**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Student and Business Website**

|  |  |
| --- | --- |
| **GROUP 8** | |
| **Group member** | Nguyễn Văn Linh – Team Leader – SE60914  Hoàng Ngọc Anh – Team Member – SE61099  Lương Quốc Trung – Team Member – SE60903  Trần Hoàng Hải – Team Member – SE60739  Vũ Thế Anh – Team Member – 60111 |
| **Supervisor** | Mr. Nguyễn Trọng Tài |
| **Ext. Supervisor** | None |
| **Capstone Project code** | SBW |

***Ho Chi Minh City***, ***January 6, 2015***

# Table of Contents

[Table of Contents 3](#_Toc408433082)

[List of Tables 6](#_Toc408433083)

[List of Figures 7](#_Toc408433084)

[Definitions, Acronyms, and Abbreviations 7](#_Toc408433085)

[A. Introduction 8](#_Toc408433086)

[1. Project Information 8](#_Toc408433087)

[2. Introduction 8](#_Toc408433088)

[3. Current Situation 8](#_Toc408433089)

[4. Problem Definition 8](#_Toc408433090)

[5. Proposed Solution 8](#_Toc408433091)

[5.1 Feature functions 8](#_Toc408433092)

[5.2 Advantages and disadvantages 9](#_Toc408433093)

[6. Functional Requirement 9](#_Toc408433094)

[7. Role and Responsibility 10](#_Toc408433095)

[B. Software Project Management Plan 11](#_Toc408433096)

[1. Problem Definition 11](#_Toc408433097)

[1.1 Name of this Capstone Project 11](#_Toc408433098)

[1.2 Problem Abstract 11](#_Toc408433099)

[1.3 Project Overview 11](#_Toc408433100)

[2. Project organization 11](#_Toc408433101)

[2.1 Software Process Model 11](#_Toc408433102)

[2.2 Roles and responsibilities 11](#_Toc408433103)

[2.3 Tools and Techniques 12](#_Toc408433104)

[3. Project Management Plan 12](#_Toc408433105)

[3.1 Software Development Life Cycle 12](#_Toc408433106)

[3.2 All Meeting Minutes 12](#_Toc408433107)

[4. Coding Convention 12](#_Toc408433108)

[C. Software Requirement Specification 12](#_Toc408433109)

[1. User Requirement Specification 12](#_Toc408433110)

[1.1 Guest Requirement 12](#_Toc408433111)

[1.2 Member Requirement 12](#_Toc408433112)

[1.3 Partner Requirement 12](#_Toc408433113)

[1.4 Staff Requirement 12](#_Toc408433114)

[1.5 Admin Requirement 12](#_Toc408433115)

[2. System Requirement Specification 12](#_Toc408433116)

[2.1 External Interface Requirement 12](#_Toc408433117)

[2.2 System Overview Use Case 12](#_Toc408433118)

[2.3 List of Use Case 12](#_Toc408433119)

[3. Software System Attribute 13](#_Toc408433120)

[3.1 Usability 13](#_Toc408433121)

[3.2 Reliability 13](#_Toc408433122)

[3.3 Availability 14](#_Toc408433123)

[3.4 Security 14](#_Toc408433124)

[3.5 Maintainability 14](#_Toc408433125)

[3.6 Portability 14](#_Toc408433126)

[3.7 Performance 14](#_Toc408433127)

[4. Conceptual Diagram 14](#_Toc408433128)

[Data Dictionary 14](#_Toc408433129)

[D. Software Design Description 14](#_Toc408433130)

[1. Design Overview 14](#_Toc408433131)

[2. System Architectural Design 14](#_Toc408433132)

[2.1 Web Application architecture description 14](#_Toc408433133)

[2.2 Mobile Application architecture description 14](#_Toc408433134)

[3. Component Diagram 14](#_Toc408433135)

[4. Detailed Description 15](#_Toc408433136)

[4.1 Class Diagram 15](#_Toc408433137)

[4.2 Class Diagram Explanation 15](#_Toc408433138)

[4.3 Interaction Diagram 15](#_Toc408433139)

