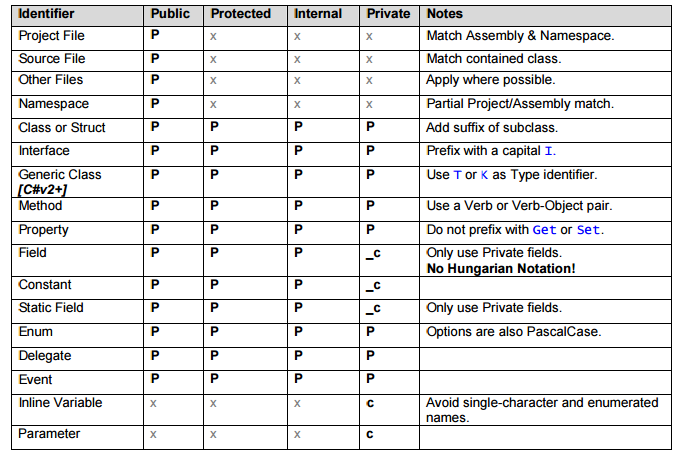
#### Definition:

“c” = camelCase

“P” = PascalCase

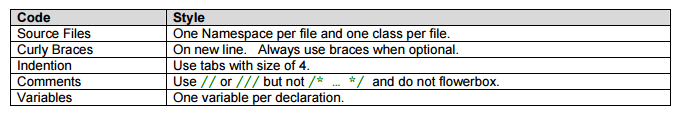
“\_” = Prefix with \_Underscore

“x” = Not Applicable.

****

**Table 2-5:** Naming Conventions

### 2.4.2. Coding Styles



**Table 2-6:** Coding Styles

### 2.4.3. Comments

* All comments should be written in the same language, be grammatically correct, and contain appropriate punctuation
* Use // or /// but never /\* … \*/
* Use inline-comments to explain assumptions, known issues, and algorithm insights
* Do not use inline-comments to explain obvious code. Well written code is self documenting.
* Always apply C# comment-blocks (///) to public, protected, and internal declarations
* Only use C# comment-blocks for documenting the API
* Always include comments. Include , , and comment sections where applicable
* Include and where possible

### 2.4.4. Language Guidelines

Using C# Code Convention From:

<http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx>

## 2.3. Other material