[5. User Interface Design 15](#_Toc408433140)

[5.1 Guest Interface Design 15](#_Toc408433141)

[6. Database Design 15](#_Toc408433142)

[6.1 Logical Diagram 15](#_Toc408433143)

[6.2 Data Dictionary 15](#_Toc408433144)

[7. Algorithms 15](#_Toc408433145)

[7.1 Document Breakdown 15](#_Toc408433146)

[7.2 String Comparison 16](#_Toc408433147)

[E. System Implementation & Test 16](#_Toc408433148)

[1. Introduction 16](#_Toc408433149)

[1.1 Overview 16](#_Toc408433150)

[1.2 Test Approach 16](#_Toc408433151)

[2. Database Relationship Diagram 16](#_Toc408433152)

[2.1 Physical Diagram 16](#_Toc408433153)

[2.2 Data Dictionary 16](#_Toc408433154)

[3. Performance Measures 17](#_Toc408433155)

[3.1 Clustering Performance 17](#_Toc408433156)

[4. Test Plan 17](#_Toc408433157)

[4.1 Features to be tested 17](#_Toc408433158)

[4.2 Features not to be tested 17](#_Toc408433159)

[5. System Testing Test Case 18](#_Toc408433160)

[5.1 Guest Test Case 18](#_Toc408433161)

[F. Software User’s Manual 18](#_Toc408433162)

[1. Installation Guide 18](#_Toc408433163)

[1.1 Setting up environment at server side 18](#_Toc408433164)

[1.2 Deployment at server side 18](#_Toc408433165)

[1.3 Setting up the environment at client side 19](#_Toc408433166)

[2. User Guide 19](#_Toc408433167)

[G. Appendix 19](#_Toc408433168)

# List of Tables

# List of Figures

# Definitions, Acronyms, and Abbreviations

|  |  |  |
| --- | --- | --- |
| **No.** | **Abbreviation & Acronym** | **Definition** |
|  |  |  |

**Table 1: Definitions, Acronyms and Abbreviations**

# Introduction

## Project Information

* Project name: Student and Business Website
* Project Code: SBW
* Product Type: Website Application
* Start Date: Jan 6th, 2015
* End Date: May 2nd, 2015

## Introduction

Nowadays, student have free time and want to find job get more income and experience. In other side, employers need to recruit suitable candidates in the shortest time. Our system automatic search candidates and notify for them.

Also, websites can support online contest in other to help business decrease time instead of traditional interview.

Business, school can create an event and system will announce to students.

System supports sharing material between students depending on hobbies, skills and jobs.

The goal of the website is to facilitate communication and sharing information between students and businesses, and also between students together.

## Current Situation

User must have register to use this web application. User can search job want to join or create a job to employ. After that, business can meet student to discussion about their job.

In Vietnam, all most websites don’t support all the features that we extend to provide, such as: jobstreet.vn only support job recruiting, freelancer.com don’t support event…

So inclusion, we create all-sided solution to solve above problems.

## Problem Definition

Users want to find job, and suppliers want to find out candidates for their work. But the interaction between them is not efficient, such as employees need to find jobs, but they don’t have chance to review or heard from others about the job that employers provide.

People who want to take job do not have place to discuss about the job or career that they care together.

students want to share information and some knowledge for group or anyone have same hobby or jobs, but almost websites for job seeking not have function share or comment to people know or learn skill for that job.

## Proposed Solution

Website will auto find candidate for job which had been created on website, after website will notify information’s students about that job to employer and notify to all students had been match with that job.

Help employer can make a contest online to filter candidate if have more than they needs.

When a student upload a material, system will auto suggest students’ needs that material, they have same hobbies, group or company…

### **Feature functions**

* Create curriculum vitae online.
* Create form recruitment online with skill, session, age, time …
* Auto search students base on skill, session, age, time... From form recruitment.
* Auto notify for both students and employer about job, list candidate.
* Auto find your friend or your group, colleague... to share material.
* Create test for students if employer need small quantity while website have more.

### Advantages and disadvantages

Advantages:

* Help employer can find candidates in short time, may be some minutes. And can help them contact with those candidates.
* Quick select candidates after employer have a large list candidates.
* Share material between student with student.

Disadvantages:

* Matching students with employers is not exactly.
* Manage test bank with session job is not effective.

## Functional Requirement

Function requirements of the system are listed as below:

* Member management: to use the website, all students and businesses may be need to register.
* Employment information management: the businesses can post employment request, then students can post their curriculum vitae to apply the jobs. The system should suggest the best matching between the request and curriculum vitae.
* Contest/events management: the business can post information about contest or events, then student can join it
* Manage sharing material learning between student together
* Searching, statistic.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Nguyễn Trọng Tài | Project Manager | Instructor | TaiNT@fpt.edu.vn |
| 2 | Nguyễn Văn Linh | Developer | Leader | Linhnvse60914@fpt.edu.vn |
| 3 | Hoàng Ngọc Anh | Developer | Member | Anhhnse61099@fpt.edu.vn |
| 4 | Lương Quốc Trung | Developer | Member | Trunglqse60903@fpt.edu.vn |
| 5 | Trần Hoàng Hải | Developer | Member | Haithse60739@fpt.edu.vn |
| 6 | Vũ Thế Anh | Developer | Member | AnhVT60111@fpt.edu.vn |

Table 2: Roles and Responsibility

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

Student and Business Website (SBW)

### Problem Abstract

Nowadays, the unemployment rate for students in Viet Nam is quite high. Having a part-time job allows students to get life experiences and work skills, which will help them have solid foundations to work after graduating. But the most difficulty is how students can find jobs easily and how employers can recruit suitable students quickly. Therefore, this website will help both students and employers be able to connect to each other.

The website will manage users and recruitments. Besides, the website can find the applicants with hiring jobs, events or competitions automatically.

Furthermore, the website also holds the online competitions for students or supports online tests for the recruitments, which helps decrease recruitment time of some jobs having the large number of applicants.

### Project Overview

#### Current Situation and Disadvantages

(Tiếng việt)

Hiện tại website chỉ tập trung hỗ trợ sinh viên trong thời gian đi học nên chưa giải quyết dc vấn đề lớn hơn là sinh viên sau khi ra trường.

Ưu điểm :

Nhược điểm :

#### The Proposed System

##### Web Site

**Student Requirement**

* Manage Profile
* Manage Job
* Manage Material
* Manage Contest
* Manage Test Bank
* Manage All Events
* Manage Account
* Automatic random questions for contest
* Automatic suggest questions for contest
* Automatic suggest candidates for employer
* Automatic suggest event for students
* Automatic suggest job for students

#### Boundaries of the System

Language : Vietnamese.

Website, server,…

#### Development Environment

##### Hardware requirements

**For Server**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wifi (4 Mbps) | Cable, Wifi (8 Mbps) |
| Operating System | XP, Vista, 7, 8 | XP, Vista, 7, 8 |
| Computer Processor | Intel® Core 2 Duo | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| Computer Memory | 1GB RAM | 4GB or more |

Table 3: Hardware Requirement for Server

**For Mobile Application**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wifi (4 Mbps) | Cable, Wifi (8 Mbps) |
| Operating System | XP, Vista, 7, 8 | XP, Vista, 7, 8 |
| Computer Processor | Intel® Core 2 Duo | Intel® Core(TM) i3 CPU , M 460 @ 2.53GHz |
| Computer Memory | 1GB RAM | 2GB or more |

**Table 4: Hardware Requirement for Mobile**

##### Software requirements

**For Mobile Client :**

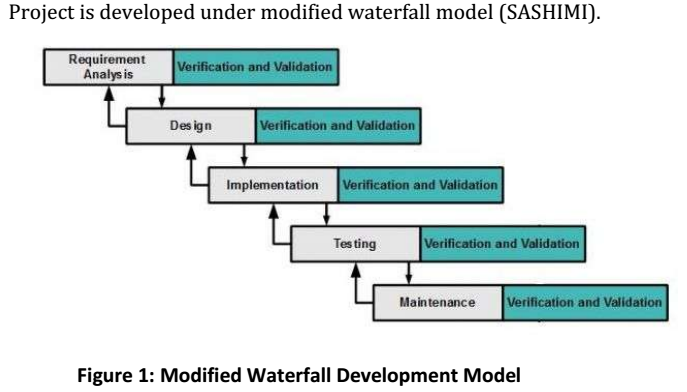
* Internet explorer 8 and above, google chrome, fire fox.

**For develop website :**

* **SQL Server 2008 Enterprise R2**: used to create and manage the database for system.
* **Visual Studio 2013**: used to implement website and web service.
* **StarUML 2.0** : used to create models and diagrams.
* **Skype, facebook.com**: used for communication and meeting.
* **Dropbox**: sharing document.

## Project organization

### Software Process Model



### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Nguyễn Trọng Tài | Project manager | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | nguyễn văn linh | Leader | * Managing process * Prepare documents * Create test plan * GUI Design |
| **3** | Lương Quốc Trung | Team member | * Coding, testing * Prepare documents * GUI Design |
| **4** | Hoàng Ngọc Anh | Team member | * Coding, testing * Prepare documents |
| **5** | Trần Hoàng Hải | Team member | * Coding, testing * Prepare documents |
| **6** | Vũ Thế Anh | Team member | * Coding, testing * Prepare documents |

Table 5: Roles and Responsibility Details

### Tools and Techniques

- **SQL Server 2008 Enterprise R2**: used to create and manage the database for system.

- **Visual Studio 2013**: used to implement website and web service.

- **StarUML 2.0** : used to create models and diagrams.

## Project Management Plan

### Software Development Life Cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource needed** | **Dependencies and Constraints** | **Risks** |
| **Requirement Analysis** | Find which systems current provide similar service, the strengths and weakness. | -Introduction of proposed system. -Software requirement specification. -Project Task Plan. - Prototypes | 50 man days |  | - Missing requirement - Unclear scope of project - Lack of member share of understand |
| **Design** | - Architecture design for the system - Detail design using top-down break down - Choose Architecture style | - Software Design Document - Base code structure - Technology notes | 60 man days | Depend on “Requirement Analysis” | - Lack of experience. - Not fulfil requirement. |
| **Development** | - Coding system core functions and other feature with GUI - Unit test | - Main user’s functions on web and mobile. - Unit test document | 60 man days | Depend on “Design”. | - Lack of experience and knowledge. - Human mistake. |
| **Testing** | - Integration test the system - Alpha test - Correct bugs - Beta test - Acceptance test | - Test document - Defect log | 40 man days | Depend on “Implementation” | - Lack of experience - Missing test case |
| **Implementation** | - Deploy on sever. |  |  |  |  |
| **Maintenance** | - Maintenance sever. | - Installation guide - User Manual | 20 man days | Depend on “Testing” | - Lack of experience.  - Missing step setup.  - User get mistake. |

Table 6: Software Development Life Cycle Detail

### Phase Detail

#### Phase 1: Preliminary Investigation

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Statistical requirements** | Find and statistical requirement of user. |  |
| **2. Identify main object** | Define which main object will be focused. |  |

#### Phase 2: Analysis

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Collect requirements** | Find which systems currently provide similar service, their strengths and weakness. |  |
| **2. Identify and clarify** **main functions.** | Define which main functions system should provide. |  |
| **3. Create System** **Introduction.** | Complete Introduction Report. |  |
| **4. Software Project** **Management Plan.** | Prepare Project Management Plan. |  |
| **5. Website Prototype.** | Build a prototype of proposed system (Website/Mobile). |  |
| **6. SRS** | Create SRS document. |  |

Table 7: Requirement Analysis

#### Phase 3: Design

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **1. Architecture Design** | Implement function import and breakdown data from docx files. |  |
| **2. Detailed Design** | Compare new document with existed documents of system. |  |
| **3. Database Design** | Get jobs from other server to recommendation. |  |
| **4. Technology** **research** | Create search engine for basic search and advance search. |  |
| **5. Design Document** | Create software design document |  |

Table 8: Detail

#### Phase 4: Development

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 9: Implementation

#### Phase 5: Testing

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 10: Testing

#### Phase 6: Implementation

#### Phase 7: Maintenance

### All Meeting Minutes

Refer to Meeting Minutes folder.

Error

## Coding Convention

* Commenting Convention.

Place the comment on a separate line, not at the end of a line of code.

Begin comment text with an uppercase letter.

End comment text with a period.

Insert one space between the comment delimiter (//) and the comment text, as shown in the following example.

Ex:

// The following declaration creates a query. It does not run

// the query.

* String data type:

Use the + operator to concatenate short strings, as shown in the following code.

Ex:

string displayName = nameList[n].LastName + ", " + nameList[n].FirstName;

# Software Requirement Specification

## User Requirement Specification

### Guest Requirement

Guest is a person who doesn’t have access to the system. Guest can use some basic functions in the system. To use main functions, guest must login or register new account. These are some functions of guest:

* Register.
* Search (jobs, event, and contest).
* Login.

### Student Requirement

Students is a type account, Guest became a students after register with role students. Student can use many functions in the system and they also can use functions of guest.

Some function of students:

* Manage Profile
* Manage Job
* Apply Job
* View Interview Info
* View Job History
* Add Job To favourites
* View Favourites Jobs
* Manage Event
* View Event History
* Add Event To favourites
* View favourite Events
* Manage Material
* Share Material
* Feedback Contest

### Employer Requirement

Employer is also type of account, Employer have many functions to create and manage such as:

* Manage Job
* Create Job
* Manage Event
* Create Event
* Manage Contest
* Create Contest
* Manage Test Bank
* Create Question
* Upload Questions
* Edit Employer Info
* Approve or Reject CV

### Staff Requirement

Staff is account to help website manage content of job, event, contest when it had been created.

And they have some functions :

* Manage All Events
* Approve Event
* Reject Event
* Inactive Event
* Active or Inactive Material
* Active or Inactive Job
* Active or Inactive Contest

### Admin Requirement

Admin is account have role to manage another account.

* Manage Account
* Create Account
* Update Role
* Active or Inactive Role

## System Requirement Specification

### External Interface Requirement

#### User Interface

• The interface of website is clear, easy for user to use (Firefox Browser, Chromes with Resolution 1024\*768 or bigger).

#### Hardware Interface

#### Software Interface

Firefox Browser, Chromes with Resolution 1024\*768 or bigger and support JavaScript and HTML5

#### Communication Protocol

• Website using HTTP protocol for communication between the web browser and the web server.

### System Overview Use Case

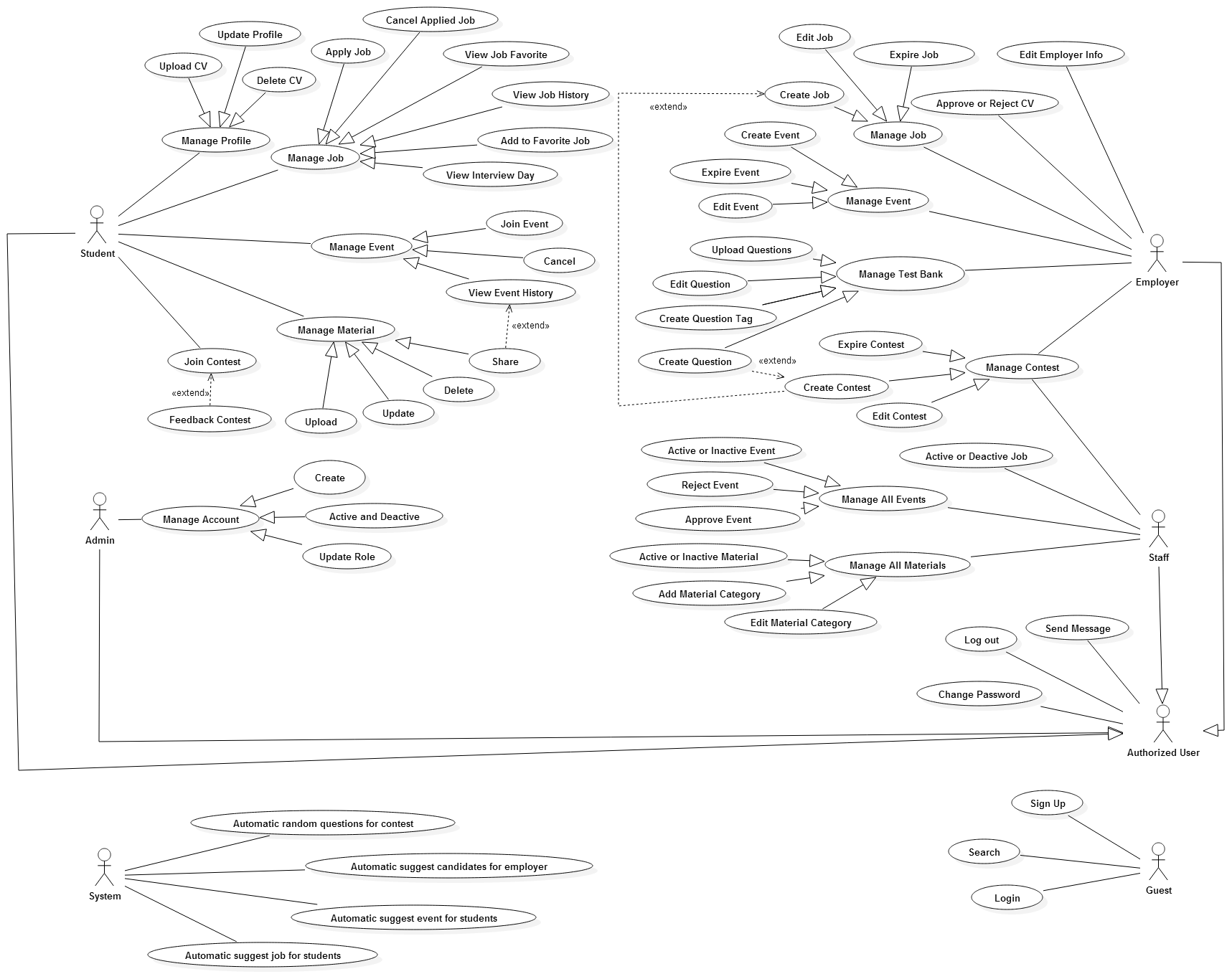


Figure 2: System Overview Use Case

### List of Use Case

#### <Guest>Overview Use Case

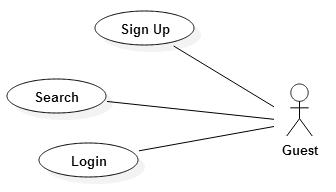


Figure 3: <Guest> Overview Use Case

##### <Guest> Register

Use Case Diagram

Figure 4: <Guest> Register

Use Case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – CPS001** | | | |
| **Use Case No.** |  | **Use Case Version** |  |
| **Use Case Name** |  | | |
| **Author** |  | | |
| **Date** |  | **Priority** |  |
| **Actor:**  **Summary:**  **Goal:**  **Triggers:**  **Preconditions:**  **Post Conditions:**   * **Success:** * **Fail:**   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Respone | |  |  |  | |  |  |  | |  |  |  |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | ctor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  |   **Relationships:** N/A  **Business Rules:** | | | |

Table 10: <Guest> Register

#### <Student>Overview Use Case

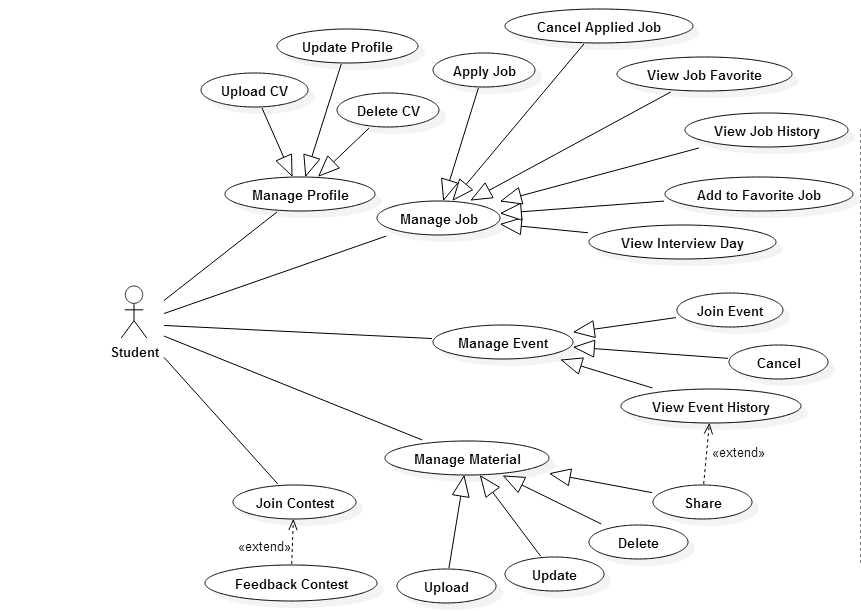


Figure 4: < Student > Overview Use Case

#### <Employer>Overview Use Case

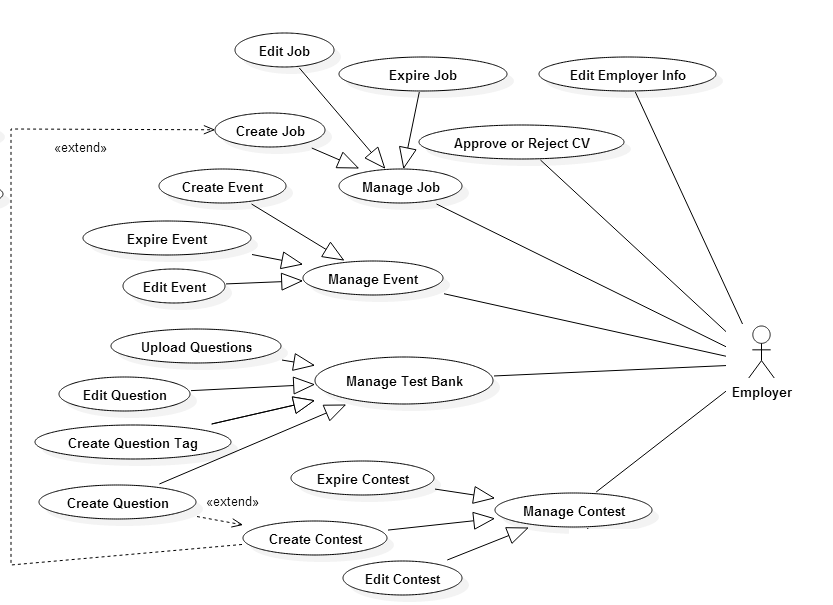


Figure 5: < Employer > Overview Use Case

#### <Staff>Overview Use Case

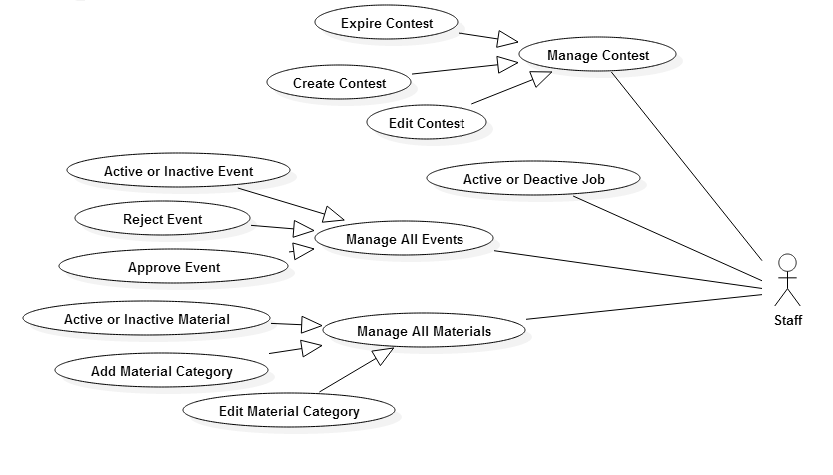


Figure 6: < Staff > Overview Use Case

#### <Admin>Overview Use Case

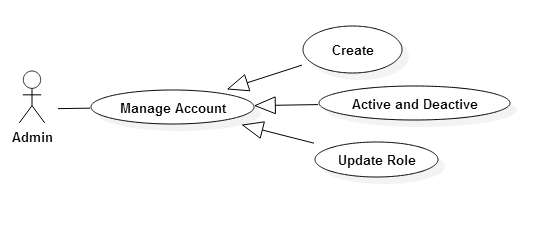


Figure 7: < Staff > Overview Use Case

#### <System>Overview Use Case

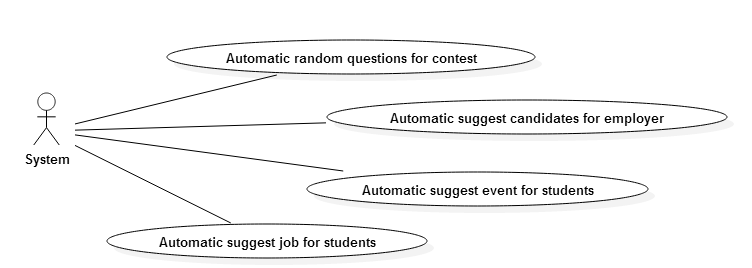


Figure 8: < System > Overview Use Case

## Software System Attribute

### Usability

#### Graphic User Interface

* All the texts, labels and should be written in Vietnamese.

#### Usability

* Administrator, staff, employer should need less than one week of training to be productive with the system.
* Guest and member can use the system easily without training.

#### Installation

* The system must be easy to deploy.
* The attached manual guide must be clear. User can read and do themselves without developer’s help.

### Reliability

* Notification would be sent immediately for users to inform user about new contents.
* Rate of image recognization: 95% (test on 100 records)
* Các công ty đăng ký trước khi được sử dụng đã dc website kiểm duyệt và xác thực.

### Availability

### Security

* The system is divided into separated modules.

The code is easy to maintain and upgrade.

### Maintainability

* Privacy: Each role of user has a specific permission to interact with system.
* System always checks authorization and authentication before doing anything.
* Only administrator can grant permission to staff.

### Portability

### Performance

## Conceptual Diagram

### Data Dictionary

|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Software Design Description

## Design Overview

## System Architectural Design

### Web Application architecture description

### Mobile Application architecture description

## Component Diagram

## Detailed Description

### Class Diagram

### Class Diagram Explanation

#### Role

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Interaction Diagram

#### <Member> View Friend List

## User Interface Design

### Guest Interface Design

#### Login

Figure 12: Login

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
|  |  |  |  |  |  |  |  |

**Buttons/ Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
|  |  |  |  |  |

## Database Design

### Logical Diagram

### Data Dictionary

## Algorithms

### Document Breakdown

#### Definition

#### Define Problem

#### Solution

#### Complexity

#### Flow Chart

### String Comparison

#### Define Problem

#### Requirement

#### Solution

#### Example

# System Implementation & Test

## Introduction

### Overview

### Test Approach

## Database Relationship Diagram

### Physical Diagram

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| Entity Name | Description |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 14: Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Domain** | **Null** |
|  |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table 15: Attribute Data Dictionary

## Performance Measures

### Clustering Performance

## Test Plan

### Features to be tested

### Features not to be tested

* qweqwe
* qweqw

## System Testing Test Case

### Guest Test Case

#### Search Event

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Inter-test Case Dependence** | **Result** | **Test Date** | **Note** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

# Software User’s Manual

## Installation Guide

### Setting up environment at server side

#### Hardware requirements

#### Software requirements

### Deployment at server side

#### Prepare deployment package

#### Configure Server before deploy

#### Deploy web application on server

### Setting up the environment at client side

#### Setting up for computer

## User Guide

# Appendix